# Visual and Textual Common Semantic Spaces for the Analysis of Multimodal Content

Lady Viviana Beltrán Beltrán

# LA ROCHELLE UNIVERSITÉ

*École doctorale Euclide*

Laboratoire Informatique, Image, Interaction (L3i)

**THÈSE** présentée par :

**Lady Viviana Beltrán Beltrán**

soutenance le : **17 Juin 2021**

pour obtenir le grade de : **Docteur de La Rochelle Université**

Discipline : **Informatique**

# Visual and Textual Common Semantic Spaces for the Analysis of Multimodal Content

---

| | | |
|---|---|---|
| **Rapporteurs** | **Aurélie BUGEAU** | Professeur des Universités, LaBRI, Université de Bordeaux |
| | **Verónique EGLIN** | Professeur des Universités, Université Claude Bernard Lyon |
| **Examinateurs** | **Jean-Yves RAMEL** | Professeur des Universités, Université de Tours |
| | **Simone MARINAI** | Professeur des Universités, University of Florence |
| **Directeurs** | **Antoine DOUCET** | Professeur des Universités, L3i, La Rochelle Université |
| | **Nicholas JOURNET** | Maître de Conférences HDR, LaBRI, Université de Bordeaux |
| **Co-encadrants** | **Mickaël COUSTATY** | Maître de Conférences, L3i, La Rochelle Université |


RÉGION
Nouvelle-
Aquitaine


La Rochelle
Université


LaBRI


BROAD
INSTITUTE

# Abstract

Multimodal learning involves the use of multiple senses (touch, visual, auditory, etc.) during the learning process to better understand a phenomenon. In the computational domain, we need systems to understand, interpret, and reason with multimodal data, and while there have been enormous advances in the field, many of the desired capabilities remain beyond our reach. The objective of such systems is to leverage different semantically related data types to output better predictions for a phenomenon of interest. For example, for users with sensory disabilities such as visual, to carry out daily tasks such as making a purchase or finding a place in a city, the visual information of their environment has to be transformed into a different modality with more semantic meaning. A system for this purpose could use auditory information provided by the user that specifies what information is required and that can be easily transformed into textual data, and visual information such as images obtained from its surroundings to help the user make a decision. Therefore, it would be a multimodal system leveraging information from three different modalities: auditory + text + images.

When it comes to the computational side, working with multimodal data comes with several challenges. This thesis focuses on advancing multimodal learning research through various scientific contributions: we simplify the creation of deep learning models by proposing frameworks that find a common semantic space for visual and textual modalities using deep learning as the backbone tool; we propose competitive strategies to address the tasks of cross-modal retrieval, scene-text visual question answering, and attribute learning; we address various data-related issues like imbalance and learning when not enough data is annotated. These contributions aim to bridge the gap between humans (such as non-expert users) and artificial intelligence to tackle everyday tasks.

Our first contribution aims to evaluate the effectiveness of a multimodal system that receives images and text and retrieves relevant multimodal information. This approach allows us to perform a complete study to evaluate the effectiveness of a cross-modal retrieval system with deep learning as the backbone tool. The cross-modal feature allows the formulation of the queries in the form of images or text and retrieves relevant multimodal data. With this approach, we can evaluate the ability of the model to produce effective multimodal representations and to

handle any multimodal query with a single model. Subsequently, in our second contribution, we adapt the system to perform a recent task called scene-text visual question answering (ST-VQA). The aim is to teach traditional VQA models to read the text contained in natural images. This task requires us to perform a semantic analysis between the visual content and the textual information contained in associated questions to give the correct answer. We find this task very relevant in the multimodal context since it truly forces us to jointly develop mechanisms that reason about visual and textual content.

Our latest contributions point to data-related issues. Data is one of the most important factors in aiming for good performance. Therefore, we determined that a relevant skill is to understand how to properly clean and analyze data and create strategies that can take advantage of it. We address very common and frequent issues such as noise, imbalance, and insufficient annotated data. To evaluate our strategies, we consider the problem of attribute learning. Attribute learning can complement category-level recognition and therefore improve the degree to which machines perceive visual objects. In the first study, we cover two key aspects: imbalance and insufficient labeled data. We propose adaptations to classical imbalanced learning strategies that cannot be directly applied when using multi-attribute deep learning models. In the second study, we propose a novel strategy to exploit class-attribute relationships to learn predictors of attributes in a semi-supervised learning way. Semi-supervised learning permits harnessing the large amounts of unlabelled data available in many use cases in combination with typically smaller sets of labeled data.

**Keywords:** multimodal learning, deep learning, information retrieval, multimodal fusion, image classification, attribute learning, visual question answering.

# Résumé

L'apprentissage multimodal implique l'utilisation de multiples sens (tactile, visuel, auditif, etc.) au cours du processus d'apprentissage pour mieux comprendre un phénomène. Dans le domaine du calcul, nous avons besoin de systèmes pour comprendre, interpréter et raisonner avec des données multimodales, et bien qu'il y ait eu d'énormes progrès dans le domaine, de nombreuses capacités souhaitées restent hors de notre portée. L'objectif de ces systèmes est d'exploiter différents types de données sémantiquement liés pour produire de meilleures prédictions pour un phénomène d'intérêt. Par exemple, pour les utilisateurs ayant des handicaps sensoriels tels que visuels, pour effectuer des tâches quotidiennes telles que faire un achat ou trouver une place dans une ville, l'information visuelle de leur environnement doit être transformée en une modalité différente avec une signification plus sémantique. Un système à cette fin pourrait utiliser des informations auditives fournies par l'utilisateur qui spécifient quelles informations sont requises et qui peuvent être facilement transformées en données textuelles, et des informations visuelles telles que des images obtenues de son environnement pour aider l'utilisateur à prendre une décision. Il s'agirait donc d'un système multimodal exploitant les informations de trois modalités différentes: auditif + texte + images.

En ce qui concerne le calcul, travailler avec des données multimodales présente plusieurs défis. Cette thèse se concentre sur l'avancement de la recherche sur l'apprentissage multimodal à travers diverses contributions scientifiques: nous simplifions la création de modèles d'apprentissage profond en proposant des cadres qui trouvent un espace sémantique commun pour les modalités visuelles et textuelles en utilisant l'apprentissage profond comme outil de base; Nous proposons des stratégies compétitives pour aborder les tâches de recherche intermodale, de réponse visuelle aux questions de texte de scène et d'apprentissage d'attributs; Nous abordons divers problèmes liés aux données tels que le déséquilibre et l'apprentissage lorsque les données annotées sont insuffisantes. Ces contributions visent à combler le fossé entre les humains (comme les utilisateurs non experts) et l'intelligence artificielle pour s'attaquer aux tâches quotidiennes.

Notre première contribution vise à évaluer l'efficacité d'un système multimodal qui reçoit des images et du texte et récupère les informations multimodales pertinentes. Cette approche nous permet de réaliser une étude complète pour évaluer l'efficacité d'un système de récupération

cross-modal avec le deep learning comme outil de base. La fonction cross-modal permet de formuler les requêtes sous forme d'images ou de texte et de récupérer les données multimodales pertinentes. Avec cette approche, nous pouvons évaluer la capacité du modèle à produire des représentations multimodales efficaces et à gérer toute requête multimodale avec un seul modèle. Par la suite, dans notre deuxième contribution, nous adaptons le système pour effectuer une tâche récente appelée réponse visuelle aux questions de texte de scène (ST-VQA). L'objectif est d'apprendre aux modèles VQA traditionnels à lire le texte contenu dans des images naturelles. Cette tâche nous oblige à effectuer une analyse sémantique entre le contenu visuel et les informations textuelles contenues dans les questions associées pour donner la bonne réponse. Nous trouvons cette tâche très pertinente dans le contexte multimodal car elle nous oblige vraiment à développer conjointement des mécanismes qui raisonnent sur le contenu visuel et textuel.

Nos dernières contributions mettent en évidence des problèmes liés aux données. Les données sont l'un des facteurs les plus importants pour viser de bonnes performances. Par conséquent, nous avons déterminé qu'une compétence pertinente consiste à comprendre comment nettoyer et analyser correctement les données et créer des stratégies qui peuvent en tirer parti. Nous abordons des problèmes très courants et fréquents tels que le bruit, le déséquilibre et l'insuffisance des données annotées. Pour évaluer nos stratégies, nous considérons le problème de l'apprentissage des attributs. L'apprentissage des attributs peut compléter la reconnaissance au niveau des catégories et donc améliorer le degré de perception des objets visuels par les machines. Dans la première étude, nous couvrons deux aspects clés. Déséquilibre et données étiquetées insuffisantes. Nous proposons des adaptations aux stratégies d'apprentissage déséquilibrées classiques qui ne peuvent pas être directement appliquées lors de l'utilisation de modèles d'apprentissage profond multi-attributs. Dans la deuxième étude, nous proposons une nouvelle stratégie pour exploiter les relations classe-attribut pour apprendre les prédicteurs d'attributs de manière semi-supervisée. L'apprentissage semi-supervisé permet d'exploiter les grandes quantités de données non étiquetées disponibles dans de nombreux cas d'utilisation en combinaison avec des ensembles généralement plus petits de données étiquetées.

**Mots-clés:** apprentissage multimodal, apprentissage en profondeur, recherche d'informations, fusion multimodale, classification d'images, apprentissage d'attributs, réponse visuelle aux ques-

tions.

# Acknowledgement

This work has been supported by the French region of Nouvelle Aquitaine under the ANIMONS project and by the MIRES research federation.

**Personal Acknowledgement**

This thesis has represented a great experience in my professional and personal life.

First of all, I want to thank all my thesis advisors, Professor Antoine Doucet, Nicholas Journet, Mickaël Coustaty, and Juan C. Caicedo for their support and guidance in pursuing this Ph.D.

A special thanks to Juan. Thank you very much for the time and support provided almost from the beginning of my thesis. I appreciate your role not only as a supervisor but, more importantly, as a great friend.

A special thanks to all the members of the jury. Thanks to the reviewer Aurélie Bugeau for her comments and suggestions to improve my thesis manuscript. A special thanks to the reviewer Verónique Eglin who agreed to review my work in a very short period of time. I would like to thank Jean-Yves Ramel and Simone Marinai for agreeing to evaluate my thesis in the role of examiners. Thanks to the French region of Nouvelle Aquitaine under the ANIMONS project and by the MIRES research federation for their generous financial support.

Thank the L3i laboratory and its administration for the support provided to pursue my thesis in the best way. More specifically to Kathy Theuil, Erlandri Chaigneaud, Muhammad Muzzamil Luquan, Dominique Limousin, Dominique Besse and Mourad Al Natour (Responsables du Pôle Audiovisuel).

Thanks to all my friends who have been there to support me. Thanks to my Colombian friends who visited me or kept in contact with me, Edwin, Daniela, Daniel, Karen, Michel, Óscar, Andrés G, Andrés P, Areli, Angélica, Mónica, Nelson. My friends here in France with whom I have had so many new experiences. Thanks to my friends and office mates Quoc-Bao Dang, Thanh-Nam Le, Florian Lardeux, Thi Tuyet Hai Nguyen (Hai, who made me try some amazing

Vietnamese vegetarian food), Jordan, Souhail, Khoa, Marwa, Fanny, Lionel, Jhony, Zuheng.

A special thanks to my dear friend Marcela. From the beginning you have helped me in all kinds of matters. Also, thanks to my friends Elvys, Valentin and Beatriz for all the experiences and the great friendship you have offered me.

A special thanks to you, Matthieu and your family who have been incredibly supportive, especially during the last part of my PhD.

And of course, to my cats, Hanna, Nala, Zoé, Nroll, Maya (and others - I forgot their names :D).

Last but not least, my deepest thanks to my beloved family. Without them, I would not have been able to achieve this PhD. Por último, pero no menos importante, quiero agradecer a mi familia, mi madre, mi hermana, mis abuelos, tíos y tías quienes han estado siempre en mi corazón. A mi madre y hermana, quienes a pesar de la distancia, siempre han estado apoyándome en los momentos buenos y difíciles, un agradecimiento con todo mi corazón.

# Contents

# CHAPTER 1

# Introduction

## 1.1 Motivation

The blog post of ABI Research[1] presents multimodal learning as the future of artificial intelligence with many tools being incorporated already in many domains. With the development of the internet and the explosion of the number of websites to share media content, the need to semantically understand and interpret abstract or raw data has increased. This data comes in very diverse shapes and formats that represent different channels or modes of communication — visual, textual, oral, etc. This is also referred to as multi-modality. For example, a video can be associated with descriptions or tags in natural language, or even with representative images of its content. In the same way, images are associated with tags or textual explanations, while text data such as Wikipedia[2] articles usually come with images to better express the principal idea. Different modalities may or may not be engaged in the process of information that allows us to

[1] Multimodal Learning And The Future Of Artificial Intelligence; https://www.abiresearch.com/blogs/2019/10/10/multimodal-learning-artificial-intelligence/ (accessed Jan 2021)

[2] https://en.wikipedia.org/wiki/Wikipedia:About (accessed Jan 2021)

understand more about a phenomenon. For example, an application to help visually impaired users based on a question-answering system should be able to give correct and relevant information to the user (see Figure 1.1). In the example, if a user is interested in the color of the bus, the system should be able to understand what information to extract to give a semantic and correct answer. Other information such as the text contained in the scene may not be relevant to the user. Nowadays, developing competitive and acceptable mechanisms that encode and decode our external surroundings manifested as different modalities is a must. We need multimodal systems to understand, interpret, and reason (at some level) with multimodal data. The objective of such systems is to consolidate heterogeneous and disconnected data to output better predictions.


Figure 1.1: Example of different levels of engagement. A system that analyzes visual information to give correct answers requires semantically analyzing the content of the target scene. This helps discriminate between relevant and non-relevant data that are specific to each question. For the first 3 questions, the textual information contained in the image is not engaged in the process to answer the question. For the last 4 questions, textual data is especially required to answer the questions correctly.

Although the earliest examples of multimodal research were in the area of audio-visual speech recognition (AVSR) [1], the most recent category of multimodal applications makes an emphasis on language and vision commonly known as media description [2]. Images and text are the most common resources because of their facility of creation and storage. From one side, language has been the primary means of communication [3, 4]. Spoken or written, it allows us to share our

ideas, opinions, views, emotions, and reach others. The research community on reading systems has made significant advances [5]. On the other hand, the increase of image creation and sharing over the years as it is reported by Mary Meeker's Internet Trends Reports[3] places images as an increasingly and relevant way of communication. For example, in 2014, people uploaded an average of 1,8 billion digital images every single day. According to the most recent Internet trends report, in 2019, images are being included in almost all social applications[4]. Social platforms such as Twitter[5], which in its beginning was a text-only application, has evolved to include features of sharing of multimedia content such as images and videos with more than 50% of tweets containing one of these data types. Instagram[6] has also increased at an exponential rate the number of users that every day share multimedia content. In turn, this has increased the need for data-driven applications in many modalities such as image-powered search, image-driven discovery, etc.

In a general way, visual data can be more telling (a picture is worth a thousand words). However, not all visual data is accessible, easy to read, or understandable for all users. For example, for visually impaired users it is required to transform visual content present in everyday tasks (like making a purchase, using public transportation, finding a place in a city, etc...) into usable information such as a textual representation easy to transform into a sonorous representation. Another example may be novice experts (or non-experts) looking for supporting information. As a recent graduate in the medical domain, it is very helpful to support a diagnosis by accessing the information on another available previous diagnosis. Also, in complex tasks such as interpreting medical imaging scans (x-rays, CT, MRI) that require a highly-skilled, manual job, and many years of training. Associating textual descriptions to these types of complex data is very appreciated and may help in giving a correct diagnosis. Decision support tools are rapidly gaining acceptance in this domain [6, 7]. In the research community, this is reflected by the amount of research arising around this area. Some of the most representative areas (see Figure

[3] https://www.bondcap.com/report/it14/ (accessed Jan 2021)

[4] This report underlines the most important statistics and technology trends on the internet. The report aims to present the evolution of image Creation + Sharing from sources such as Instagram releases (2011-2019), Pinterest (2011-2019), Google (2017-2019), Twitter, Canvas, among others. https://www.bondcap.com/report/itr19/ (accessed Jan 2021)

[5] https://about.twitter.com/ (accessed Jan 2021)

[6] https://about.instagram.com/ (accessed Jan 2021)

1.2) include image annotation where the task is to tag/annotate the image with representative words [8]; image captioning where the task is to generate a text description of the input image [9]; visual question answering where the task is to construct a system to answer questions presented in an image using natural language [10]; and multimodal information retrieval which goal is to access multimedia content through a retrieval system and that can also produce content that meets diverse demands [2].

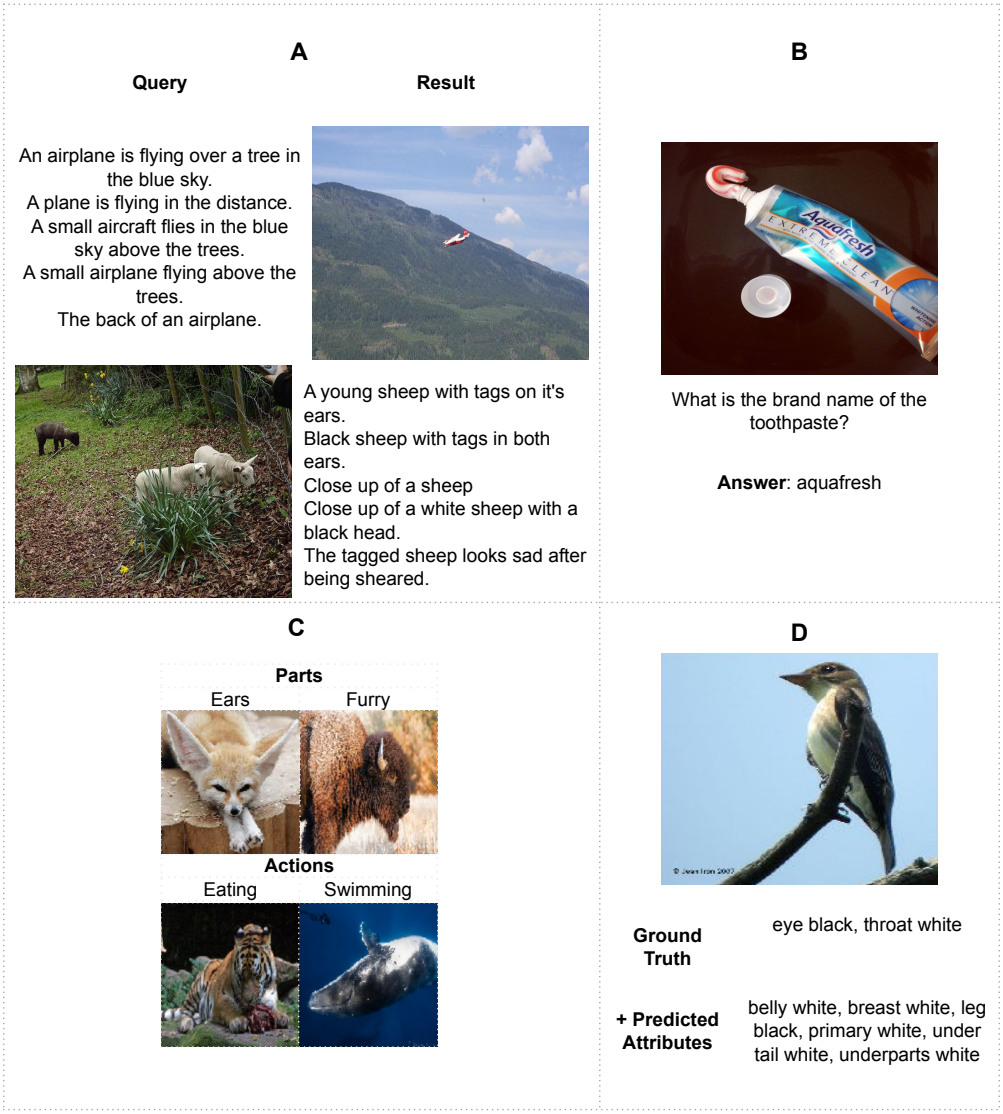



Figure 1.2: Examples of multimodal applications including images and text data. A) Multimodal/cross modal information retrieval; B) Scene text via visual question answering; C) Image annotation for different vocabulary spaces; D) Attribute assignation/discovery.

We described how studying the dependencies among different modalities allows us to discover patterns that may help to understand a phenomenon. The correlation of these multiple sources is generally semantic and therefore, it may provide complementary and additional information richer than when working with only one modality. This thesis is centered on multimodal learning. We propose to limit our explorations to two of the most frequent and principal modalities studied in the state of the art, the visual and the textual modalities. We address the problem by proposing frameworks that find a common semantic space for both modalities using deep learning as the backbone tool. We evaluate our strategies in several well-studied as well as very recent applications in the state of the art over several widely used databases. The following section presents the main factors and challenges when addressing the problem of multimodal learning for images and text.

## 1.2 Problem Statement

As explained before, multiple modalities related to a phenomenon provide different perspectives when studying it. We can learn complementary and additional information than when working with only one modality; we can discover patterns or changes that are only visible when two or more modalities are studied; we can learn latent correlations that explain behaviors, etc. This is the way our brains work. It develops stronger memory circuits when experiencing learning through multiple sensory modalities, such as vision, hearing, and movement [11]. Hence, if we aim to create systems with a better understanding of the world around us, we need to create systems that learn from multiple modalities [2]. However, for a machine, learning from multiple modalities comes with a set of problems and challenges due to the differences in the characteristics of each type of data. This is due to the inherent nature of each data type. For example, a model trained with images only understands this data type and can only be used to obtain visual features, same with a model trained with only textual data. Therefore, we cannot directly combine features obtained with the visual model with features obtained with a textual model as they reside in different spaces; finally, there exist difficulties in the utilization of current state-of-the-art tools to address these problems. Next, we describe each one of them.

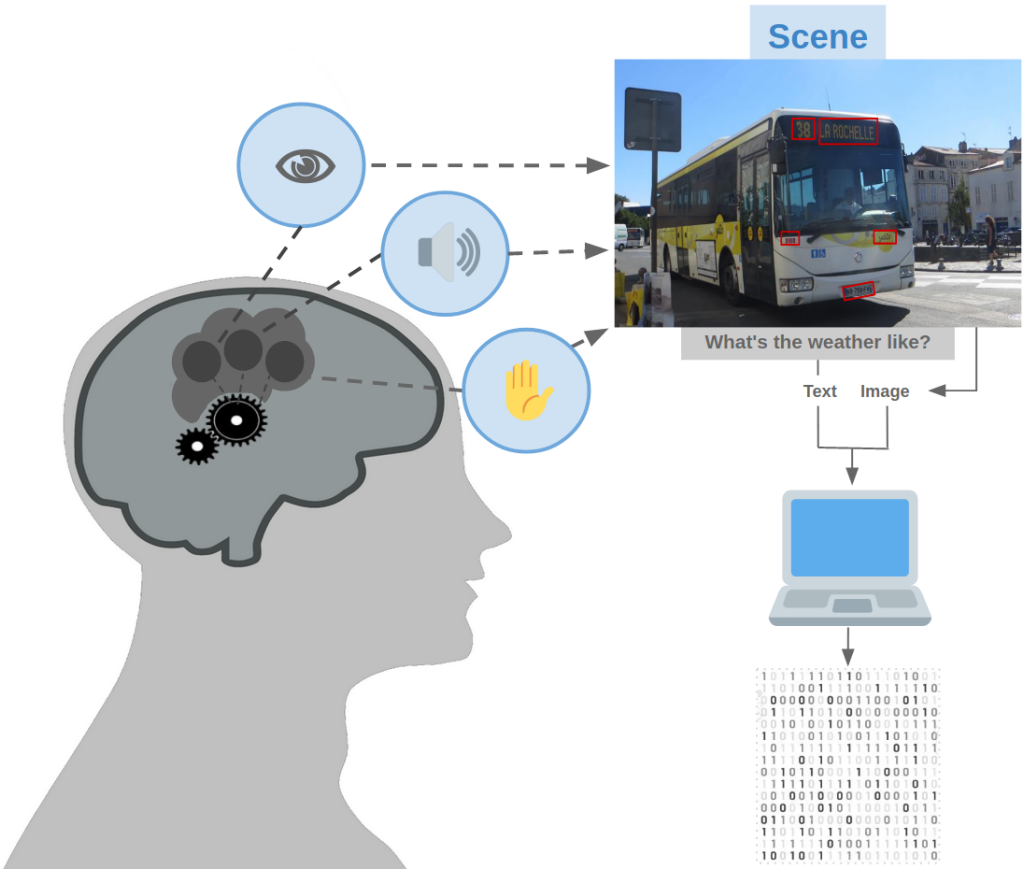



Figure 1.3: For humans to take the information from separate sensory modalities and match it appropriately is an easy task. This is natural since we live in a world of multimodal objects and therefore, our perception is multimodal. For example, we can use our senses (vision, hearing, touch, etc) to give the correct answer to a question based on the information contained in a scene. For computers, multimodal information such as visual data represented as an image or questions represented as textual data is stored and represented as numbers. Therefore, finding a semantic meaning from different modalities represents a tough challenge.

**Learning from multiple modalities.** At a glance, combining different types of data to improve a prediction appears as a simple task as this is something that as humans, our brain does automatically. Our brain can reason through a high-order semantic abstraction from different signals. For example, we can perceive the objects and the landscape in a scene and then process the information contained in that piece of visual data by using previously learned knowledge. But in practice, teaching a machine to automatically extract semantic meaning from a combination of different modalities represents one of the biggest challenges in the research community.

This problem is due to the inherent properties of each data type. Precisely, the nature of the data itself creates conflicts when combining different modalities. Computers are digital devices that, in a simplified way, store and represent all data as numbers (see Figure 1.3). Therefore, images and text are represented as numbers that, in a higher-order, contain a semantic meaning. For this, we cannot combine directly textual features with visual features because they are contained in different feature spaces (see Figure 1.4). Also, each modality may have a different quantitative influence over the phenomenon studied. State-of-the-art models work almost perfectly for tasks such as image classification (the best ImageNet classification accuracy at the moment of writing this document is of $\approx 90\%$ [12]) or text classification [13]. However, when learning from two data types as different as images and text, this requires a reasoning component that finds the correlations and correspondences among them. We need to create systems that find a new feature space in which visual and textual data can be mapped and represent the same semantic meaning.

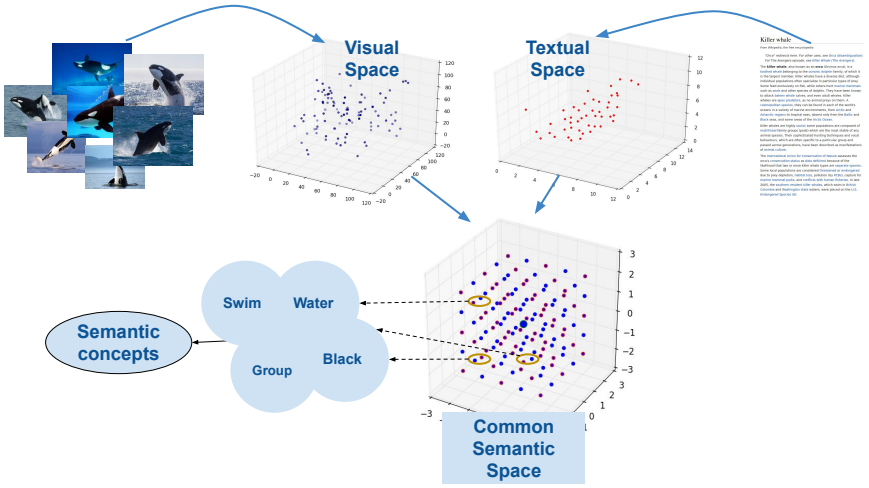



Figure 1.4: Multimodal semantic space: traditional text-based search models no longer satisfy people's needs for multimedia information. Humans search in multimodal collections by using high-level semantic concepts. But when the collection contains visual data, it is very challenging to automatically extract its semantic content and therefore relevant content may not be retrieved. Combining multiple data types such as images and text allows us to exploit complementarities that exist between them. Different modalities reside in different feature spaces, whereby, a mapping function that transform the modalities into a common and semantic feature space is required. Learning this mapping still represents a complex challenge.

**Problems associated with the data.** Another important factor that plays a role to have systems with good performance is the data itself. Data is one of the most important keys for success when studying a simple or a complex phenomenon, and the success of machine learning models can be attributed largely to the data itself [14]. In the most common scenario, the main requirement to obtain a good performance is to fulfil two factors concerning the data: quality and quantity. Unfortunately, in reality, this is not possible and researchers have to deal with these two problems in almost every task. The research community through conferences and competitions such as ImageNet [15] and Kaggle [16] challenges, and motivates researchers to create models and strategies to solve different tasks in computer vision. They also create and make available many databases [8] that hugely contribute to fostering research in the field. This has contributed to advancement in the field, but there are still many problems requiring better solutions. Among the problems, we describe the most important ones: 1) Each data type comes with different levels of noise. For example, data that contains wrong, highly subjective, and non-discriminative annotations, or data with very bad quality; 2) Data comes in different shapes and formats. This causes mismatches between the amount of required vs the available computational resources; 3) Data that contains imbalance present at different levels. This can be due to irregular or very rare events in the data, lack of annotated data for some or all classes, etc.; 4) The problem of having insufficient amount of data. Models in the current state of the art require a large amount of data that is not always available. A suitable quantity depends also on the target problem. 5) Finally, there is a lack of databases not even without annotations to study and explore more specific and complex problems. Machine learning is used in many applications today. However, there is a lack of expert-level databases to study very specific problems. For example, in medicine, there may be sensitive user data that makes its availability difficult. For all these reasons, we extend our efforts to the treatment of the data itself. Throughout the document, we present different strategies that aim to solve the problems around the nature and state of the data along with the appropriate processing for the state-of-the-art applications.

**Difficulties for non-expert users.** Finally, we emphasize the complexity of recent developments, especially for non-expert users. Machine learning is nowadays the major approach to perform multimodal learning because of its outstanding success in diverse applications. Its ob-

jective is to design algorithms to assist computer systems and progressively improve their performance. In general, it consists of algorithms or models that automatically learn from training data and make decisions without being specifically programmed for them. Andrew Ng, the co-founder of Coursera and professor at Stanford University, defines machine learning as follows: "the science of getting computers to act without being explicitly programmed" [7]. Specifically, models based on neural networks (or deep learning [17]) have shown to have highly competent performance in many applications.

Currently, these models are being used in diverse domains in research as well as the industry [18]. Microsoft, Facebook, Google, and Amazon are examples of top companies leading the integration of deep learning into everyday applications [19–22]. The most significant advancements in technology are including the use of machine learning for applications such as self-driving vehicles [23], algorithms in robotics [24–26], many analytic tools [27], chat-bots [28], bioinformatics [29], rise and fall of stocks [30], and others [31], etc. The most current example is the explosion in the number of articles addressing the problem of COVID[8] and using deep learning as the main tool [32, 33]. However, the learning curve of deep learning is very hard on users without a background in computer science or applied mathematics, especially the most recent developments that concern very complex models. A recent trend that aims to facilitate the use of deep learning models is explainable artificial intelligence (XAI). XAI emerged to understand which aspects of the input data drive the decisions of a deep learning model [34]. It can be defined as a set of tools with great diversity in the definitions, approaches, and techniques used by researchers to provide a rationale for the decisions for deep learning models. These tools can also be useful for non-expert users and the research around facilitating its use is getting more attention [35]. On the other hand, traditional models also offer competitive performances and may be easier and faster to use for these non-expert users. Besides, these models contain much information not completely explored and exploited and can offer impressive results. In this thesis, we focus our efforts on creating strategies that leverage the capabilities and potential of deep learning tools. We test these strategies on state-of-the-art applications for multimodal learning with images and textual data. We hope non-expert users will find these strategies useful when

[7] https://www.coursera.org/learn/machine-learning (accessed Jan 2021)

[8] Global research on coronavirus disease (COVID-19); https://www.who.int/emergencies/diseases/novel-coronavirus-2019/global-research-on-novel-coronavirus-2019-ncov (accessed Dec 2020)

considering deep learning as a support tool.

To summarize, many problems remain unsolved in the state of the art when working with multimodal data. These problems are derived from the combination of modalities in different feature spaces, the nature of the data itself, and the complexities and limited explainability of current tools in the state of the art. We have mentioned the most frequent ones and emphasized those on which we focus throughout the development of this work. Next, we present more details of current work, challenges, and applications in multimodal learning.

## 1.3 Multimodal learning

Learning from multiple modalities is advantageous since it provides diverse points of view, and can contain additional and useful information or reveal hidden patterns in the study of a phenomenon. On a higher level, pieces of information from different modalities can represent the same semantic concept. However, computationally, each modality represents a data type residing in a different space, which therefore requires different processing. Although images and text are the most common data types (see Section 1.1), there still exists a large gap to interpret semantically some types of content, especially visual. This gap increases when studying visual and textual data in a multimodal scenario. Given the nature of each type of data, some unique computational challenges emerged. These along with their main applications are explained next.

### 1.3.1 Multimodal challenges

Multimodal Machine Learning brings some computational challenges given the heterogeneity of the data. These need to be addressed when studying applications involving more than one modality. These challenges are representation, fusion, translation, alignment, and finally, co-learning [2]. We explained them next.

**Representation.** Good representations are important for the performance of machine learning models. This challenge is concerned with how to represent and summarize multimodal data. This poses many difficulties such as how to combine data from different sources with a heterogeneity gap, how to deal with different levels of noise contained in real and available data, and

also how to deal with missing data. While the development of uni-modal representations has been extensively studied [36–39], the boost of more complex multimodal representations occurred in recent years [2, 40, 41]. The two most popular types of multimodal representations are joint representations and coordinated representations (see Figure 1.5 - A (above) on page 13). For the case of finding a common semantic space, the preferred category lies in joint representations. For example, to construct a multimodal representation using neural networks, we can project the neural representation of each modality into a joint space composed of one or several layers, and that later, can be used to perform semantic predictions.

**Fusion.** This challenge lies in the integration of information extracted from different unimodal data into a single compact multimodal representation to perform a prediction. This task can be performed in an early, middle-level, attention-based, or late fusion approach. Essentially, this refers to different fusion points [42, 43]. One of the main problems around this challenge resides in how to adjust the weights of the fusion function (see Figure 1.5 - A (below)).

**Translation.** As its name suggests, this challenge deals with the translation from one modality to another. Some applications in this category include image caption [44] (see Figure 1.5 - B); video description [45], and cross-modal retrieval [46].

**Alignment.** This challenge targets finding the relationships and correspondences between subcomponents of instances from two or more modalities [47]. For example, to generate region-level descriptions it is necessary to find the correspondences between specific words and image regions (see Figure 1.5 - C). This challenge has not been widely studied because it faces many difficulties. For example, there exist very few databases with explicitly annotated alignments. Also, it is difficult to design similarity metrics between modalities. And finally, it is not clearly defined as there may exist multiple possible alignments and not all elements in one modality have correspondences in another.

**Co-learning.** This specific area of multimodal machine learning seems to be under-studied. In this challenge, knowledge from one rich modality is used to improve the modelling of a poor one (with little to no data). The richer or helper modality is usually used during training but not

during test time. The main applications are transfer learning and zero-shot learning (see Figure 1.5 - D). Due to the heavy resource consumption in acquiring high-quality labelled data sets, recent approaches are following a co-learning strategy [48]. In reality, tackling almost any task in multimodal learning requires handling more than one of these challenges. For example, representation is responsible for the process of finding quality representations of multiple modalities, while fusion directs how this process is carried out. The rest of the challenges are more related to the type of application they address.

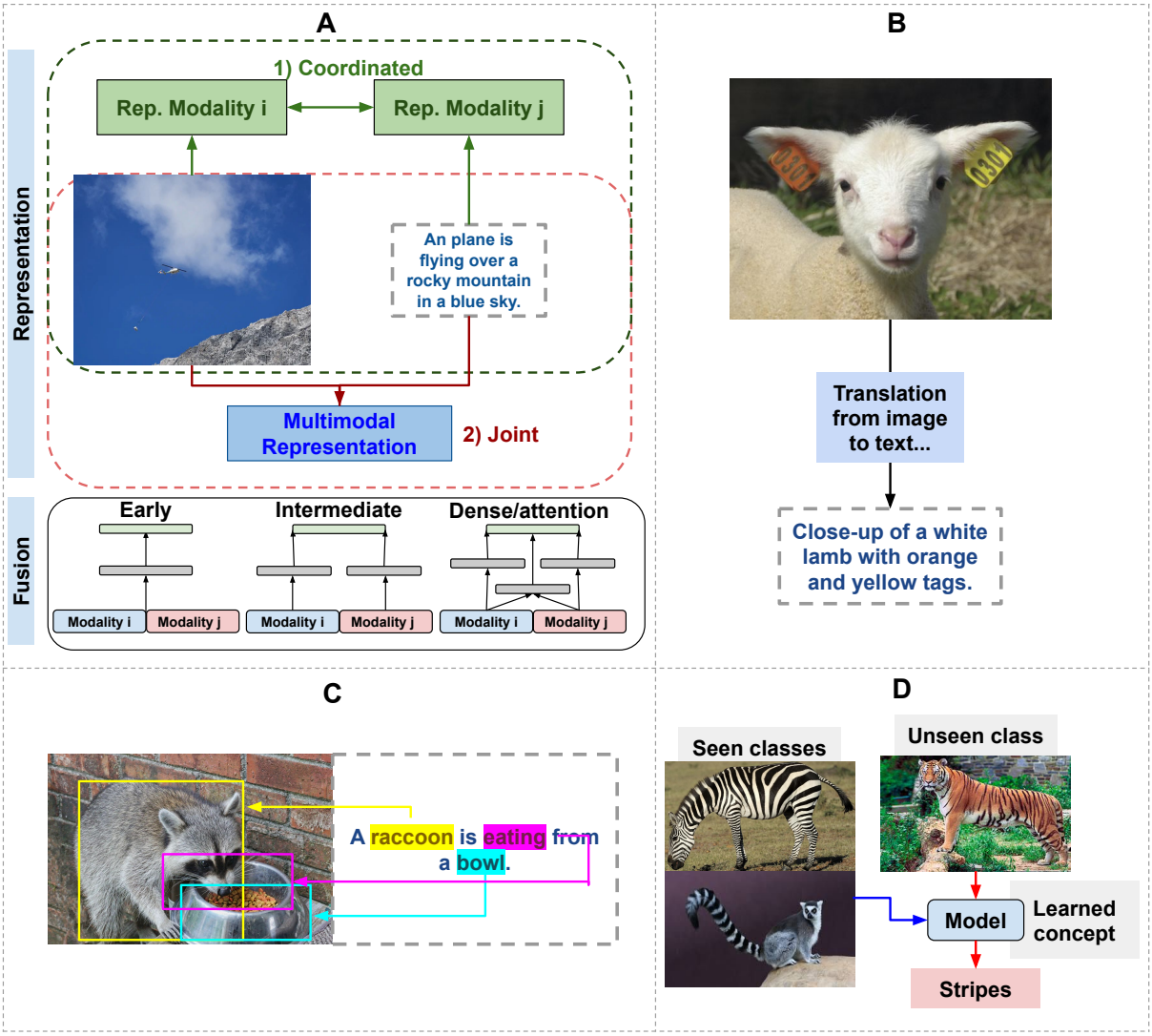



Figure 1.5: Multimodal challenges. A) Representation and fusion challenges. Representation is classified into two categories: 1) Coordinated representations aim to approximate each representation by using a similarity or a correlation measure, but separately. 2) On the contrary, the joint representation fuses the individual representations. Fusion is another challenge that studies each one of these different points in which the modalities are combined such as Early, Intermediate, or Dense (in which the layer of fusion can vary or be designed to perform as an attention mechanism); B) Translation deals with transforming one modality into another. One popular application is image captioning. Its goal is to provide a detailed semantic description of the image content; C) Alignment aims to find the relationships and correspondences between sub-components of instances from two or more modalities. The main barrier in this task is its requirement of high volumes of labelled data; D) Co-learning exploits the knowledge from one rich modality to model a poor one. For example, to predict semantic concepts to new or unseen classes, we can use information from classes seen during training. Once the model is trained, we can use it to transfer the semantic concept to unseen classes.

### 1.3.2 Multimodal applications

This thesis comprises the application of media description and multimedia retrieval with images and text. For media description, we address the tasks of image description [44, 49], and visual question answering (VQA) [10]. We also address multimodal retrieval mainly as a cross-modal retrieval task. We describe the addressed tasks next.

**Image description.** This is one of the most well-studied applications that connects computer vision and natural language processing (NLP). This application is in charge of generating semantic textual representations of a piece of visual data [44, 50, 51]. It has evolved from image annotation [52] to more complete descriptions such as image captioning [53, 54] and image alignment [55].

**Visual question answering (VQA).** VQA aims to answer questions presented in an image and using natural language [10, 56]. Recently, this task has branched out into more specific ones such as scene-text VQA that consider the semantic information conveyed by text within an image. This is because textual information is contained in about 50% of the images in large-scale databases as well as in our everyday surroundings [57, 58].

**Multimodal information retrieval.** Multimodal information retrieval aims to identify relevant data across different modalities. Specifically, the task of cross-modal retrieval has been a hot research topic in both computer vision and NLP communities. This is mainly carried on between images and text [46, 59, 60]. And the principal approach to address this task is to learn a joint semantic embedding space that can capture the inherent relationships between both modalities [61, 62] (see Figure 1.6). In the next section, we describe the methodology followed through our work in more detail, along with our objectives and principal contributions.

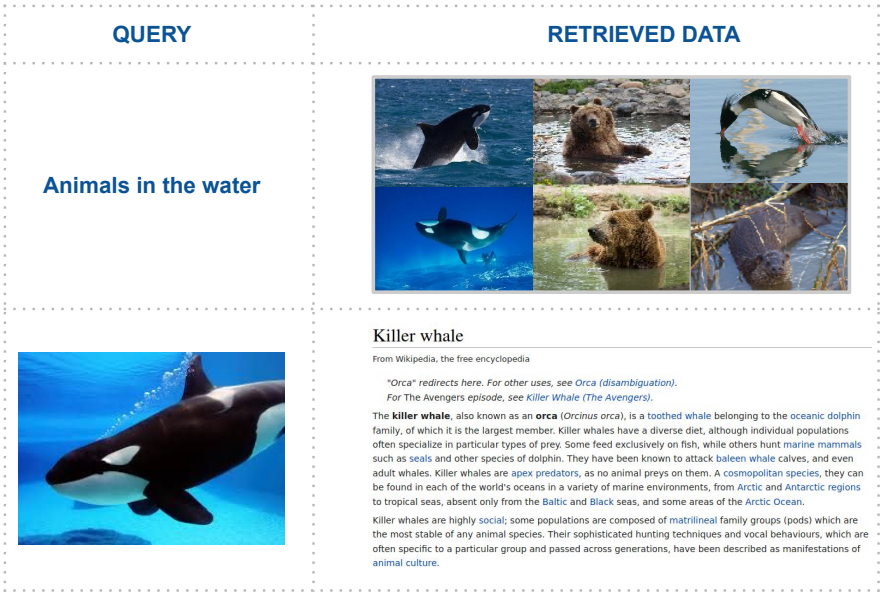



Figure 1.6: Examples of multimodal information retrieval. The first query is done using natural language and the expected results contain relevant and semantic visual representations. For the second query, the goal is to retrieve meaningful and semantic information related to a piece of visual information. For example, we can take a photograph of an unknown bird and look for textual information that describes the type of species.

## 1.4 Contributions and additional goals

As we explained in previous sections, multimodal learning comprises a large set of challenges and applications. Although developments in this area have achieved outstanding performance in different applications, research in this field continually grows as improvements in the precision of these systems are demanded. In this dissertation, we propose to tackle the problem of multimodal learning from images and text being the two principal data types. Our principal objective is to develop strategies that find a common semantic space that produces effective multimodal representations. We also aim to create approaches easily adapted and evaluated for several applications. The advantage of finding a common semantic space is that it allows us to easily perform comparisons between target text and visual content by mapping each modality to this space. This approach has been a successful strategy not only when working with images and text but in the combination of multiple modalities [63, 64]. In this section, we present generic goals pursued in the field of machine learning that are linked to our contributions. Next, we

present our contributions to the field of multimodal machine learning along with a description of the methodology adopted in each one. Finally, we present the organization of the document.

During the development of this thesis, we pursued generic goals that are relevant in the field of machine learning. These are linked to one or more of our main contributions. We listed them next.

- **To develop flexible systems that find a common semantic space between images and textual data.** This objective seeks to achieve the development of systems that explore the semantic relationships between images and text and that are easily adaptable to perform different multimodal tasks. This goal is linked to our first two contributions that carry out the tasks of multimodal information retrieval and scene-text visual question answering in which the proposed frameworks have similarities.

- **To explore/develop learning strategies that evaluate the impact of the quality of the data in the model performance when the problems of noise, imbalance, and insufficient labelled data are present**. Data is one of the most important factors when aiming to reach good performances. We determine that one relevant skill is to understand how to properly clean and analyze data and create strategies able to leverage it. In this thesis, we address frequent problems around the data, such as noise, imbalance, and when we have little annotated data available that are present in any machine learning task. This goal is linked to our third and fourth contributions where we first present strategies to tackle imbalanced (3rd contribution) and second by proposing a semi-supervised learning strategy (5th contribution). Semi-supervised learning permits harnessing the large amounts of unlabelled data available in many use cases in combination with typically smaller sets of labelled data [65]. This learning approach is the preferred solution to deal with machine learning problems because, in reality, there exist many data sets with little to no annotations.

- **To decrease the gap between multimodal machine learning tools and non-expert users.** There exist a lot of advanced models created and published that could potentially solve specific needs. However, it seems quite difficult to find and adapt them to extract value for different domains and users. Nowadays, bridging the gap between humans and

artificial intelligence is an objective for industries in many domains. Many artificial intelligent systems are more desired and implemented in everyday tasks. Therefore, workers are looking for a fast rapprochement with the academia that allows them to use these tools[9]. This goal is linked to our fifth contribution.

Next, we present a detailed summary of our main contributions made in this thesis in the field of multimodal machine learning.

1. In our first contribution, we developed an approach that evaluates the effectiveness of a multimodal system in a classical application that receives images and text and retrieves relevant multimodal information. With this approach, we can evaluate the ability of the model to produce effective multimodal representations and to handle any multimodal query with a single model. This framework is called Deep Multimodal Embeddings (DME) and is based on a deep learning (see Chapter 2 and Appendices A and B for a description of deep learning techniques). The results of our experiments showed that the DME model learns effective multimodal representations. This can handle any multimodal query with a single architecture while producing improved or competitive results in all retrieval tasks. Specifically, this performs well in the top results of the ranked list, which are the most important to users in the most-common scenarios of information retrieval. Our work represents a new baseline for a wide set of different methodologies for cross-modal retrieval (see Chapter 4). The results of this work are presented in the journal article *"Deep Multimodal learning for Cross-Modal Retrieval: one model for all tasks"* [66].

2. With this contribution, we demonstrate the flexibility of adaptation of the previously proposed model to carry on a more specialized application. Scene text visual question answering, ST-VQA, has been recently proposed as a new challenging task in the context of multimodal content description. The aim is to teach traditional VQA models to read the text contained in natural images (see Figure 1.1, Page 2 for examples of questions in which the answer is text contained in the scene). Here, we need to perform a semantic analysis between the visual content and the textual information contained in associated

[9] https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/the-analytics-academy-bridging-the-gap-between-human-and-artificial-intelligence (accessed Jan 2021)

questions to give the correct answer. We evaluated the relevance of different modules in our adapted framework using several experimental setups and baselines. We also exposed some of the main drawbacks and difficulties when facing this problem. We make emphasis and present alternative solutions to the out-of-vocabulary (OOV)[10] problem which is one of the critical issues of this task (see Chapter 5). The results of this work are presented in two published articles titled *"Semantic Text Recognition via Visual Question Answering"* [67] and *"An extended evaluation of the impact of different modules in ST-VQA systems"* [68].

3. With this contribution, we targeted the problem of imbalanced data at different levels. To evaluate our strategies, we consider the application of attribute learning. Attribute learning can complement category-level recognition and therefore improve the degree to which machines perceive visual objects. For example, in Figure 1.5 - D, Page 13, the attribute *Stripes* complements the information of a machine model because it transcends the category-level by defining through several classes what represent the concept in a semantic way. This application has been long studied as a classification problem [69] and recently, to study the problem of zero-shot learning that is the scenario where there is few annotated data. For all these reasons, we found this application very relevant to develop and evaluate our strategies around the quality of the data. We study the specific and common problem of data imbalance at different levels in real databases as most of the bad performance problems are due to the data itself. We propose adaptations to classical imbalanced learning strategies that cannot be directly applied when using multi-attribute deep learning models. Our strategies around classical imbalanced learning are designed to be used for multi-attribute deep learning models, i.e., multi-task or multi-label architectures with competitive performance in real databases (see Chapter 6). The results of this work are presented in the published article titled *"Multi-Attribute Learning with Highly Imbalanced Data"* [70].

4. With this contribution, we target the problem of having available few annotated data. We consider the problem of propagating attribute annotations from classes to single im-

[10] Words that are not part of the training text data but appear in the test set are called Out of Vocabulary words.

ages. We propagate as many correct labels as possible during training to create robust attribute classifiers by exploiting class-attribute relationships in a semi-supervised way. This allowed us to extend the annotations available in an image collection with diverse vocabularies without having to annotate individual images manually (see Chapter 7). This work is titled *"Learning To Assign Attribute Annotations To Images From Class-Level Relationships"* and currently is in the process of revision and submission.

5. With this last contribution, we intend to decrease the gap between deep learning tools and non-expert users. The developments proposed in this dissertation are intended to decrease the gap between deep learning tools and non-expert users. We demonstrate that with little to no imbalance, straightforward deep learning models work well and do not impose extra complexities. For non-experts, these models can be seen as black boxes, where all efforts are invested in pre-processing the data. Therefore, to simplify the problem, in our strategies we avoid using features that are costly to extract such as part or object localization which is widely used in the state of the art of attribute classification (see Chapters 6 and 7). These results are also depicted in the previously mentioned articles titled *"Multi-Attribute Learning with Highly Imbalanced Data"* [70] and *"Learning To Assign Attribute Annotations To Images From Class-Level Relationships"*.

### 1.4.1 Thesis organization

This document is organized as follows. Chapter 2 contains a description of background theory related to the tools used in this thesis. Chapter 3 contains the state-of-the-art works specifically related to the applications addressed in this thesis. Chapter 4 presents our proposed approaches to address the application of multimodal information retrieval along with the experimental setting and results obtained. Chapter 5 presents details about the application of scene-text visual question answering (ST-VQA) along with the proposed experimental setting and results obtained for two studies: first, an initial strategy to solve the problem of ST-VQA, and second, an extended evaluation of the impact of the different modules contained in an ST-VQA system. Chapters 6 and 7 present details about the application of attribute learning. It comprises the description of the experiment setting and results obtained for two specific tasks: multi-attribute

learning with different architectures, and learning to assign attribute annotations to images from class-level relationships. Finally, Chapter 8 presents general conclusions of the proposed developments and strategies and suggestions for future research.

# CHAPTER 2

# Background

## 2.1 Overview

Machine learning in the multimodal context aims to build models that can process and relate information from multiple modalities. One important part of any problem in multimodal learning is related to the representation of the data involved. Representation learning aims to find representations of raw and unstructured data as useful information to perform tasks such as classification or prediction. Good representations are important for the performance of machine learning models. While the development of uni-modal representations has been extensively studied, multi-modal representations still represent a challenge. In this chapter, we present the evolution and description from uni-modal to multi-modal representations. In this chapter, we perform a general review of the main representation models for language and vision separately, ending with the main approaches for multimodal learning for these two modalities. In Chapter 3 we present details of works in the state of the art in the applications addressed in this thesis.

## 2.2 Language representations

The book of Zhiyuan et al. [71] presents a recent and detailed explanation of the advances of representation learning techniques for NLP. Next, we briefly present a description of these techniques. The evolution of these models is also depicted in Figure 2.1.

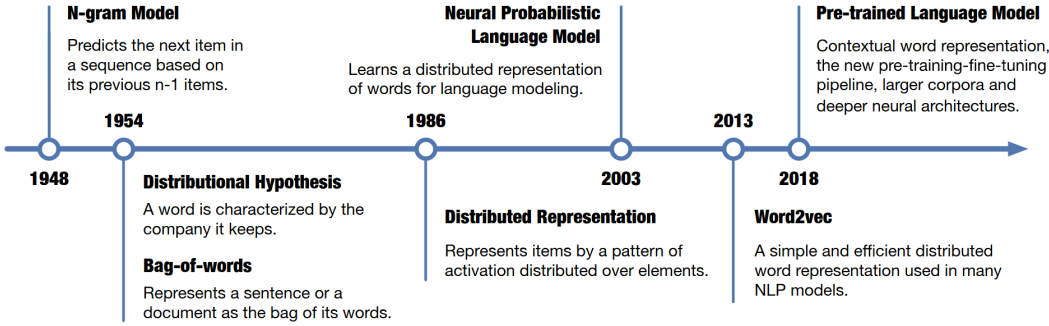



Figure 2.1: Timeline for the development of representation learning in NLP, extracted from [71].

**One-hot vectors.** The easiest and first way to represent a word is a one-hot vector representation. This vector has the dimension of the vocabulary size and assigns 1 to the corresponding position of the word and 0 to others [72]. However, this representation has a lack of semantic information.

**Bag-Of-Words (BOW).** In these types of models, a document can be represented as a bag of its words, disregarding their order. Then a score or weight can be computed for each word based on the number of occurrences [73].

**N-grams.** N-gram models learn vectorial representations for each word belonging to a given vocabulary. The representations are trained to predict words appearing within the context window of a given center word. This model incorporates information about the structure in terms of n-gram embeddings. To predict the next word in a sequence, some previous words (and in the case of n-gram, they are the previous n-1 words) are considered [74]. This type of approach introduces the core idea of some of the most popular models in NLP, "a word is characterized by the company it keeps" [71].

**Neural probabilistic language models (NPLM).** These approaches use a combination of a vector representation and a neural network. It first assigns a distributed vector for each word, then uses a neural network to predict the next word. This strategy has successfully learned to model the joint probability of sentences bringing some quality encodings and semantic meanings to the words. Some of the most popular embeddings using this strategy include Word2Vec [75], GloVe [76], and Fasttext [77].

**Pre-trained language models.** Finally, the latest models take into account a deeper context of the text into consideration. The key idea is to generate dynamic representations for the words based on their context instead of generating a fixed one. This has been very useful for words with multiple meanings. These models are also called pre-trained language models because they require large amounts of data to be trained, and therefore, the best strategy is to train them over a large corpus of data and use it for the final target application. The most popular models using this approach include for example ELMo [78] and BERT [79].

**Recurrent neural networks (RNN).** As we have shown, there have been several techniques to represent and study the natural language. Some of the most representative ones are based on deep learning, specifically using recurrent neural networks. Recurrent neural networks (RNNs) and their variants have been the solution to most problems dealing with sequential data and natural language as they can process sequences and lists. RNNs can learn to use past information. It consists of several repeated sub-networks each one passing a message to a successor. This allows the information to persist throughout the network. Long short-term memory networks (LSTM) are a special variant that emerged to tackle the problem of very long-term dependencies, i.e, LSTM can remember the information for long periods of time [80]. The most essential idea is to include a type of structure called *gates* that are a way to optionally let information through (see Figure 2.2).

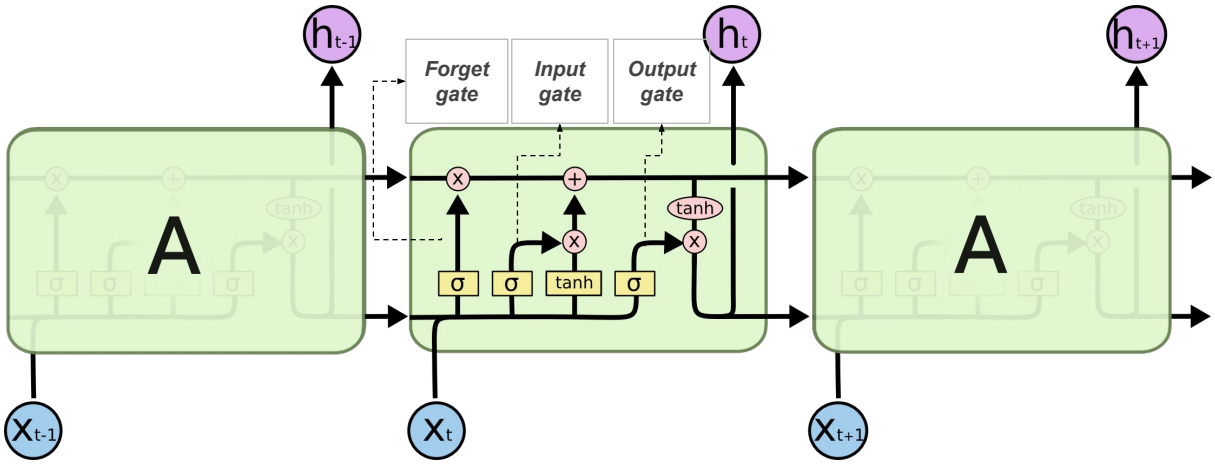



Figure 2.2: The basic structure of an RNN, specifically, an LSTM. These models contain a chain of repeating modules of neural networks and include the concept of gates that control the flow of information. In the graphic, A represents a cell (or chunk) of a neural network; X represents the input sequence, h represents the outputs. Each cell contains three basic gates to protect and control the cell state and with different activation functions been applied (such as sigmoid or tanh functions). The forget gate decides what information not to store; the input gate decides what information is updated; finally, the output gate that filters the information based on the current state of the cell. Figure partially generated from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Jan 2021).

## 2.3 Visual representations

There have been three principal methods to represent visual data, and specifically, images. These are (1) local and global descriptors, (2) bags-of-visual-words and Fisher vectors, and (3) representations based on neural networks that have been the most successful and widely used [81]. We describe them next.

**Local and global features.** Local features (also called descriptors) represent patterns in a region that differs from its immediate neighborhood. These can be points, edges or small image patches representing key points in the image [82]. Global descriptors are composed of features such as intensity, textures, and color histograms computed on the entire image. Therefore, the image is represented by a single feature vector. Some of the most popular descriptors in

this category include SIFT, GIST, and SURF for local descriptors and invariant moments and histogram-oriented gradients (HOG) for global descriptors [83].

**Bags-of-visual-words (BOV) and Fisher vectors.** These models represent an image by extracting a set of small image patches, quantizing them into a finite number of prototypes also called visual words, and finally, building a histogram of visual word occurrences. The Fisher vector representation can be seen as an extension of the BOV model. They extract a set of local patch descriptors to encode them into a high dimensional vector [84].

**Convolutional neural networks (CNN).** CNN can be used to learn image representations that can be used for several tasks. The main difference with other neural network architectures is that a CNN can successfully capture spatial and temporal dependencies in an image through the application of relevant filters (see Figure 2.3). The filters allow us to capture low-level features very desired in images such as edges, color, gradient orientation. This filter operation is carried on by using convolutional operations. Through several applications of these filters (known as convolutional layers), the network can learn feature maps that summarize the presence of detected features in the image. The most successful architectures based on this network structure include AlexNet [85], VGGNet [86], and ResNet [87] (see Appendix A for details in the theoretical foundations of these models).

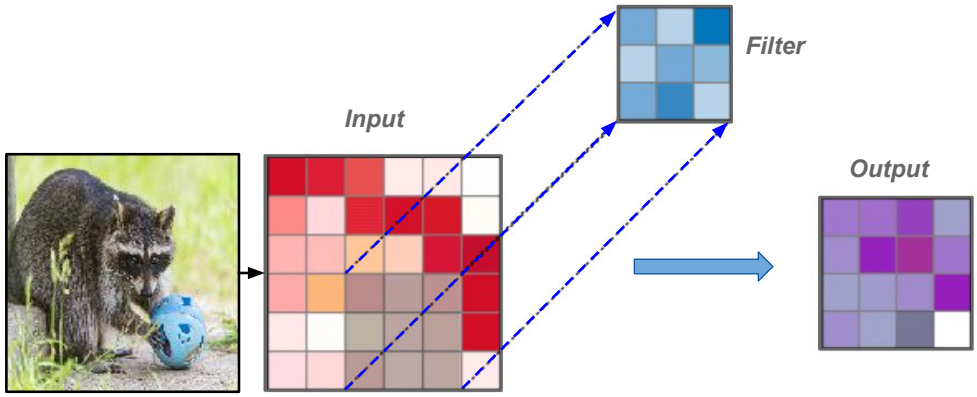



Figure 2.3: The convolution layer uses filters that perform convolution operations while scanning the input.

## 2.4 Multimodal representation

Multimodal representation learning is an active research field that still represents a challenge. Some of the most popular models such as graphical, sequential, and especially artificial neural networks (ANNs) have been shown to have advantages when representing multimodal data through a joint combination of uni-modal representations [2, 88]. We study this challenge and evaluate the quality of the representations through several multimodal applications (see Chapters 4, 5, 6 and 7).

### 2.4.1 Graphical models

There exists a family of popular graphical methods for multimodal fusion. The majority of these models have been used for sequential data. Some of the most popular include hidden Markov models (HMM), dynamic bayesian networks (DBN), and conditional random fields (CRF). HMM are probabilistic models based on augmenting the Markov chain. A Markov chain is a model that tells us something about the probabilities of sequences of random variables (states), each of which can take on values. Such a chain is useful when we need to compute a probability for a sequence of observable events that may be hidden: we cannot observe them directly [89]. This model was originally applied to speech tagging because it allows us to talk about both observed events like words that we see in the input and hidden events like part-of-speech (POS) tags that we think of as causal factors in our probabilistic model [90]. For text data, it has been used as a summarization technique [91], [92]. The second type of model, DBN, uses Bayesian inference for probability computations. These networks aim to model conditional dependence, and therefore causality, by representing conditional dependence by edges in a directed graph [92]. Finally, CRF comprises a set of popular discriminative undirected probabilistic graphical models that can represent relationships between different variables. This is usually used for the labelling problem and therefore, the previous context is required when making predictions on a data point. Data points are arranged as a graph consisting of a set of nodes, and edges. An edge between node $i$ and node $j$ denotes a dependency between them. For example, in image segmentation, the class label for the pixel depends on the label of its neighboring pixels. CRFs are used in sequential data processing such as POS tagging in NLP

and image segmentation in computer vision [93, 94].

### 2.4.2 Kernel-based methods: Multiple kernel learning

Kernel-based methods represent a well-established learning paradigm whose main idea is to capture the nonlinear patterns behind data. Data such as images may not be easy to classify in a 2-dimensional space because of their non-linear nature. Hence, we need to find a function that maps our data into a higher space, where we can easily find a clear separation between different classes. However, to operate the data in a higher-dimensional space represents a tremendous computational resource consumption. The "kernel trick" was proposed to tackle this problem. This allows us to operate in the original feature space of the data without computing the coordinates of the data in a higher-dimensional space. In other words, this offers a more efficient and less expensive way to transform data into higher dimensions [95]. Choosing the most suitable kernel function with the right parameters and regularization is of great importance. Popular kernel-based methods include support vector machines (SVM) [96], kernel canonical correlation analysis (KCCA) [97], and matrix factorization [98]. In the context of multimodal learning, operating different kernels may correspond to using information coming from multiple sources. These tools fell slightly out of favor when deep learning approaches became popular. However, they may be useful when dealing with small databases [99, 100].

### 2.4.3 Artificial neural networks

The first works related to ANNs dates from the year 1943 when Warren McCulloch, a neurophysiologist, and a young mathematician, Walter Pitts, wrote a paper on how neurons might work. Popular models ensued [101, 102] such as CNNs [103] and recurrent models such as LSTM [80]. Since then, these models have been extensively used when solving many complex real-world problems. Figure 2.4 shows how the interest in models based on this structure has escalated through time. ANNs are inspired by the human nervous system that consists of billions of neurons of various types. These neurons interact and communicate through multiple and simultaneous signals [101]. In a simplified way, ANNs are representations of a system of neurons. Each neuron receives inputs from several other neurons, multiplies them by assigned

weights, adds them, and passes the sum to one or more neurons. An activation function to the output before passing it to the next variable may be applied as well [102].

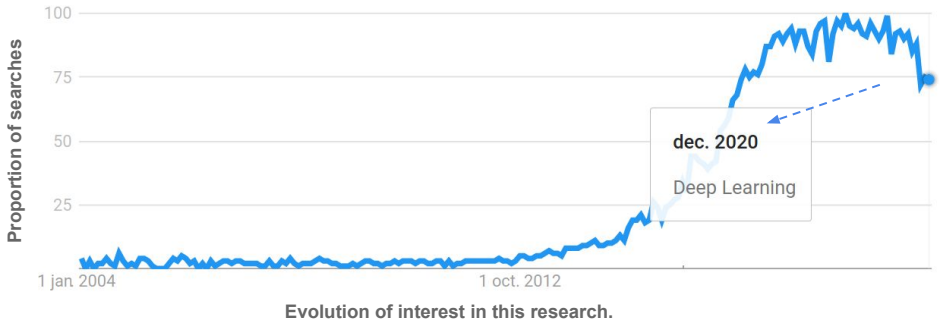



Figure 2.4: The results reflect the proportion of searches for the keyword "deep learning" in the specific region of "all the countries" and period (01 Jan 2004 - 31 Dec 2020), compared to the region with the highest usage rate for this keyword (value of 100). For example, a value of 50 means that the keyword was used half as often in the affected region, and a value of 0 means that there is insufficient data for that keyword. Figure partially generated from the source data at https://trends.google.com/trends/ (accessed 14 Jan 2021).

ANNs are in general composed of an input layer, which receives data, one or several hidden layers connected that process the data, and an output layer that provides one or more data points based on the function of the network. Figure 2.5 details an example of an ANN model after the training process. In the example, the network is composed of the input layer in which we can include different features or representations for the input variables; 3 hidden layers with dimensions of (4, 4, 2) respectively (i.e., the number of neurons in each layer); and one output layer with two neurons representing each class. The structure of the network is known as model architecture. An ANN learns by an optimization process until a criterion is reached (see Section A.1.3). In the example, we showed that the network is effectively able to learn to discriminate between the two classes for the training (softer data points) and test data (hard data points). Some popular ANNs are perceptrons. The perceptron is the basic unit of deep learning models. It takes several binary inputs and produces a single binary output. Hence, a deep learning model is a network of perceptrons [104]; multi-layer perceptrons (input layer, a set of hidden layers and an output layer of neurons) [105]; and auto-encoders in which the general idea is to reconstruct the input (the output values equal to the input values) [102]. Appendix A describe the theoretical

foundations for the use of deep learning models.

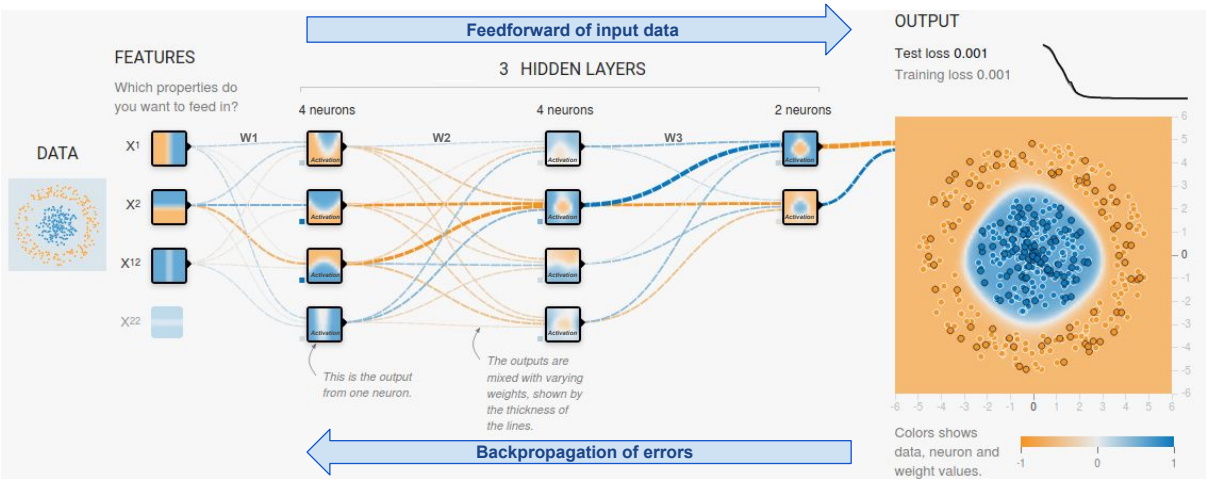



Figure 2.5: Structure of an ANN model. A traditional ANN consists of three parts: an input layer that corresponds to the target data to study or a feature representation of it; a set of n hidden layers (in the example, 3); and the output layer. The most important step in the training phase is the optimization process. This comprises two general phases: a feed-forward step to pass the input data $X$, and a back-propagation step to find the optimal set of parameters $W$. To carry on these processes, it is required to establish activation functions (applied to the outputs of each neuron), loss functions (that represents the objective function to minimize), etc. In the example, the objective is to classify the input data into two classes (positive and negative). The output shows the result after the training process with an effective classification (or separation) of the two classes for the training (softer data points) and testing (hard data points) data. Figure partially generated with the tool developed by Daniel Smilkov and Shan Carter, https://playground.tensorflow.org (accessed Jan 2021).

## 2.5 Discussion

In this chapter, we have presented the evolution of the representation models involved in uni-modal and multimodal learning. We reviewed the main representation models for language and vision (images) separately, ending with the main approaches to multimodal learning for these two modalities in the state of the art. While uni-modal representations have been studied extensively and performance in different applications is outstanding, for multimodal applications this is not yet the case. The semantic gap between the different modalities imposes a barrier

when it comes to jointly analyze the information contained in them.

In this thesis, we approach the specific problem of multimodal learning with images and text from various perspectives. In our strategies, we make use of ANNs as the main structure of our approaches due to their success in different multimodal applications (see Chapter 3). Additionally, we explore multimodal learning from various multimodal applications to gain a better perspective on the performance of the proposed strategies. These strategies are presented in Chapters 4, 5, 6 and 7. In the next chapter, we present the state of the art of approaches that address the problem of multimodal learning using the visual modality in the form of images and the textual modality. Specifically, we present the state of the art related to the applications addressed in this dissertation (see Section 1.3.2), which are cross-modal information retrieval, scene-text visual question answering, and attribute learning (image description).

# CHAPTER 3

# Related work

## 3.1 Overview

In this chapter, we present the state of the art of tasks addressed in this thesis. This includes cross-modal information retrieval (Section 3.2), scene-text visual question answering (Section 3.3), and attribute learning (Section 3.4). We describe the current state of the art in each one of these applications, finalizing with a summary of our proposed strategies that will be presented in the following chapters.

## 3.2 Cross-modal information retrieval for images and text

In this section, we present the state of the art for works addressing the problem of cross-modal retrieval. After the review of several works, we classify the principal approaches into two categories: works based on finding a sub-space and works based on deep learning. In turn, each category contains a classification according to the type of strategies followed. This classifica-

tion is also supported by the survey works presented in [59, 106]. Although the classification is larger in the survey works, we selected the most relevant approaches that address the task of cross-modal retrieval and that are implemented for the visual (images) and textual modalities. For this task, the most general performance metric used is mean average precision (see Appendix A.1.4 for a definition of this measure). We include some of the main results reported in the works for a general perspective of their performance. The goal of this task is to take one type of data as the query to retrieve relevant data objects of another type. The most popular types used to perform the cross-modal task are images and texts corresponding therefore to two cross-modal retrieval tasks: 1) Image query vs. text database (Img2Txt), 2) Text query vs. image database (Txt2Img).

The most popular databases contain natural images and textual data such as tags, sentences, or paragraphs that contain descriptions of the image content. Wikipedia retrieval database (WRD) [107] is the most popular database for cross-modal retrieval tasks. It contains image-document pairs corresponding to Wikipedia articles[1], classified in a very diverse range of domains such as art, biology, geography, history, etc. Another database, Pascal sentences [108], was created by using the platform Amazon's Mechanical Turk [109] and provides pairs of images and a set of up to 5 descriptions of their content provided by the users of the platform. MIR-Flickr-25k [110] consists of images downloaded from the social photography site Flickr and annotated using different semantic concepts. Other databases less popular for the task of retrieval include Nuswide [111], XMedia database [112], and MS COCO database [113].

### 3.2.1 Sub-space learning

Sub-space learning comprises techniques that aim to find a set of projections for diverse modalities such that the correlation between them is maximized. Figure 3.1 shows an example of a framework and its components for typical approaches implementing this technique. The principal idea is to learn a mapping function to project data from different modalities to a common space so that similarities between them can be directly measured.

[1] https://en.wikipedia.org/wiki/Wikipedia:About (accessed Feb 2021).

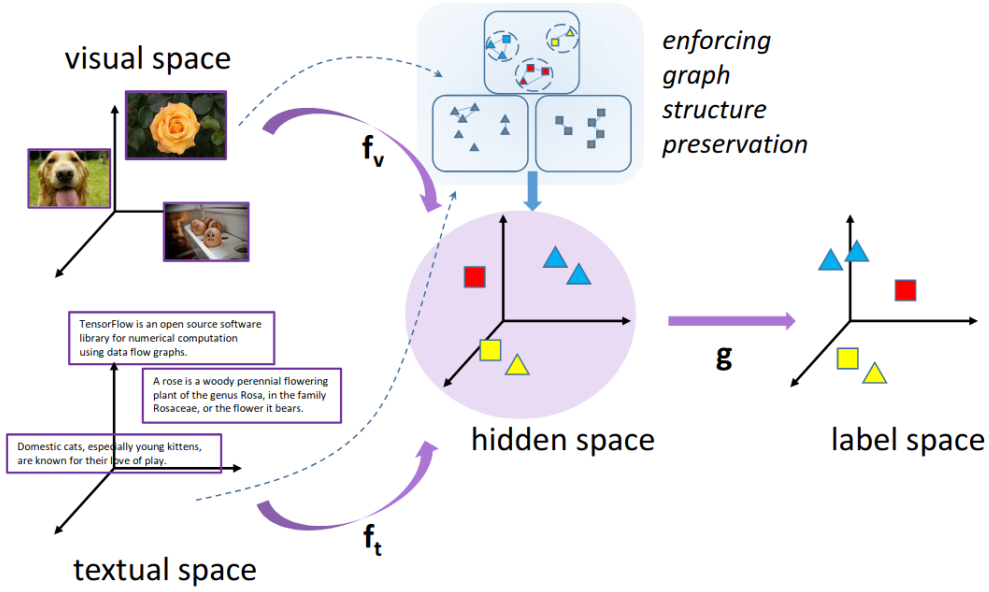



Figure 3.1: Example of a framework that implements the strategy of finding a common multi-modal sub-space (extracted from [114]). The inputs are features of each modality residing in its own space; the f functions represent the target projection functions to learn, and that allows to transform inputs from their own spaces to a hidden common space; this hidden space can be also constrained by supervision such as labels and therefore, it contains the semantic structure-preserved and can be used for cross-modal retrieval tasks.

**Correlation models.** Several of these techniques are based on the commonly known canonical correlation analysis (CCA) [115] algorithm [116]. These techniques benefit from explicitly modeling the correlation between two elements. They rely on the strong assumption that the modeling is more useful in feature spaces having higher levels of abstraction such as the visual one. Features such as SIFT for images and latent Dirichlet allocation (LDA) [117]) for text were widely used as the base for these models. Firstly, both visual and textual features are projected onto a latent space using the CCA method, and then the probabilistic interpretation of CCA is utilized for calculating the representative distribution of the latent variable for each class. The general formulation of the optimization problem is presented in Equation 3.1 where the goal is to maximize the correlation between the textual and visual modalities.

$$J = arg\ max_{W_I, W_T}\ corr(W_I^\top X_I, W_T^\top X_T) \tag{3.1}$$

Where $X_I$ and $X_T$ are matrices containing corresponding features from images and text and, $W_I$ and $W_T$ are the target transformation functions that project the two modalities onto the shared space. Popular variants include correlation matching (CM), semantic matching (SM), and semantic correlation matching (SCM). CM is an unsupervised approach that models cross-modal correlations, SM is a supervised method that relies on semantic representation and SCM is the combination of both of them [118]. There exist many problems around these traditional techniques such as the waste of information when applied on supervised problems as it is an unsupervised technique; and second, it is not able to compute non-linear correlations more present in real applications. Therefore, variants to address these problems have appeared such as Kernel CCA [119] and mostly, several others based on deep learning implementations such as encoder-decoder architectures [120] or more deep architectures [121] (see Section 2.4.3 for a description of the architectures). Some representative MAP results for the WRD include (Img2Txt - 0.306, Txt2Img - 0.266) [122], (Img2Txt - 0.39, Txt2Img - 0.29) [118], and (Img2Txt - 0.272, Txt2Img - 0.232) [120].

**Projection matrices.** In this category, we can include methods based on tensor factorizations and graph embeddings. Tensor computations are better found as matrix factorization models, in the case of two modalities [62], that find projection matrices (PM) to map the data to an embedding space. This space may be restricted by regularization and constraint terms [123]. Some problems of these models are the complexity created when non-linear mappings are included (kernel functions), which leads to higher computational requirements without significant improvements in their performance [114, 124–126]. Non-linear mappings change linear projection functions $W$ to $\Omega$ projection functions in Equation 3.1 where the non-linearities of the data are easily modeled [127]. Graph approaches aim to ensure the intrinsic geometric structures of different feature spaces. Their approaches can use supervision spaces such as labels to model the correlations among different modalities (MAP in WRD: Img2Txt - 0.338, Txt2Img - 0.406) [128].

### 3.2.2 Deep learning models

As in many applications, methods based on deep learning have become the preferred approach to tackle the problem of cross-modal learning [46, 60, 129–132]. These methods are designed to reflect the local and global individual structures from different modalities and make the resulting embeddings useful for a variety of tasks. Figure 3.2 presents a generic framework of a deep learning model for multimodal retrieval. The main idea is to combine different networks depending on the nature of the data, for example, CNN for the images + fully connected networks (FC) for the text, or represented by the same type of network such as CNN [133], with multi-modal layers with single [134] or multiple connections [135]. The textual modality can be used to guide the process of learning semantic features, i.e., this is used to supervise the model [136]. Supervision is an important factor and therefore, loss restrictions are widely applied to improve the quality of the semantic space generated. In [137], the authors propose a method with novel within-modality losses that aims to improve the semantic coherence in both the text and image sub-spaces. Images and textual embeddings are close in the joint space only if they are in the textual modality in its uni-modal text space.

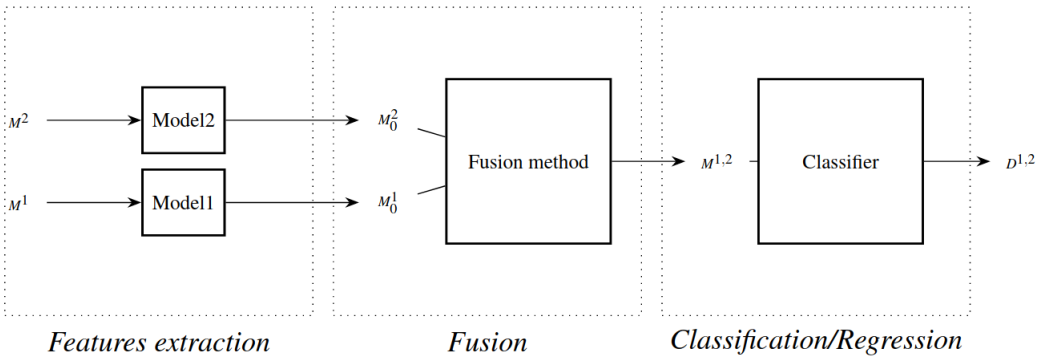



Figure 3.2: Example of a framework that implements the strategy of finding a common multimodal sub-space using a deep learning approach (extracted from [135]). The features inputs of each modality are computed using different models; then a central component is used to fuse these features and finally, a classification or regression task to optimize the model.

Although supervision is quite important, any type of supervision data can be difficult and expensive to obtain. One critical aspect is to create models able to perform without demanding huge amounts of annotated data. Recent works are presenting solutions for the problem of insufficient

annotated data such as how to train the model with minimal supervision, where there do not exist pairs of data for all samples also by incorporating composed terms in the loss [138] such as in Equation 3.2.

$$Loss = Loss_{MS} + Loss_{MI} + Loss_{DIS} \tag{3.2}$$

where the $Loss_{MS}$ represents the modal-specific term and it is in charge of finding representations for each modality separately. The $Loss_{MI}$ represents the modal-invariant term in charge of imposing similarity between each sample pair across modalities. And finally, the $Loss_{DIS}$ discriminative term that uses the class labels as the supervision information and is represented as a common classifier for the different modalities (WRD: Img2Txt - 0.381, Txt2Img - 0.35). Other works take advantage of raw data in databases such as descriptions and captions in the form of topics [139, 140] or propose strategies based on zero-shot learning (ZSL), whose main purpose is to use very few annotated data [141].

**Popular variants.** There exist many variants when implementing deep learning models for cross-modal tasks [129–131]. This can be related to the number of points of connections or the type of model used to initially encode the modalities involved. These models for multimodal learning are usually composed of two (or more) sub-networks, one per modality [46, 132]. The connections of these sub-networks can be performed at several points [135, 141], or can represent forms of attention mechanisms [142]. The goal of the attention mechanism is to generate an attention mask that focuses on relevant data in the feature representations. In [143], they show how the performance increase when the attention is applied to both modalities (MIR-Flickr-25k: Img2Txt - 0.772, Txt2Img - 0.807). The type of model used to embed each modality can also vary. For example, the visual part is passed through a neural network, and the textual part is passed through a model such as a BOW, and a late fusion component is used to combine both of them. We found many variations especially for the method used to represent the textual modality. Some of the combinations found in the state of the art include CNN + BOW [144], CNN + LDA [139, 145], CNN + kernel methods [146], and CNN + FC [147]. Another type of variant includes the use of restricted Boltzmann machines (RBM) [148, 149]. These are undirected graphical models that define a probability distribution of the generated features of different modalities using shared hidden layers. These are usually designed as a set

of stacked models (see Figure 3.3). For example, in [150], they propose three stacked models: a model to learn a modality-friendly representation, whose statistical properties are similar, and the modality-mutual representation, which contains some missing information in the original input instances. A second model is stacked and contains a joint auto-encoder and a three-layer feed-forward neural net to obtain the hybrid representation. And finally, a third model stacked over the first two, that obtains a shared representation for each modality by implementing bi-modal auto-encoders. Some representative MAP results for deep learning models for the WRD are (Img2Txt - 0.428, Txt2Img - 0.374) [146], and (Img2Txt - 0.434, Txt2Img - 0.388) [145].

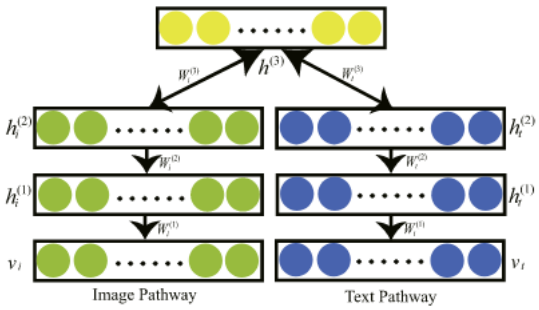



Figure 3.3: Example of stacked models (extracted from [150]). The model represents specifically a deep belief network (DBM) and its goal is to learn the common representation of various modalities.

**Adversarial learning-based models.** Adversarial learning is a research field concerned with the analysis of models to adversarial attacks, and the use of such analysis in making the models more robust to attacks. This learning strategy simultaneously trains at least two models: a generative model that captures the data distribution and a discriminative model that estimates the probability that a sample came from the training data rather than the generated one [151]. These models can be seen as different networks. Figure 3.4 shows an example of a framework following this type of learning strategy. For cross-modal retrieval, these methods explore how to jointly extract and utilize both the modality-specific and modality-shared features effectively [61].

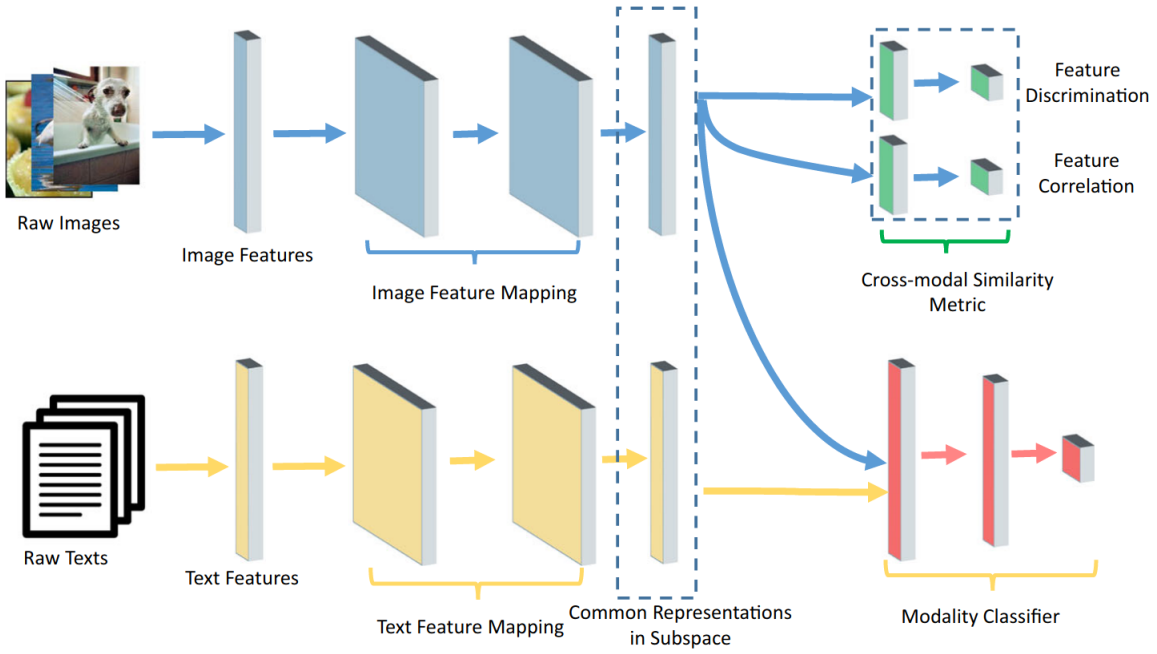



Figure 3.4: Example of a framework that implements an adversarial learning strategy (Figure extracted from [152]). The framework consists of some major components such as image and textual feature projections to a common sub-space, and two major processes to optimize. In the example framework, the processes are represented by two networks targeting two tasks: a modality classifier, and a cross-modal similarity metric. The adversarial learning manner is adopted to jointly optimize the networks during training.

The processes followed are as follows: first, a feature projector tries to generate a modality-invariant representation in the common sub-space and to confuse the other process, modality classifier, which tries to discriminate between different modalities based on the generated representation [153]. This can be seen as generators projecting the different modalities into a common and a discriminant space, while the discriminators compete against the generators to alleviate the heterogeneous discrepancy in the space [154]. Other works include more than two processes. In [155], the authors propose three training processes seen as different paths. These include a multi-modal feature embedding path, an image-to-text generative feature learning path, and a text-to-image generative adversarial feature learning path (see Figure 3.5). Because of the success in different tasks, several works in cross-modal retrieval apply now this approach [61, 129, 152, 156–160].

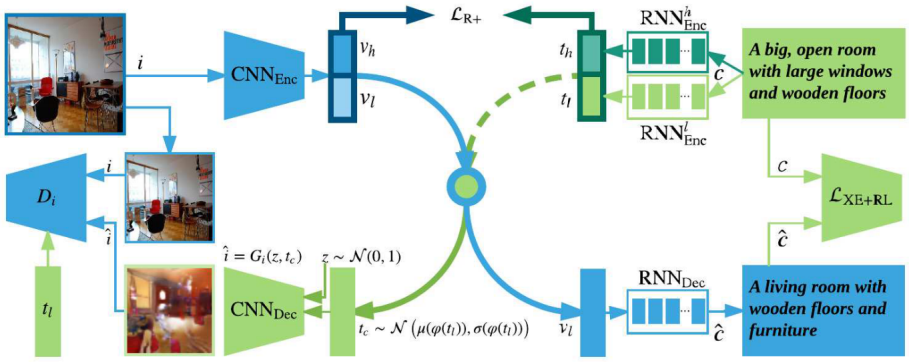



Figure 3.5: Example of a framework with multiple training paths (extracted from [155]). Their proposed generative cross-modal learning framework (GXN) consists of three training paths: cross-modal feature embedding (the entire upper part), image-to-text generative feature learning (the blue path), and text-to-image generative adversarial feature learning (the green path). It includes six networks: two-sentence encoders in dark green and light green, one image encoder in blue, one sentence decoder, one image decoder, and one discriminator. This number of models to train also represents a huge demand for computational resources.

The most recent developments aim to tackle the problem of insufficient annotated data with different approaches [141, 161]. For example, in [162], they present a framework with the different processes to optimize aiming to transfer data among modalities. The first process, a modal-sharing knowledge transfer sub-networking, aims to jointly transfer knowledge from a large-scale single-modal database in the source domain to all modalities in the target domain with a star network structure. The second process, modal-adversarial semantic learning sub-network, is proposed to construct an adversarial training mechanism between common representation generator and modality discriminator to enhance cross-modal semantic consistency during the transfer process. Some representative MAP results are Pascal sentences: (Img2Txt - 0.281, Txt2Img - 0.261), WRD: (Img2Txt - 0.331, Txt2Img - 0.287) [156], and for MIR-Flickr-25: (Img2Txt - 0.772, Txt2Img - 0.8001) [157], and (Img2Txt - 0.741, Txt2Img - 0.756) [159].

**Hash-based models.** Most recent methods follow another learning strategy when addressing the problem of cross-modal retrieval. They transform the problem of finding common multi-modal representations to the learning of hash codes representing embeddings for differ-

ent modalities. The goal of hashing is to map the data points from the original space into a hamming space of binary codes where the similarity in the original space is preserved in the hamming space. In this sense, the binary codes are learned as a way of regularization. The cross-modal hashing aims to learn a common hash matrix $H\epsilon\{-1,+1\}^{l*n}$ of dimension $l$ and computed by $n$ samples, as depicted in Equation 3.3.

$$min_H(\|H-F\|_F^2+\|H-T\|_F^2) \tag{3.3}$$

where $F$ and $T$ represent the feature matrices for the image and text modalities. Figure 3.6 shows an example of a framework implementing this learning strategy. These methods are based on binary supervision, considering multi-modal relationships as either completely similar or completely dissimilar. By using binary hash codes to represent the original data, the storage cost can be dramatically reduced.

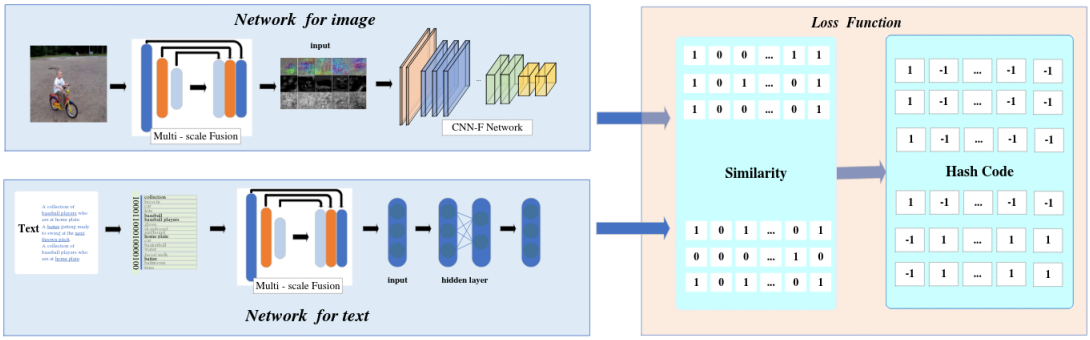



Figure 3.6: Example of framework that learn hash codes for cross-modal retrieval (Figure extracted from [163]).

This is one reason why hashing-based methods have become more and more popular for neural networks search in large-scale databases. For cross-modal retrieval, this is one desired advantage and it is implemented in several state-of-the-art works [164–170]. These models are mainly based on deep learning [171–178], but can also use matrix factorization [179, 180] or a combination of methods [181]. The most important aspect of these methods is the importance of the code length that has a significant influence on the final result [163]. Usually, larger codes obtain better results [182]. Most of these methods use supervision such as labels. For example, frame-

works that use neighborhood information from different modalities to create a joint-semantic affinity matrix. They explore semantic relations among labels or they use the semantic label data to improve the feature learning part, which then can preserve semantic information of the learned features and keep the invariability of cross-modal data [177, 183, 184].

Many variations using this approach include graph CNN [185], deep RBM [148], and PM [179, 186]. As with previous approaches, the challenge is in how to learn with insufficient data, and some unsupervised methods have been proposed [187–189]. The biggest variants implementing this strategy are models based on adversarial learning [178, 190–193]. This is due to the fact of the two inherent processes to optimize: a process that can be based on supervision such as labels, and the process in charge of learning the hash codes [60, 194–196]. One shortcoming of these hand-crafted feature-based methods is that the feature extraction procedure is independent of the hash-code learning procedure which is one of their main problems [197]. Some representative MAP results for WRD are: (Img2Txt - 0.279, Txt2Img - 0.626) [165], (Img2Txt - 0.374, Txt2Img - 0.709) [166], and (Img2Txt - 0.331, Txt2Img - 0.701) [195]; while for MIR-Flickr-25k: (Img2Txt - 0.699, Txt2Img - 0.812) [166], (Img2Txt - 0.721, Txt2Img -0.768) [195].

### 3.2.3 Discussion

Cross-modal information retrieval is a challenging task due to the semantic gap between the modalities. Due to this gap, we cannot compare different modalities directly with each other. Figure 3.7 shows some example results extracted from very recent works for the two most popular retrieval tasks Img2Txt and Txt2Img for Nuswide and WRD databases. We can see that the task is still very challenging, in special when the query is an image and we aim to retrieve some relevant textual content.

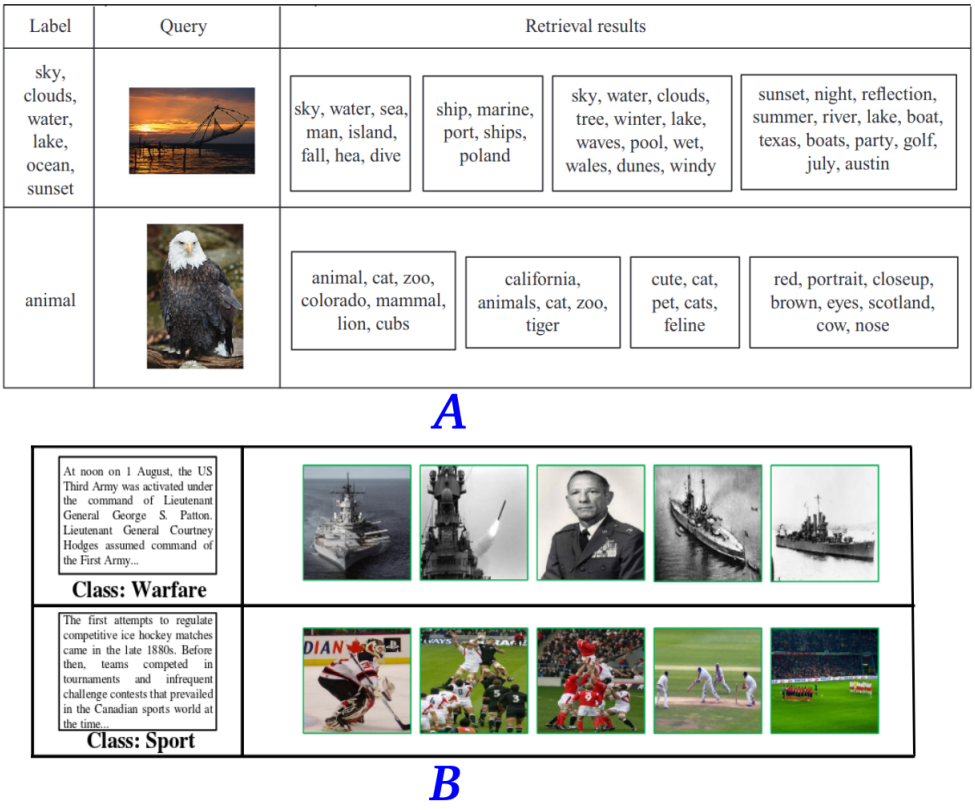



Figure 3.7: Retrieval example results for the two most popular cross-modal tasks: A) Img2Txt on the Nuswide database (extracted from [184]). B) Txt2Img on the Wikipedia retrieval database (extracted from [162]).

In this section, we present a set of different strategies that aim to tackle the problem of finding a common semantic space where modalities can be compared as equals. The most popular are based on deep learning. The most popular deep learning variations are adversary models and hash-based models. Adversary learning has been very successful in many applications, however, there are also downsides to working with this type of learning strategy. For example, the additional complexities and computational burden due to multiple processes to optimize. Some works have shown how its performance is highly correlated with the characteristics of the data set, and data points that are far enough away from the training data distribution can damage it [198]. On the other hand, the main problem that hashing methods for multi-modal retrieval need to solve is to minimize it as much as possible, and deep hashing methods usually perform better in case of a colossal amount of multi-modal data but with higher hardware cost. Also, the full use of tag information has important significance in the final performance. Because of

the hash collision problem (producing the same hash value), longer hash codes are needed to ensure the precision of the retrieval, which brings additional time and computational overhead, and leads to a decrease in the recall rate [199].

Despite all these disadvantages, deep learning models are successful in extracting and modeling the intrinsic characteristics of different modalities and are the preferred base method for cross-modal retrieval tasks. In Chapter 4, we explore the general problem of multi-modal retrieval for databases containing images and textual data. Inspired by visual question-answering architectures, our approach learns combined representations to build an effective multi-modal retrieval system. Unlike recent models such as adversarial, or models based on hashing that requires the optimization of many models (up to six models to train [155]), we propose to train a simplified end-to-end model that demonstrates the capabilities and that support our pursued generic goals and contributions: 1st and 5th contributions (see Section 1.4). Also, performance results for WRD such as (Img2Txt - 0.279, Txt2Img - 0.626) [165], and (Img2Txt - 0.331, Txt2Img - 0.701) [195] appear to be over-optimized for the task of Txt2Img while having very low performances for Img2Txt. Our goal is to investigate the potential of a single model to retrieve relevant documents in cross-modal tasks without being optimized exclusively for any one of them. We present an extended analysis of our approach and without the need to fine-tune the model or results for every single retrieval task.

## 3.3 Scene-text visual question answering

In this section, we present the state of the art for works addressing the problem of scene-text visual question answering (ST-VQA). Although there have been enormous advances in VQA and text recognition as separate tasks, these models still fail when both tasks come together, i.e., when the model must recognize the text in the image, by including a level of reasoning. The most popular work in VQA is the one presented by Antol et al. [10], in which they introduce the problem clearly, as well as baselines over a new proposed database. Since then, many other works have been introduced with improvements in each one of their components, for example by the inclusion of attention mechanism [200–202] that incorporate a level of reasoning to the model. Textual recognition has been addressed by using optical character recognition (OCR),

which nowadays performs very well in cleaned documents, but, those have failed when there are very diverse text appearances that occur in the real world and also because of the amount of processing required, they end up having bad performances. Recently, deep neural models have been successful in presenting more robust models [203], also based on end-to-end configurations [204, 205] and attention mechanism as well [206].

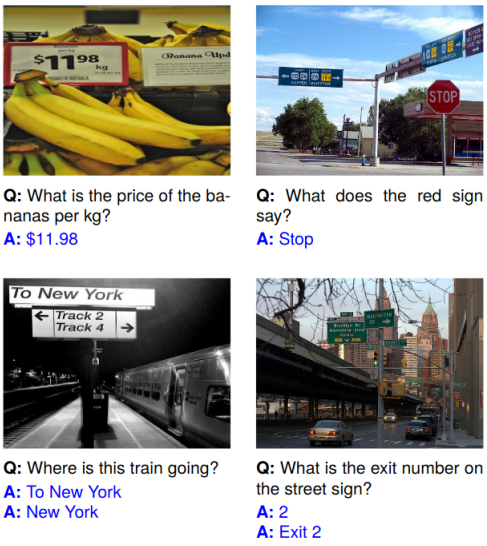



Figure 3.8: Examples of triples in ST-VQA database. Each triple is composed of an image, a question, and an answer. The answer can be obtained by interpreting the textual content in the image.

### 3.3.1 State of the art

To interpret written information in man-made environments is required to perform most everyday tasks, such as checking if a store is open or retrieving even more vital information, such as reading food labels to find allergenic ingredients. The task of ST-VQA pursues this objective, to interpret textual content in visual data such as scenes in our surroundings (see Figure 3.8 for examples of triples: images, questions, and answers). Because one of the main problems in addressing this task was the lack of databases, the first works in the specific task of ST-VQA aimed to introduce new databases that contain images, questions, answers. The main difference with traditional VQA databases is that the images contain target textual information required in the associated questions [57, 58, 207–209].

Figure 3.9 presents a generic and basic framework that represents the components contained in this task, generally by using a deep learning based model. The very first work introducing the task of ST-VQA was proposed by Singh et al. [207], they created a new database called TextVQA and presented a strategy (LoRRA) based on deep learning to solve the task. Their strategy contains the following components: a VQA system to process inputs obtained by using an object detection model for the visual features and GloVe vectors [76] to encode the question; a reading component to include OCR extracted by using a text recognition model as weighted Fasttext features [210]; and an answering module composed of a fixed + a dynamic answer space included by using a copy module (see Section 5.3.1) to handle the OOV problem which is one of the biggest problems to address in this task (see Figure 3.10 for an explanation of the copy module). Current VQA models are only able to predict fixed tokens which limits the generalization to out-of-vocabulary (OOV) words because they rely on fixed answer spaces (out-of-vocabulary are words not contained in the original pool of answers). The reality is that the text in images frequently contains words not seen at training time, and therefore it is hard to answer text-based questions based on a pre-defined answer space alone. The accuracy reported for their TextVQA database in the test set is 0.276% which demonstrates how difficult is this task.

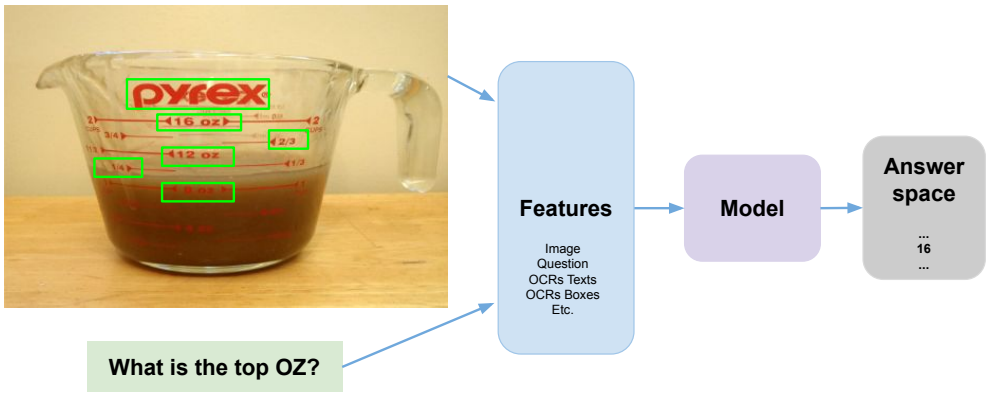



Figure 3.9: ST-VQA basic framework: there are three basic components of a framework that addresses this task: 1) a component that extracts and pre-processes features from the available data, such as images, questions, and OCR; 2) a component that uses the extracted features to model and infer the response (VQA component); 3) and an answer component that ultimately predicts the answer from a fixed or dynamic space.

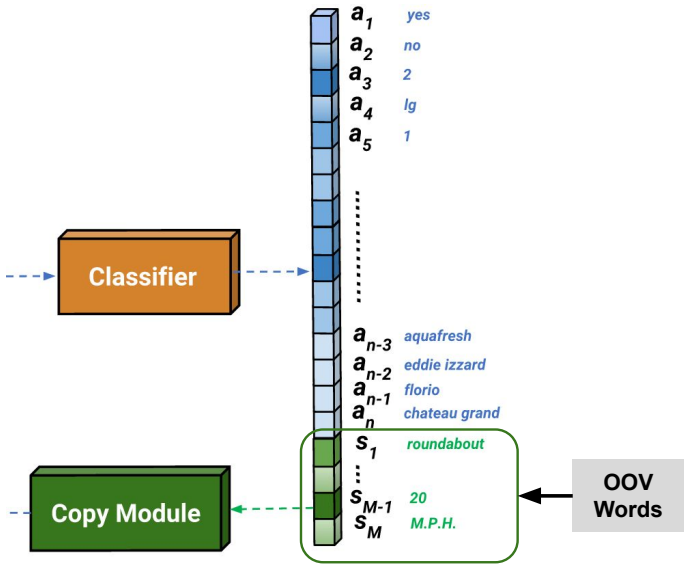



Figure 3.10: The copy module is a mechanism created to handle the OOV problem. It works by adding additional spaces to the answer space. Therefore, it has to decide whether the answer to a question is an OCR token detected in the image, or if the OCR tokens should only inform the answer to the question, i.e, the answer is already in the answer database (one of the *a* answer spaces). Figure partially extracted from [207].

The second most relevant work introducing the task was presented by Biten et al. [57]. Similar to [207], they also introduced a new database, ST-VQA, created for the "ICDAR 2019 Robust Reading Challenge on Scene Text Visual Question Answering" (see [211] for details on the competition). In this work, they presented final results for the proposed competitions from different participants addressing the task of ST-VQA under three tasks of increasing difficulty. The winning strategy, VTA, makes use of a strong architecture based on two types of attention, bottom-up (since they used an object detection model as a visual feature extraction method) and top-down attention (by including OCR information extracted with an OCR recognition system). For the text, they use a pre-trained Bert model [212] to turn all the text into sentence embeddings, including object names, OCR recognition results, questions, and answers (from the training set). Having these embeddings for the text and the images, they use a similar architecture as the one presented by Anderson et al. [201] to get the answer. The accuracy reported for this method for the database of ST-VQA is 0.2820%.

In [208], their principal input is the introduction of a large-scale database called OCR-VQA–200K

composed of book covers images along with methods providing a baseline performance. They perform the task by a combination of scene text-based method engines such as Tesseract (and OCR engine) [213] and VGG [86] detectors as feature extraction methods and VQA standard models. The results showed how the performance depends on the type of question and the information required. For example, binary questions result easier to answer than those that require semantic reasoning such as the book edition (0.58% vs 0.425% of accuracy).

**Graph based approaches.** Graph approaches are becoming more popular to model relationships among the modalities [214–217]. In this problem, generally, there are at least three modalities with many possible relations to model among them. Hence, each one can be represented as a graph structure. The graphs represent the visual, and the semantic data, in turn, divided into the questions, and all the textual data contained in the images such as words and numbers (see Figure 3.11) [214]. The principal idea then is to perform an iterative process to pass the information among the graphs to refine the final representation and finally, predicting the answer. The architecture may be a combination of standard models for text and images such as Glove, and LSTM for texts, and CNN for images.

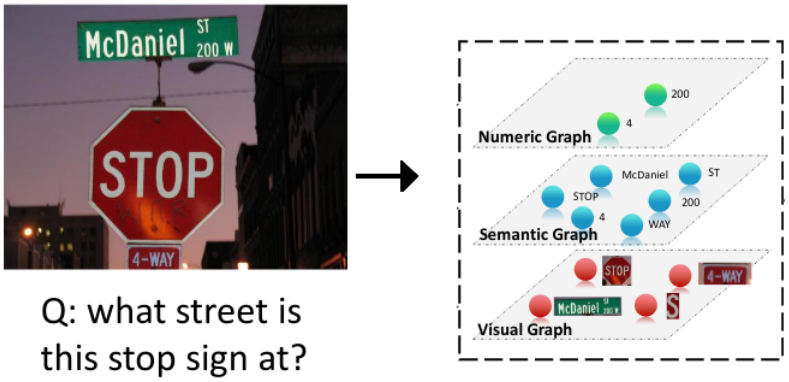



Figure 3.11: Graph Construction process. Figure partially extracted from [214]. In this model, they construct a multi-modal graph composed of three sub-graphs: a visual graph, a semantic graph and a numeric graph for representing the information in three modalities.

**Attention mechanisms.** Most of the models implementing attention mechanisms apply it or conditioned it on the question [215, 218]. For example, in [215], they propose a graph-based model that aims to encode the object-object, object-text, and text-text relationships appearing in

the image to apply an attention reason mechanism-based (where object refers to strictly visual data and text represents textual data in the image). Their main input is the graph structure constructed around the information contained in the question that guides the direction of the attention. In [218], the main idea is to compute attention weights over a grid of multi-modal spatial features. These features are constructed by concatenation of convolutional features and a spatially-aware arrangement of word embeddings. Then, by using the attention mechanism that is guided by the question, these features can be interpreted as the probability that a certain spatial location of the image contains the answer text to the given question. Attention over the OCR data extracted from the image is also implemented [219, 220]. For example, in [219], they implement an approach that uses a combination of question words and OCR tokens to predict the answer localization. For this, they impose an attention mechanism guided for the OCR boxes recognized in the image instead of their text content (see Figure 3.12). The final classification process is based on the most probably OCR boxes that contain the answer. However, the performance is directly affected by inaccurate OCR extracted from the image. Some performance accuracies are 0.335% for TextVQA database [220], and 0.641% in the OCR-VQA database [219].

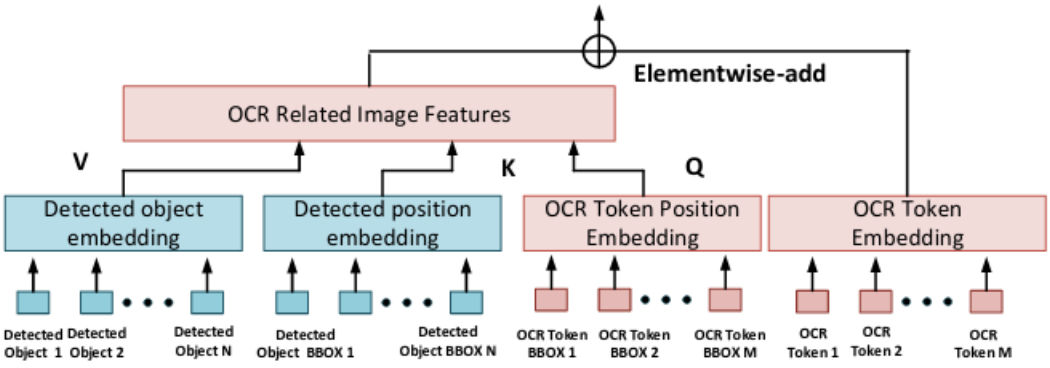



Figure 3.12: OCR attention. Figure partially extracted from [214]. In this model, they proposed to leverage context-enriched OCR representation to integrate the object features based on the spatial information (bounding box)

.

**Transformers.** The most recent architectures are based on transformers [221]. These can be seen as special attention mechanisms over standard features [209, 222, 223] or more specialized such as spatial features [216]. These methods extract features for question words using tex-

tual models, visual objects (with object detectors), and OCR tokens based on an external OCR system. Then, the features are projected into a common space using a multi-layer transformer which is a mechanism of attention that is applied on all the projected features [224]. In [222], they also include a copy mechanism that at each step in the iterative process, the copy of an OCR token from the image, or the selection of a word from a fixed answer vocabulary is carried out. This copying mechanism is first implemented in [207]. Training models incorporate two or more semantic training tasks according to the available data, which may increase the performance of the model [209, 223]. For example, in [209], they incorporate vision-language, scene-text language, and scene-text visual as three different pre-training tasks to improve the multi-modality representation. The inclusion of these refined attention mechanisms improves the performance by more than 20% [209, 223] and represents one of the future research directions to address this task.

### 3.3.2 Discussion

This task was recently proposed and that is why state-of-the-art works that address it are still lacking. This is due to all the additional challenges to the traditional task of VQA, which this task imposes, the main one being the location of relevant textual information in the scene. Figure 3.13 presents some examples of predicted answers of a recent model (samples extracted from [216]). We can see how models can be misled by textual data contained in the image but that is not part of the correct answer. The models truly require complex reasoning about positions, colors, objects, and semantics, to locate, recognize and eventually interpret the recognized text in the context of visual content, or any other available contextual information [57]. In general, these approaches propose frameworks composed of successful feature extraction methods according to each available modality, together with the mechanisms of attention, the correlation between the modalities in the form of graphs and answer spaces. Because many approaches use traditional OCR methods in documents to extract the text contained in the image, the performance also depends on the quality of these OCR. Another important aspect to address is the size of the answer space. It is necessary to explore fixed versus dynamic answer spaces (such as the use of copy mechanisms) to achieve a level of generalization and robustness of the models.

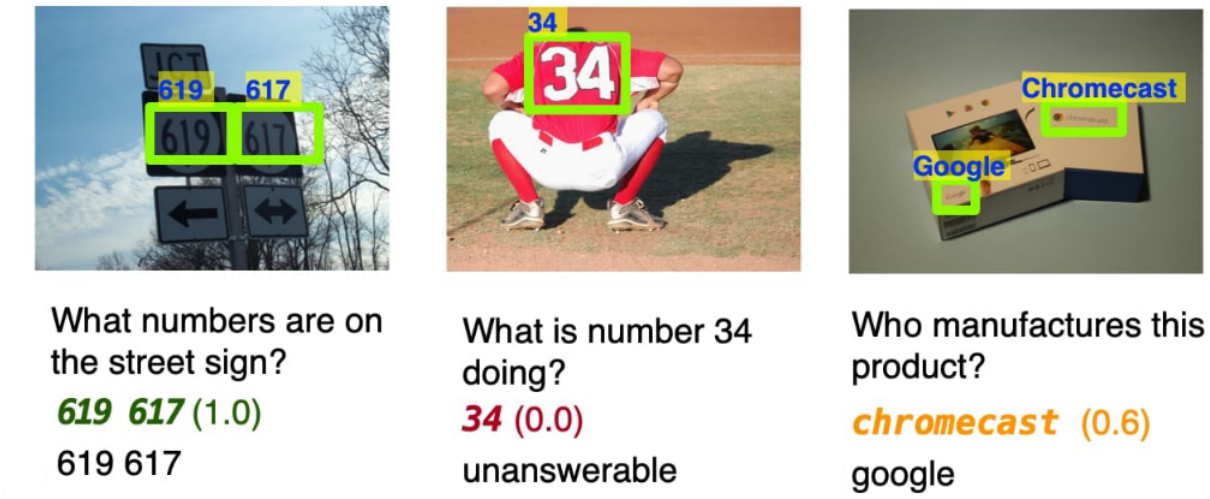



Figure 3.13: Qualitative Examples (samples of results extracted from [216]). In the first line, there is the associated question, the second line is the predicted answer and the last line is the ground truth answer.

In Chapter 5, we present our proposed contribution number three (see Section 1.4) in which our goal is to adapt our previous frameworks and improve them to tackle the task of ST-VQA. We evaluate the impact of various basic and auxiliary modules under different strategies. We explore and assess the quality of different feature extraction methods for the modalities available in the databases, including target images, questions, and answers, and other additional data. We explore the performance of the model concerning variations in the answer space, such as using a BOW or a copy module under two different metrics for the calculation of scores. Our final evaluations assess the performance by including supplemental data to train the system.

## 3.4 Attribute learning

In this section, we present the state of the art for attribute learning. Research on attribute learning has been extensively explored for different purposes and various applications (see Figure 3.14). In this thesis, we are interested in the state-of-the-art works that address the problem of learning by attributes under two problems, first, the problem of imbalanced learning, and second, the problem of learning under weak supervision. For semi-supervised learning of attributes, some articles may refer to it as a zero-shot learning problem. Learning visual attributes has been

shown to be beneficial especially for transferring learned information between object classes. For example, learning the color "blue" or the pattern "spotted" from a series of images can then be used to recognize these attributes in a variety of unseen images and object classes. The term "attribute" is defined in Webster's dictionary as "an inherent characteristic" of an object, and there exist various types of attributes such as appearance adjectives (color, texture, shape) and the presence or absence of parts (has wings? has a tail?) [225]. Attributes can also be classified into semantic (timid, solitary) and non-semantic (has leg? is blue?).

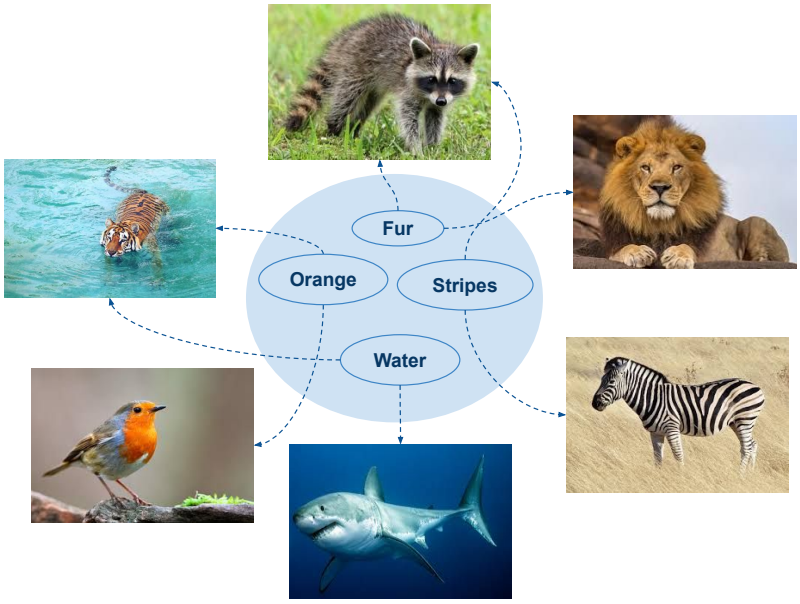



Figure 3.14: The task of attribute learning generally aims to learn and discover attributes that are only defined at a class-level. The aim is to learn to predict attributes to unseen samples either from classes seen or unseen at training.

### 3.4.1 State of the art

As stated in [226], attribute recognition was first addressed with approaches based on traditional classification methods, such as support vector machines [69, 225]. Since then, it has been widely studied and evolved towards the use of deep learning methods [227–231]. Some of the most popular databases have attribute annotations only at the class level [232–235] because annotating each instance is a very expensive process and it may also require a high level of expertise. Therefore, in most cases, attributes are seen merely as intermediate clues for performing fine-

grained classification[2] where accuracies exceed the 80% [236–238]. This auxiliary data may be used as an input where each sample has an attribute and an object [239], or as an intermediate representation [240] (see Figure 3.15).

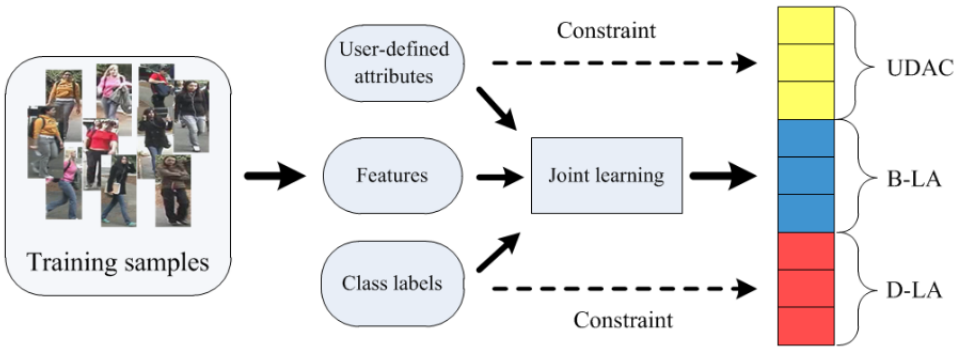



Figure 3.15: Attribute data is generally used as auxiliary information to perform the task of fine-grained image classification or to learn more discriminative mid-level representations for instance-level object recognition (Figure partially extracted from [241], where the definition of abbreviations is as follows: user-defined-attribute-correlated (UDAC), discriminative latent attribute (D-LA), and background latent attribute (B-LA) dictionary subspace.

Two tasks in attribute recognition, pedestrian recognition, and face recognition, have been extensively studied and as a result, there are several existing databases. Recognizing the attributes of pedestrians is an important sub-task in attribute learning due to its important role in video surveillance [229, 242]. Given the image of a person, pedestrian attribute recognition aims to predict a group of attributes to describe the characteristics of this person from a predefined list of attributes [243]. Due to the importance of the task, several annotated data sets contain labels at different levels. The preferred scenario followed in this task is to estimate all attributes in a model and treat each attribute estimate as a task, either as a multi-task or multi-label learning scenario [244]. Similar to pedestrian recognition, the facial attribute recognition task has received attention due to its importance in surveillance [245, 246]. This task consists of two basic subtopics: estimation of facial attributes, which recognizes whether facial attributes are present in given images, and manipulation of facial attributes, which synthesizes or removes the desired facial attributes [247]. Facial attributes represent intuitive semantic features that describe

[2]The fine-grained image classification tasks focus on differentiating between classes of objects that are very difficult to distinguish, such as species of birds, or animals.

human-understandable visual properties of facial images, such as smiles, glasses, and mustache [248]. Similar to pedestrian recognition, there are several annotated databases, the most popular of which are CelebFaces Attributes database (CelebA) [249] and Berkeley Human Attributes [250]. Other popular attribute-oriented databases include Animals with Attributes (AwA) [234] which is composed of animal classes with 85 different semantic and non-semantic attributes; the Caltech-UCSD Birds-200-2011 (CUB) [232] which is a database that contains images of 200 bird species with 312 attribute labels; and SUN Attribute (SUN) with 102 attributes representing scene classes of natural images such as fire, diving, camping, etc., [251]. Figure 3.16 shows examples of images and their attributes from three different databases.

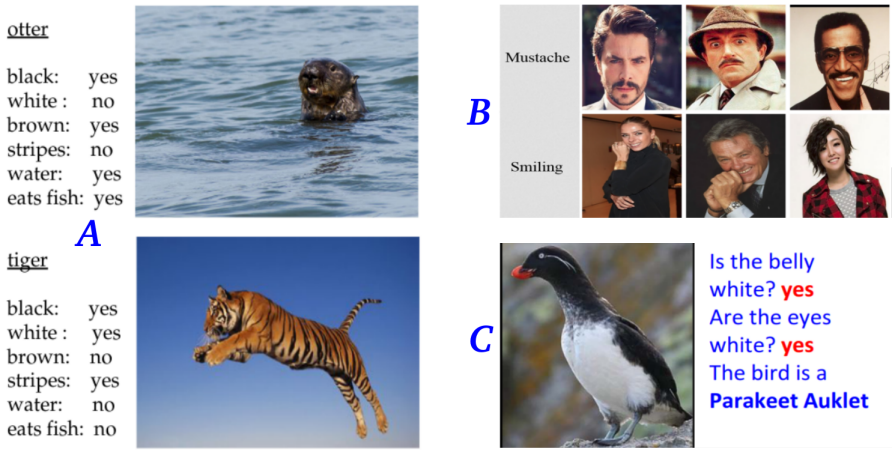



Figure 3.16: Examples of images and their attributes from three different databases. A) AwA; B) CelebA; C) CUB

Multi-tasking learning is a training strategy that allows each attribute to be treated independently (see Appendix B for a description of these types of learning strategies). Multiple attribute spaces can also be learned at the same time and tasks do not have to share the same attribute space as they refer to different domains [252]. The work in [253] proposed a method based on graph neural networks in attribute learning. The authors study the dependencies between different relative attributes of images. They explore the similarity between multiple images in a graph, where each node represents an image and the edges are formed based on the relationship given by the attributes. The framework is multi-tasking where it is possible to add different types of nodes to the model that represent different attributes to learn. In [241], a multi-task transfer

learning framework is implemented. Their unified framework allows learning both user-defined semantic attributes and discriminative latent attributes based on a dictionary learning model (see Figure 3.15 for their architecture proposed).

Models based on ZSL for classification also make use of attribute information by performing projections in a semantic space driven by attribute data. For example, in [254], they propose a framework that propagates attribute annotations to non-annotated images using sparse attribute propagation. This is done by learning bidirectional projections between visual features and attributes. When learning bidirectional projection, a vector of visual features is projected first into a semantic space and then again into a space of visual features to be reconstructed. This self-reconstruction aims to improve the generalizability of the model. In [255], they proposed a ZSL framework that aims to jointly learn global and local discriminative characteristics using only class-level attributes. The model contains a visual-semantic embedding layer that learns global characteristics and a network of attribute prototypes that learns local characteristics. This network simultaneously regresses and de-correlates attributes from intermediate features. The accuracies reported by these models are (CUB - 0.73%, AwA2 - 0.717%, and SUN - 0.657%) [255] and (CUB - 0.493%, AwA2 - 0.775%, and SUN - 0.843%) [254].

**Imbalanced Learning.** With increasing access to data, the imbalance problem of the available data has also emerged [256]. When this imbalance ratio is high, most of the existing imbalanced learning methods seriously decline in their classification performance. The survey work in [257] presented a very comprehensive summary and description of the ranking of strategies that address the problem of imbalanced learning. They classify these strategies into two general groups: 1) Pre-processing techniques that include sampling, feature selection, and extraction strategies; and 2) Cost-sensitive learning by incorporating specific weights for the target classes at the algorithm level or the sampling/feature selection technique. Current works primarily address the problem by giving a greater focus to one of these strategies. Cost-sensitive learning strategies make the loss function the main focus. This is done by capturing the misclassification of the majority and rare classes alike by reducing the mean square false error [258]; maximizing the sum of the true positive and true negative rates using a classifier such as decision trees [259, 260]; or maintaining both inter-cluster and inter-class margins [227, 238]. Most current work

uses a combination of classical techniques [261]. These approaches typically use sampling and a weight loss strategy [262]. This could be through batch updates [263], [264] that helps to get better generalization and discrimination. The sampling pre-processing technique can be replaced with a feature learning strategy combined with a cost-sensitive learning strategy as well [265]. More difficulties arise with non-ideal database properties, such as when a highly subjective annotation process creates a discrepancy amongst the ground truth data. For this, many current works assumed ideal conditions. For example, expecting the combination of attributes related to a class to be present in all images annotated with it [240]. In real databases such as the CUB database, this aspect is not fulfilled. The subset of attributes for images belonging to the same class can also be different, making the problem even more difficult. Some representative reported accuracies are (CelebA - 0.921%) [238], (AwA2 - 0.701%) [264], and (AwA2 - 0.699%) [240].

**Weak supervision.** The most basic disadvantage of supervised learning is that the database must be manually labeled, and this process sometimes requires a level of expertise. Although most machine learning methods require large amounts of data, the reality is that it is a very expensive process. The solution is to develop strategies based on a combination of labeled and unlabeled data. For attribute learning, this is the desired learning strategy since attribute data can be very expensive to obtain for individual images. The most popular approaches are called ZSL and semi-supervised learning. The original ZSL focuses on improving classification performance by recognizing objects of unseen classes [266, 267]. One of the advantages of ZSL is that it allows us to recognize new objects without having to annotate samples for them, which can represent a very expensive process. Most works using attribute-based strategies report that attribute data plays an important role in improving classification performance. The key idea is to transfer attribute-based knowledge from known to unknown classes. Attribute data can be exploited in different ways: finding new attribute data based on known relationships between semantic attributes and visual objects [239], assuming correlations amongst attributes [268, 269], selecting the most relevant attributes [270], mixing independent and correlated attribute relationships [271] and improving feature generation models for attribute localization [255, 272]. Attributes can also be used to model attention mechanisms that focus on the most relevant image

regions (see Figure 3.17). However, part localization training data may be required, and that data is more difficult to obtain at scale. As these approaches are based on determining specific relevant regions in each image for each attribute, they may not apply to visual concepts shared by groups of images with the same semantic data (classification accuracy: CUB - 0.581%, AwA2 - 0.671%, SUN - 0.332%) [231].

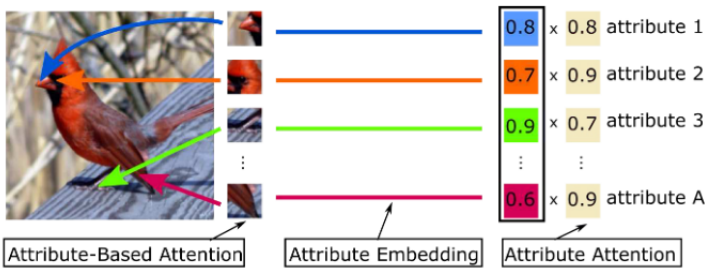



Figure 3.17: Attention over attribute regions. Methods can search for discriminative regions through dense attribute-based attention mechanisms that may rely on part-localization data not always available (Figure extracted from [231]).

Semi-supervised and unsupervised learning approaches are increasingly used for vision tasks, however, for attribute learning, these approaches are less frequent. The ability to train with little or non-annotated data is invaluable [272] (classification accuracy: CUB - 0.694%, AwA2 - 71.5%). Unsupervised learning of attributes is rare. Huang et al. [273] are one of the first works to address a completely unsupervised scenario for the attribute learning problem. They proposed a two-stage pipeline to learn attributes like binary hash codes with multiple CNNs that share architectures and weights (classification accuracy: CUB - 0.894%). Unsupervised learning sees the attributes as a compact set of binary codes instead of textual or semantic labels [274, 275].

Figure 3.18 shows example results for the principal task that is classification and attribute visualization. As we explained, the main task addressed using attribute information is image classification. Some works present an additional visual analysis regarding the learning with attributes as in the Figure.

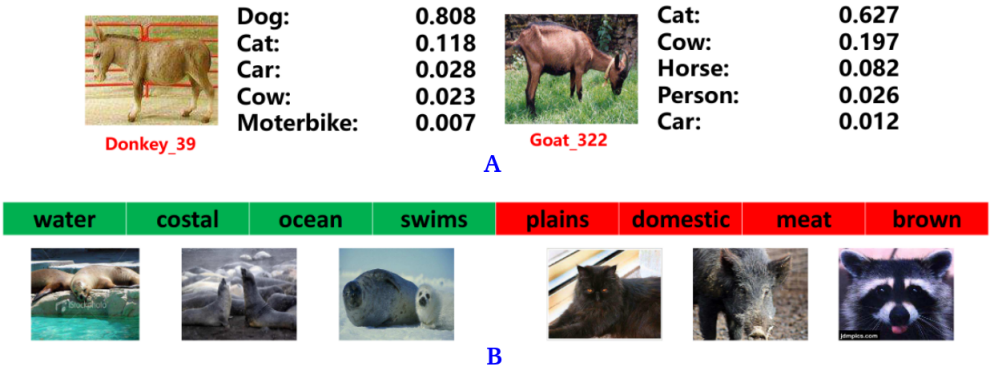



Figure 3.18: Example results for two tasks: A) Similarity representations of unseen class samples where only the top five similar classes are shown, B) attribute visualization: green blocks show attributes with largest activations and red ones show attributes with smallest activations (Figure partially extracted from [268]).

### 3.4.2 Discussion

Attribute learning has been extensively explored for different purposes and for different applications as described above. Most of them use the attribute information as a type of supervision to perform fine-grained classification or as auxiliary information. Another factor is the ideal database conditions that are expected. In our work, we explore the attribute learning problem first by analyzing two problems: imbalanced learning at the attribute level and weak supervision in the form of semi-supervised learning. These problems aim to address our contributions three and four (see Section 1.4) about problems with the data. For the first problem, we explore and propose adaptations to classical imbalance strategies which mainly include "sampling" and "cost-sensitive learning" for the objective task of multi-attribute learning (see Chapter 6). We also explore the problem from the perspective of non-computer users. Users may find it difficult to use highly complex deep learning models, while simpler models offer adequate performance. Therefore, we see these models as black boxes and we emphasize the development of explainability strategies around the data.

In the second part of our work on attribute learning, we tackle the problem of weak supervision by proposing a semi-supervised learning strategy to propagate attribute data to unseen samples (see Chapter 7). Under the ZSL setup, the testing instances are assumed to come from the

unseen classes. This problem setting is somewhat unrealistic. The ideal learning scenario is that the testing instances can come from both the seen and the unseen classes [276]. We implement this idea in our approach where test instances can come from all classes because the emphasis is on propagating attributes to samples regardless of their class. We propose to use five attribute spaces, each one targeting different descriptions for the visual information contained. Similar to the proposals in [252, 277], we use heterogeneous attribute spaces, but unlike them, we do not have access to such annotations in the training phase, i.e., we do not have annotations available per sample during training. We take inspiration from the works in [278, 279], which present a learning algorithm using curricular learning through an iterative process. The key idea is to propagate labels to unlabeled samples in an iterative and self-paced fashion for the task of image classification. We adapt these principles for attribute annotation using multi-task learning with few labels inferred from class-attribute relationships.

In this chapter, we have presented the state of the art for each one of the tasks addressed in this thesis. For each task, we present the methods in the state of the art along with a discussion around the problems and challenges to address. We finalize each section with a summary of our proposed strategies that are presented in the chapters on multi-modal retrieval (Chapter 4), scene text visual question answering (Chapter 5) and finally attribute learning (Chapters 6 and 7).

# CHAPTER 4

# Multimodal Information Retrieval

## 4.1 Motivation

In our first approximation to the problem of multimodal learning, we adapt a VQA (Visual-Question-Answering) architecture for multimodal fusion. VQA systems are generally trained to answer natural language questions related to the content of images. There has been great progress in the design and understanding of VQA systems, and the vision community has introduced databases and improvements to make the systems accurate ([280]). Given their natural design to process images and text, we aim to solve the question of **how useful is the multimodal representation encoded by such models to solve queries in cross-modal retrieval problems?**

More precisely, in this chapter, we propose to evaluate how VQA systems could be used to create a common latent representation for multimodal data. Our goal is to evaluate the representation learning capability of VQA network architectures to encode the input modalities in a meaningful way for cross-modal retrieval tasks. Searching data in documents generally relies on one data modality (image or text) while users generally expect to take advantage of all the available data.

By leveraging images, text contents, and their semantic relationships in an integrated fashion, an ideal system should propose a diversity of ways in which a multimedia collection could be used.

The state of the art in cross-modal retrieval is vast. The most successful methods are based on deep learning and the most popular deep learning variations are adversary models and hash-based models [106] (see Chapter 3, Section 3.2). Adversary learning has been very successful in many applications, but it also presents downsides such as the additional complexities and computational burden due to multiple processes to optimize. At least three paths require the learning of training parameters when implementing this strategy [61, 153]. Also, these methods show that the performance is highly correlated with the characteristics of the data set, and that data points that are far enough away from the training data distribution can significantly damage the performance [59]. On the other hand, the main problem of hashing methods is the requirement of a colossal amount of labeled data that is not always available. Because of the hash collision problem (producing the same hash value), longer hash codes are needed to ensure the precision of the retrieval, which brings additional time and computational overhead, and leads to a decrease in the recall rate [175, 176, 178]. Finally, another flaw in the state of the art is that most of the methods appear to be over-optimized for the task of Txt2Img while having lower performances for Img2Txt [139]. Or in a general way, they seem to be optimized for a single retrieval task while not tackling the remaining ones.

Hence, we aim to assess our hypothesis and address the problems mentioned in the state of the art by using our proposed framework. We propose to use three very popular databases in two different configurations: the topic classification performance with the Wikipedia Retrieval database ([107]); the retrieval tasks with the Wikipedia database, the Pascal Sentences database ([108]), and the MIR-Flickr-25k database ([110]). The first database, Wikipedia retrieval, is one of the most widely used in the state of the art, this is why we are using it as our case study. Pascal sentences and MIR-Flickr-25k are also well-known databases and present different structures, especially in their associated text. This allows us to make a fair and larger comparison against the state-of-the-art methods. Our extensive experiments simultaneously evaluate all cross-modal retrieval tasks under the same computational framework as well as the uni-modal

tasks. We compare our approach against recent and specialized models that represent a variety of methodologies, allowing researchers to compare performance in different cross-modal retrieval tasks under different assumptions. Surprisingly, even though the VQA architecture is a well-established model that can be used for other image-text problems, previous work had not investigated its usefulness for cross-modal retrieval. Even the most recent approaches for cross-modal retrieval do not take advantage of the innovations introduced in VQA architectures. In summary, the main contributions in this chapter are listed next:

1. The re-interpretation of the VQA architecture for cross-modal retrieval, followed by an extensive experimental evaluation of its capabilities in this problem.

2. Our work is the first to evaluate all cross-modal and uni-modal tasks with a single model trained only once without fine-tuning in specific tasks, reaching highly competitive results as well as state-of-the-art results in some cases, even though their baselines may be exclusively over-optimized for single tasks. This shows that a unifying model for multimodal retrieval is possible.

3. The simplicity of our approach serves as a baseline for future research and can inspire extensions to push performance even further. We expect future researchers to take advantage of this well-established architecture to reach higher performance in their models.

## 4.2 Approach

To design search systems that allow both texts to image queries, image to text queries, and uni-modal queries using a single model, we make use of an architecture inspired by Visual Question Answering (VQA) models ([10]), and we extend it for Deep Multimodal Learning. This model matches the inputs and outputs of a multimodal system (see Figure 4.1 for details), and it can be used to compute embeddings for different modalities. In what follows, we describe the main three components of this architecture.

**Visual network.** We evaluated two CNN architectures to compute visual representations: VGGNet [86] (similar than in the VQA architecture from [10]) and ResNet50 ([87]). We use pre-trained weights from object classification using the ImageNet database and keep the network

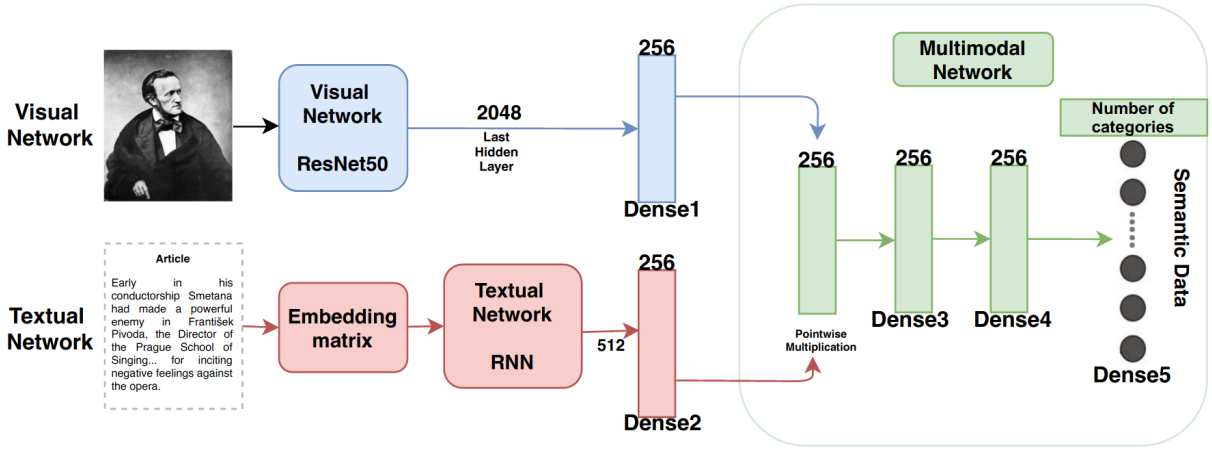



Figure 4.1: Overview of the architecture for Deep Multimodal Embeddings (DME). There are three components: 1 - The visual network (in blue), which takes the last hidden layer of a pre-trained ResNet50 model with 2048-dim as input for the visual neural network, followed by a fully-connected layer with output 256-dim. 2 - The textual network (in red) composed of an embedding matrix of GloVe vectors of 300-dim, followed by two LSTM cells, and a fully connected layer with output 256-dim. 3 - Finally, a multimodal network (in green), where the different modalities are fused. This component contains a set of fully connected layers with the last layer (output layer) used as a classifier during the training phase. At the retrieval phase, we can use different layers to compute the embeddings. Either by using layers of each modality (Dense1 for visual data, Dense2 for text data) or by using layers after the pointwise multiplication (Dense3, Dense4, or Dense5) that contain multimodal information.

frozen for feature extraction. The feature vector produced by VGGNet has 4,096 dimensions, while the ResNet produces 2,048. These activations from the last hidden layer are used as inputs to a fully connected layer with an output of 256-dim to reduce the dimensionality and match the connection with the multimodal network.

**Textual network.** In the original VQA architecture proposed in [10], the textual data is modeled by using the words in the questions to create a bag-of-words representation. In our framework, we modify this textual embedding for a more suitable and successful model. Hence, we make use of a recurrent neural network (RNN) that models the sequence of words in sentences. This provides the advantage of having an ordered representation of word sequences in a fixed-length feature vector, which contrasts with order-less bag-of-words models. LSTMs are powerful neural networks to learn sequence models ([281]) and are used in VQA systems to represent the

words used to formulate questions ([282]). In the case of other sequences of text, we can evaluate the performance by identifying the most 'relevant' words present in the input text. While question words such as "Where" and "What" are important to understand what the requested information in the questions of VQA systems is, in this work, we are more interested in nouns. Nouns are words with semantic and discriminatory content that are possibly more related to the general content/topic/concept of an image (such as "King", "Cat", "Church", etc) than question words. To process sentences with the RNN, each word in the text is first encoded into 300-dimensional embeddings using the GloVe model ([76]). The vocabulary used consists of all the words seen in the training database, and those with an embedding representation in the pre-trained GloVe embeddings. The matrix of word representations is used as an input to the RNN, which has two LSTM cells followed by a fully connected layer with intermediate dropout layers and activation functions ReLU to get a 256-dim vector. We also explored different sequence lengths for the input texts and evaluated the varying impact on performance.

**Multimodal network.** In the original VQA architecture proposed in [10], they use as the multimodal network a multi-layer perceptron (MLP) classifier with 2 hidden layers and one last layer of dimension 1000 (size of the answer space) [10]. Instead, we implement a deep multi-layer neural network with four layers as follows. We use an element-wise fusion layer (point-wise multiplication or Hadamard product), where both modalities, visual and textual, are combined. We found this operation to be critical for obtaining improved performance, which is defined as:

$$(V \circ T)_i = (V)_i (T)_i \; for \; all \; 0 \leq i \leq n \; , \tag{4.1}$$

where V = Visual vector, T = Textual vector n = dim(V) = dim(T). This fusion layer is followed by two fully connected layers with dropout, and the network has a classification layer in the output. This classification layer is in charge of the model supervision and is composed of classes or concepts that semantically discriminate the samples in the database. A database usually contains different classes, where a class represents a group to which items are assigned based on a similarity metric or defined criteria. Each image-text pair sample is associated with one of these classes. The model is trained in an end-to-end configuration with cross-entropy as the loss function. The last activation function is a softmax function applied to the final output vector and

computed as follows:

$$f(score)_i = \frac{e^{score_i}}{\sum_j^{Cat} e^{score_j}} \tag{4.2}$$

where $score_i$ is the score inferred by the model for the class $i$ in $Cat$, $score_j$ are the scores inferred by the net for all the classes in $Cat$, after applying it to all classes, it gives us a vector in the range (0, 1) in which all the resulting elements add up to 1. As this can be seen as a probability vector, it helps to determine the most probable class to be associated with each image-text input pair. Once the system is trained, we can compute embeddings using different layers of the model, and evaluate their quality in retrieval tasks. We can compute separate embeddings by using layers before the modalities are combined, but this would imply using only data of the target network, this is either visual or textual, and therefore not multimodal. We are interested in computing features in a latent space that can match any data modality, and therefore we use features from the outputs of layers Dense3, Dense4, or Dense5 as embeddings for cross-modal retrieval (green layers in Figure 4.1, Page 62). To compute multimodal features with a single modality we simply remove the other network and skip the point-wise product, which is equivalent to using a constant vector of ones as a replacement. Figure 4.2 shows an example of the process followed when the query only contains data from one modality. In the example, the query is composed of visual data (path A) or textual data (path B) that is sent to the framework to find the corresponding multimodal embedding.

As we presented, our framework can handle all types of multi-modal queries without separated training phases. Unlike adversarial and hash models that require optimizing several models and loss functions, we only optimize a loss function that simplifies the training process and avoids additional complexities and computational burden (see Section 3.2). With this framework, we aim to demonstrate that it is possible to achieve competitive performance for different tasks. We also demonstrated that for testing, we can send queries containing only uni-modal data since our model can skip multi-modal dependent layers (see Figure 4.2). In our experiments, we will present several evaluations in multiple databases that support our proposals.

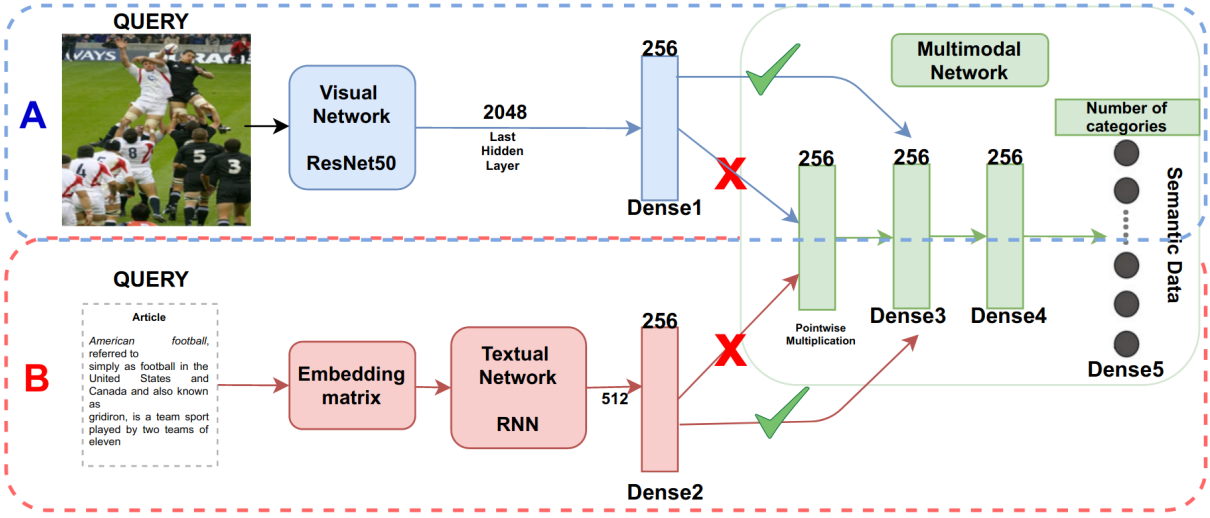



Figure 4.2: Example of the process followed when a visual (A) or textual (B) query is sent to the framework. Because there is only one type of data, the connection between layers Dense1 or Dense2 and the point-wise multiplication layer is skipped. Instead, the connection is made directly to the Dense3 layer.

## 4.3 Evaluation

In this section, we include the results of our experiments. First, we introduce the databases used in our analysis. Then, we present the implementation details and finally, the results for our ablation studies, including classification results and, multimodal retrieval results that include the comparison against the state of the art.

### 4.3.1 Databases

For our experiments, we make use of three databases: Wikipedia Retrieval, Pascal Sentences, and MIR-Flickr-25k. We use the Wikipedia retrieval database, a widely used database in cross-modal retrieval research, to make an extended analysis of the classification performance of the model. Pascal and MIR-Flickr-25k are also widely used in retrieval tasks because they have good ground truth annotations and well-organized textual representation for each image.

**Wikipedia Retrieval database [107].** The database consists of 2,866 image-document pairs with a train/test division of 2,173 and 693, respectively. We re-separated the train set, into train

and validation, with a ratio of 80%-20%. This gives us a partition of 1,738/432 training and validation samples, respectively. The median text length is 200 words. After analyzing the distribution of words, we kept only those with frequencies between 10 and 150. The length of words on average for all the samples after pre-processing is 101 tokens (with a minimum of 26, and a maximum of 400). We explored different sequence lengths for training our model, including 26 words, as well as 46, 76, and 100. There is also a third source of information for the data, each image-text pair is labeled with one of 10 semantic classes. The images and text belonging to this database are very diverse and cover a large range of domains. These are Art & architecture, Biology, Geography & places, History, Literature & theatre, Media, Music, Royalty & nobility, Sport & recreation, and Warfare. The text associated with each image contains words that are related implicitly but also explicitly to the images. Those words were extracted from the section in which the image was placed in the article. The implicit case imposes additional challenges to infer relationships among the data. Figure 4.3 presents some samples contained in the database Wikipedia Retrieval ([107]). We can see the challenges associated when training a multimodal model with this data. This model must learn the high semantic level contained in the database on the one hand and to be able to deal with the references to different topics in each image on the other hand.


Figure 4.3: Examples of Wikipedia database, at the top, the topic of the article, at the bottom, the first 10 tokens after pre-processing.

**Pascal sentences [108].** This database was created with the Amazon Mechanical Turk platform [109] and provides pairs of images and a set of descriptions of their content provided by the users of the platform. This contains 1,000 samples of images associated with several sentences and descriptions of their content (approximately 5 sentences per image) from 20 classes,

50 images per class. We follow the standard partition of 600/400 (i.e., 30/20 samples per class) for training and test samples. For texts, we concatenate all sentences for each sample and extract all words. We set the sequence length as the average number of words in all samples, resulting in a sequence length of 25 words. We did not prune the distribution of words in this database.

**MIR-Flickr-25k [110].** This database consists of 25,000 images downloaded from the social photography site Flickr and annotated using two sets of annotations: a potential set of annotations with a total of 24 semantic concepts, and an extended relevant set of annotations with a total of 38 semantic concepts. All images are annotated with at least one of the concepts. We set the sequence length in this database to 15 tags. Note that these are not sentences; instead, the text annotations are just a collection of tags. The median number of tags per image is 5, and we use all tags independently of their frequency. To compare with the state of the art, we make two partitions of the data, (1) as in ([283], [178]), taking approximately 2,000 samples as the test partition, and (2) taking 95% of the data as training, 5% for test and reporting the average of 3 experiments, following the setup of ([175]).

### 4.3.2 Implementation details

To implement our approach[1], we used the Keras framework, and trained the model described in Figure 4.1, Page 62 with a learning rate of 0.001. The input size of the convolutional network is 224×224 pixels [2]. We explored data augmentation strategies for images and text and obtained better results when using minimal augmentation for images. For the text, we augment the word sequences by creating different combinations of up to $N$ random tokens from all the tokens available for each image (where $N$ is the maximum sequence length in the experiment). The results are depicted in Table 4.1, Page 69. The first part of the system relies on a classical training phase as it is based on deep neural networks. We systematically explored the batch size (16-256) having better results when using bigger batches sizes. Indeed, for this type of model, it is better to pass in each batch a randomly but well-distributed (i.e. similar number of samples in each class) set of samples of each class. On the contrary, batches of smaller sizes can result

[1] Code available at https://github.com/lvbeltranb/DME
[2] https://keras.io/api/applications/resnet/#resnet50-function (accessed Feb 2021)

in a lack of samples from an entire class, biasing the learning process. We explore dropout rates between [0,0 - 0,5] with better results obtained with dropout rates greater than 0.3. This indicates that the model is preventing over-fitting to the training data. Too much dropout can also reduce performance because the dimensions of the architecture are not too high, thus, too many disconnections make the model losing valuable information. Finally, we also explored the performance of different optimization algorithms for training, having the best results using the Adam and RMSPROP ([284]).

### 4.3.3 Results

This section gathers the results of our evaluations in this chapter. First, we worked on the classification analysis. The two next parts are related to the retrieval results for all the selected databases in a uni-modal and a cross-modal way.

#### 4.3.3.1 Classification accuracy

To the best of our knowledge, there are no results for classification accuracy reported for Wikipedia Database. To get a better understanding of the system performance, we evaluated the classification accuracy in each separate component: the visual and the textual networks. For image classification, we explored embeddings as well as the number of suitable layers before merging the data with the multimodal network. The best results of visual classification are obtained when using the last hidden layer from a pre-trained ResNet with 50 convolutional layers to compute visual embeddings, followed by 1 fully connected layer and the classification layer. For text classification, we explored sequence lengths leaving the rest of the network fixed. The best configuration for the architecture includes two LSTM cells followed by 1 fully connected layer plus the classification layer. We found the length of the sequence input to work well with 100 tokens. Finally, we explored the number of layers in the multimodal component for pair classification, where the best configuration uses 2 hidden layers in the multimodal network. We also compared the performance of our neural network classification model against a representative baseline method based on SVM classifiers for each modality. The SVM baseline used BERT embeddings for text ([285]), and ResNet embeddings for images ([87]). The baseline classi-

fication performance was 0,7445 for text, and 0,4271 for images, while our approach reached 0,7474 for text and 0,5165 for images (see Table 4.1). The final accuracy for the multimodal model is 0,7561.

Table 4.1: Classification accuracy evaluated by using different sets of data: first, testing different image architectures when only visual data is passed, second, evaluating different sequence lengths for the representation of the samples when only text is passed, and finally, by testing a different number of hidden layers in the multimodal component when visual and textual data is passed. Results are reported for the use case, the Wikipedia Retrieval database along with baselines results for the uni-modal evaluations.

| **Modality** | **Model** | **Acc** |
|---|---|---|
| **Images** | VGGNet19 + 1 hidd + Class | 0,4733 |
| | VGGNet19 + 2 hidd + Class | 0,4574 |
| | VGGNet19 + 3 hidd + Class | 0,4531 |
| | ResNet + 1 hidd + Class | **0,5165** |
| | ResNet + 2 hidd + Class | 0,4920 |
| | ResNet + 3 hidd + Class | 0,5122 |
| | Baseline ResNet + SVM | 0,4271 |
| **Texts** | 26-LSTM + 1 hidd + Class | 0,7272 |
| | 46-LSTM + 1 hidd + Class | 0,7359 |
| | 76-LSTM + 1 hidd + Class | 0,7301 |
| | 100-LSTM + 1 hidd + Class | **0,7474** |
| | Baseline BERT + SVM | 0,7445 |
| **Multimodal** | Visual Net + Text Net + Class | 0,7243 |
| | Visual Net + Text Net + 1 hidd + Class | 0,7142 |
| | Visual Net + Text Net + 2 hidd + Class | **0,7561** |

In our experiments, we observed that the performance does not degrade in the multimodal system. Although the visual modality has significantly lower performance than the text modality, since samples from the same class at the textual level share key words between them, the visual content can vary drastically, it also opens interesting questions for future research to balance the contribution of performance from each modality.

The figure 4.4 presents the learning curves for the best multimodal configuration found in Table 4.4. The loss and accuracy training curves have some small disturbances in their convergence, however, they can reach optimal values in less than 40 epochs. The accuracy in the test set is expected when compared to the accuracy obtained in the train set ($\approx$ 0,99%). This also shows

the absence of over-fitting since the test accuracy still achieves a value no less than $0,75\%$.

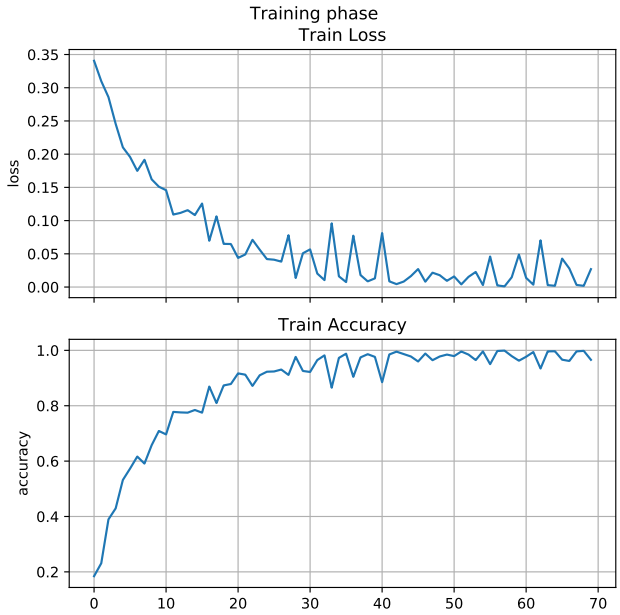



Figure 4.4: Learning curves for loss and accuracy in training phase for the Wikipedia retrieval database in the multi-modal framework. Both curves converge in less than 40 epochs.

#### 4.3.3.2 Uni-modal and Cross-modal retrieval

In this section, we focus on the evaluation of multimodal retrieval tasks to answer the question "How useful is the multimodal representation encoded by such models to solve queries in cross-modal retrieval problems?". Assuming our DME model is trained, the cross-modal retrieval process is then divided into two phases, the indexing phase, and the search phase. The indexing phase generates embeddings for all the documents in the database using the target modality only. In the retrieval phase, the query modality is processed using our DME model without activating the network of the target modality (visual or textual, see Figure 4.2, Page 65). For example, if we want to carry on the task of Img2Txt, where we assume the query contains visual data such as an image and we want to retrieve relevant information in the form of texts (a database composed of textual data). First, we encode all texts in our database using path B, and then, we send the query image using path A. Then, we can compare the embeddings of the query image with the embeddings in our text database in a semantic way and retrieve the most similar ones. Most previous works report results in only one or two retrieval tasks. Instead, in our

work, we evaluated all cross-modal retrieval tasks using the same trained model. These tasks include Img2Img (also known as Query-by-Example), Txt2Txt, Txt2Img (also known as Query-by-String), and Img2Txt (also known as image captioning). As an example, the Txt2Img refers to the task where the queries are texts and the retrieved samples are images. As a performance measure of the ranking of retrieved results, we use the mean average precision (MAP), which is a standard evaluation metric in information retrieval. We calculate the MAP on a different number of samples to retrieve. The MAP 100% corresponds to the case when we use all the samples in the test set (see Appendix A.1.4 for the definition of the MAP metric). Therefore, for the Wikipedia database, the maximum size is 693, for Pascal, it is 400, and for MIR-Flickr-25k it is 2,000 in the setting (1) [283].

**Ablation study.** We studied the impact of two factors in the performance of the model: the distance metrics used to retrieve documents and the quality of embeddings taken from different layers in our model. The first evaluation is focused on identifying good distance metrics to retrieve points in the multimodal feature space. We compared the MAP results in the evaluated tasks on the Wikipedia database by using different functions and evaluated at the first 50 results (AP@50): Kullback-Leibler divergence (KL), Euclidean distance, and normalized correlation (NC), with average performances for each distance for the fourth tasks of *0,4424*, *0,4513*, and ***0,4917*** respectively. Since normalized correlation had the best performance in almost all experiments, we use it as the similarity metric in the rest of our evaluation (see Table 4.2).

Table 4.2: MAP using different distance metrics on Wikipedia Retrieval database and evaluated at the first 50 retrieved samples (AP@50). The results are displayed for the 4th tasks of cross-modal retrieval along with the average for each distance metric evaluated.

| **Task** | **Measures** | | |
|---|---|---|---|
| | **KL** | **Euclidean** | **NC** |
| **Img2Txt** | **0,4415** | 0,4367 | 0,4359 |
| **Txt2Img** | 0,3421 | 0,3823 | **0,4904** |
| **Img2Img** | 0,3769 | 0,3764 | **0,3995** |
| **Txt2Txt** | 0,6092 | 0,6101 | **0,6413** |
| **Avg** | 0,4424 | 0,4513 | **0,4917** |

The second evaluation is focused on the quality of the embeddings obtained from different lay-

ers. We compute embeddings for the visual and textual modalities using the corresponding layer in the model: Dense1 for images, Dense2 for text, and layers Dense3, Dense4, or Dense5 to obtain a fused representation after the point-wise multiplication layer (see Figure 4.1, Page 62). Each evaluation is made independently (only one layer is used at each time). We had the hypothesis that a multimodal network can generate better embeddings in deeper layers, where it has the opportunity to learn higher-level representations of the combined information. The results for the first 3 tasks are consistent with our hypothesis, the performance increases when we compute the embeddings from deeper layers (Dense1/2 < Dense3 < Dense4 < Dense5), with average performances for the 4 tasks of 0,3945, 0,3978, 0,3908 and **0,4080** in the case when we use the 100% of samples (see Table 4.3).

Table 4.3: MAP comparison from features computed from different layers on Wikipedia database using the normalized correlation distance and the 100% of samples in the database.

| **Task** | **Layer** | | | |
|---|---|---|---|---|
| | **D1(V) / D2(T)** | **D3** | **D4** | **D5** |
| **Txt2Txt** | 0,5615 | 0,5662 | **0,5672** | 0.5671 |
| **Img2Img** | 0,2555 | 0,2707 | 0,2733 | **0,2831** |
| **Img2Txt** | 0,3792 | 0,3681 | 0,3755 | **0,4221** |
| **Txt2Img** | 0,3821 | **0,3863** | 0,3773 | 0.3600 |
| **Avg** | 0,3945 | 0,3978 | 0,3908 | **0,4080** |

**Comparison with state of the art: Wikipedia Database.**

In this part, we present a comparison of results for cross-modal retrieval tasks for all databases. We report the state-of-the-art results under the same protocol of experimentation but covering different methodologies. Previous works have not reported results for uni-modal tasks, (similar to results in Section 4.3.3.1), thus, we used BERT (for text), and ResNet (for images) as reproducible baselines to compare against. Up to our knowledge, we are the first to report Txt2Txt results in these databases, and we also report results for Img2Img in the same way that few works have done before. By reporting results for these two tasks, we aim to establish baselines for future works too.

Tables 4.4 and 4.5 report MAP results obtained for different tasks for Wikipedia retrieval database. Table 4.4 presents the MAP performance when we consider the top 8, 50, and 500 results in the

Table 4.4: Mean Average Precision (MAP) on the Wikipedia retrieval database under different retrieval tasks using the first K results in the ranking (8, 50, and 500). For each task, we present the results of prior work under the same configuration of our evaluated approach. Results with "-" were not reported by the authors.

| **Task** | **Method** | **K** | | |
|---|---|---|---|---|
| | | **8** | **50** | **500** |
| **Txt2Txt** | BERT ([285]) | **0,735** | 0,60 | 0,406 |
| | DME (ours) | 0,681 | **0,641** | **0,576** |
| **Img2Img** | CMSTH ([123]) | - | - | **0,449** |
| | ResNet ([87]) | 0,433 | 0,323 | 0,193 |
| | DME (ours) | **0,455** | **0,399** | 0,299 |
| **Img2Txt** | CMSTH ([123]) | - | - | 0,337 |
| | RE-DNN ([286]) | 0,351 | 0,281 | - |
| | DME (ours) | **0,451** | **0,435** | **0,426** |
| **Txt2Img** | CMSTH ([123]) | - | - | **0,387** |
| | RE-DNN ([286]) | 0,23 | 0,241 | - |
| | DME (ours) | **0,489** | **0,49** | 0,382 |

ranking list. Our model exhibits the best performance in the case of top 8 and top 50 results in almost all tasks, which are the most useful cases for users in real-world conditions. The lowest performance, in general, is obtained when searching images with visual queries, which is a challenging task. However, our multimodal approach can improve performance in the top results of this task, showing the benefits of our general-purpose fusion framework.

An alternative evaluation protocol uses 100% of the ranked results instead of only the top K when computing MAP. Previous work using the Semantic Correlation Matching (SCM) reported results following this protocol [107, 118]. The SCM approach finds a semantic space that considers correlated multimodal projections, and we used it as a baseline under this protocol. To analyze the statistical significance of our approach we run a t-test comparing our model against the baseline using an average of 18 runs, with 15 different queries each in the Img2Txt and Txt2Img tasks (see Appendix A.1.4 for a definition of the t-test). In the terms of statistical significance of the results, the null hypothesis is that our approach has no significant difference concerning the baseline. It was rejected in both cases with significance threshold 0,05 (p-values: Img2Txt = 0,0001, Txt2Img = 0,036), underlining the reliability of the results. Previous works do not explicitly fuse modalities ([287]) but instead use the textual modality for supervising the

Table 4.5: Mean Average Precision (MAP) on the Wikipedia retrieval database for different retrieval tasks when evaluating 100% of samples in the ranked list. The best two results are highlighted in green (first) and orange (second). For each one of the tasks, we put the results of methods in the state of the art, under the same configuration of our approach.

| Task | Method | MAP (100%) |
|---|---|---|
| **Txt2Txt** | BERT ([285]) | 0,3873 |
| | DME (ours) | 0,567 |
| **Img2Img** | ResNet ([87]) | 0,195 |
| | DME (ours) | 0,283 |
| **Img2Txt** | SCM_1([107]) | 0,277 |
| | RE-DNN ([286]) | 0,340 |
| | MDCR ([116]) | 0,435 |
| | Marginal SM ([287]) | 0,332 |
| | SCM_2 ([118]) | 0,362 |
| | Self_Supervised ([139]) | 0,391 |
| | LCALE ([140]) | 0,367 |
| | DME (ours) | 0,422 |
| **Txt2Img** | SCM_1 ([107]) | 0,226 |
| | RE-DNN ([286]) | 0,352 |
| | MDCR ([116]) | 0,394 |
| | Marginal SM ([287]) | 0,241 |
| | SCM_2 ([118]) | 0,273 |
| | Self_Supervised ([139]) | 0,434 |
| | LCALE ([140]) | 0,357 |
| | DME (ours) | 0,360 |

visual representation learning. For example, a recent self-supervised model addressed the task of cross-modal retrieval, training an LDA model of text and topics as prediction target for the CNN that processes the visual modality [139]. This model gets the best performance in the Txt2Img task but it is not competitive in the Img2Text task showing how the models can be optimized for only one task instead of being truly multimodal.

The most similar approach to our DME model is the Regularized Deep Neural Network (RE-DNN) ([286]). This model has three parts: the visual, textual, and multimodal subnets; each one pre-trained separately before fusion. RE-DNN is not trained end-to-end, and in contrast to our model, the textual network does not use sequential word embeddings. Notably, RE-DNN does not use a pointwise multiplication layer for fusion but instead relies on the concatenation of features. The results show that our model surpasses their performance in all tasks, indicating

that the choices of our model yield enhanced fusion performance. The analysis of cross-modal results in Tables 4.4 and 4.5 confirm that ours is generally the best performing approach. The one case where it is not the best is when MAP is computed on 100% of the ranked results (Table 4.5). However, our result is very competitive and has improved performance in all other metrics and tasks. Our approach may also be more computationally efficient than models like MDCR that learn two projection functions while we aim to learn only one generic multimodal mapping.

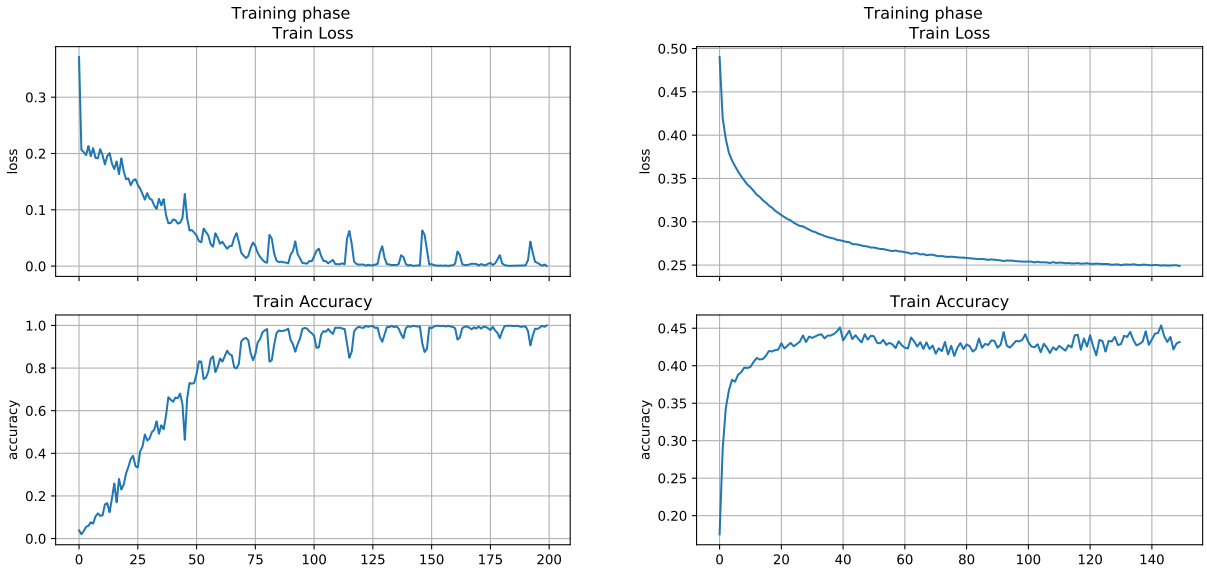
Table 4.6: MAP comparison on Pascal Sentences and MIR-Flickr-25k databases under different retrieval tasks and by using the first K results (10, 100, and 200).

| **Task** | **Method** | **Pascal** | | | **MIR-Flickr-25k** | | |
|---|---|---|---|---|---|---|---|
| | | **10** | **100** | **200** | **10** | **100** | **200** |
| **Txt2Txt** | BERT | 0,449 | 0,274 | 0,230 | 0,870 | 0,793 | 0,764 |
| | DME (ours) | **0,556** | **0,486** | **0,465** | **0,883** | **0,839** | **0,823** |
| **Img2Img** | ResNet | 0,584 | 0,405 | 0,376 | 0,946 | 0,897 | 0,874 |
| | DME (ours) | **0,586** | **0,495** | **0,486** | **0,962** | **0,948** | **0,943** |
| **Img2Txt** | DME (ours) | 0,532 | 0,474 | 0,458 | 0,942 | 0,915 | 0,899 |
| **Txt2Img** | DME (ours) | 0,523 | 0,470 | 0,451 | 0,919 | 0,893 | 0,876 |

**Comparison with state of the art: Pascal and MIR-Flickr-25k databases.**

The accuracy in the test set (0,7561%) is expected when comparing to the accuracy obtained in training (≈aFigure 4.5 presents the learning curves for the loss and accuracy in training for the multimodal framework for Pascal Sentences and MIR-Flickr-25k databases. In both cases, the learning curves converge (≈ 100 for Pascal sentences and 45 for MIR-Flickr-25k). For Pascal sentences, the curves present more disturbances in the convergence requiring more epochs to converge. This may be explained by the smaller size of the database (only 600 training samples). Also, the different sentences associated with each image may delay the learning of the optimal embeddings due to the subjectivity of each annotator. 0,99%). This also shows the absence of over-fitting.

Tables 4.6, 4.7 and 4.8 present the MAP performance for Pascal Sentences and MIR-Flickr-25k databases. Table 4.6 presents the results when MAP is evaluated by taking the top 10, 100, and 200 retrieved results (we choose these values as being suitable for both databases) showing good results, especially for the MIR-Flickr-25k database. We start comparing our performance against BERT (for text) and ResNet (for images) as uni-modal baselines methods.




Pascal sentences MIR-Flickr-25k

Figure 4.5: Learning curves for loss and accuracy in training, for the databases of Pascal sentences and MIR-Flickr-25k using the multimodal framework.

Tables 4.7 and 4.8 present results of methods with comparable setups for both databases. We obtain competitive results in the Pascal Sentences database, and we reach the best performance in the MIR-Flickr-25k database. Our model reaches state-of-the-art performance in the MIR-Flickr-25k database when compared with recent cross-modal retrieval approaches, including TFNH [175], a hash-based method and CPAH [178]. We evaluated the retrieval performance in the Txt2Img task when gradually increasing the length of the text sequence in the query by adding more sentences in the Pascal database. An illustrative example is presented in Figure 4.6, Page 79. Note that the relevance of the top 5 results changes with different textual queries, and when more descriptions are available our system can retrieve more accurate results. We achieve these results with a single model that can handle variable sequence length as inputs for computing the multimodal representation.

Figures 4.7 and 4.8 present visual retrieval results for all databases. In Figure 4.7, we can observe that retrieved results are semantically correlated with the query, showing that embeddings generalize well for the first top results, for the Wikipedia Retrieval database. Figure 4.8 shows top 3 results for textual and visual queries for A) Pascal Sentences, and B) Mirflick-25k. In the

Table 4.7: MAP comparison on Pascal Sentences database. The best two results are highlighted in green (first) and orange (second). This database contains 1,000 samples of images associated with several sentences and descriptions of their content (approximately 5 sentences per image) from 20 classes, 50 images per class. We follow the standard partition of 600/400 (i.e., 30/20 samples per class) for training and test samples.

| Task | Method | MAP (100%) |
|---|---|---|
| **Txt2Txt** | BERT ([285]) | 0,196 |
| | DME (ours) | 0,453 |
| **Img2Img** | ResNet ([87]) | 0,361 |
| | DME (ours) | 0,436 |
| **Img2Txt** | MDCR ([116]) | 0,455 |
| | Marginal SM ([287]) | 0,222 |
| | Self_Supervised ([139]) | 0,326 |
| | LCALE ([140]) | 0,414 |
| | DME (ours) | 0,429 |
| **Txt2Img** | MDCR ([116]) | 0,471 |
| | Marginal SM ([287]) | 0,173 |
| | Self_Supervised ([139]) | 0,360 |
| | LCALE ([140]) | 0,394 |
| | DME (ours) | 0,425 |

case of the Img2Txt query, the retrieved results can be seen as a method of image captioning. For Pascal Sentences, the system is retrieving all the sentences associated with the retrieved samples, which can be used also in a separate way to describe the content of the image.

### 4.3.4 Visual analysis of embeddings

In this section, we present visually the quality of the embeddings for the Wikipedia Retrieval database. Table 4.9 presents the results for t-SNE [288] when it is computed by using visual, textual and both (multimodal) embeddings from the Wikipedia retrieval database, computed from different layers of the model. We can observe different things: first, the discrimination among the classes for all the modalities is higher when the embeddings are computed using a deeper layer (in this case, the layer D5), this means, the model is learning a better representation (knowledge) when using the information from both modalities. The second observation validates the quantitative results obtained in previous results, where the textual modality has better performance (for the embeddings in layer D5, the textual modality learns better to discriminate

Table 4.8: MAP comparison on MIR-Flickr-25k database. This database consists of 25,000 images with up to 38 semantic concepts. Two partitions are tested: (1) 2,000 testing samples [178, 283], and (2) 95% of the data as training, 5% for test [175]. The best two results are highlighted in green (first) and orange (second).

| Task | Method | MAP (100%) |
|---|---|---|
| **Txt2Txt** | BERT ([285]) | 0,717 |
| | DME(2) | 0,759 |
| **Img2Img** | ResNet ([87]) | 0,793 |
| | DME(2) | 0,868 |
| **Img2Txt** | RCCA(1) ([283]) | 0,695 |
| | CPAH(1) ([178]) | 0,791 |
| | TFNH(2) ([175]) | 0,688 |
| | DME(1) | 0,725 |
| | DME(2) | 0,809 |
| **Txt2Img** | RCCA(1) ([283]) | 0,694 |
| | CPAH(1) ([178]) | 0,789 |
| | TFNH(2) ([175]) | 0,761 |
| | DME(1) | 0,731 |
| | DME(2) | 0,805 |

between classes than the visual modality, where the discrimination is lower).

The distribution of samples in the test explains why there are more misclassified samples from some classes (for example, for the class "Warfare", there are many more samples, some of which are misclassified - shown with blue dots in the graphs of Table 4.9, Page 82). Also, the class "History" -represented with red dots- is a challenging class because it can be highly related to all the other classes, which makes it more difficult for the model to infer the differences among them. Another interesting observation is how visible the relations between the classes are, particularly in the textual embeddings of layer D5, for instance showing connections among Biology with Geography & places, Literature & theatre with Media and Music, and Royalty & nobility with Warfare.

## 4.4 Discussion

We have analyzed and evaluated an approach that can process visual and textual modalities in a document and learns a semantic relationship between them. Inspired by visual question

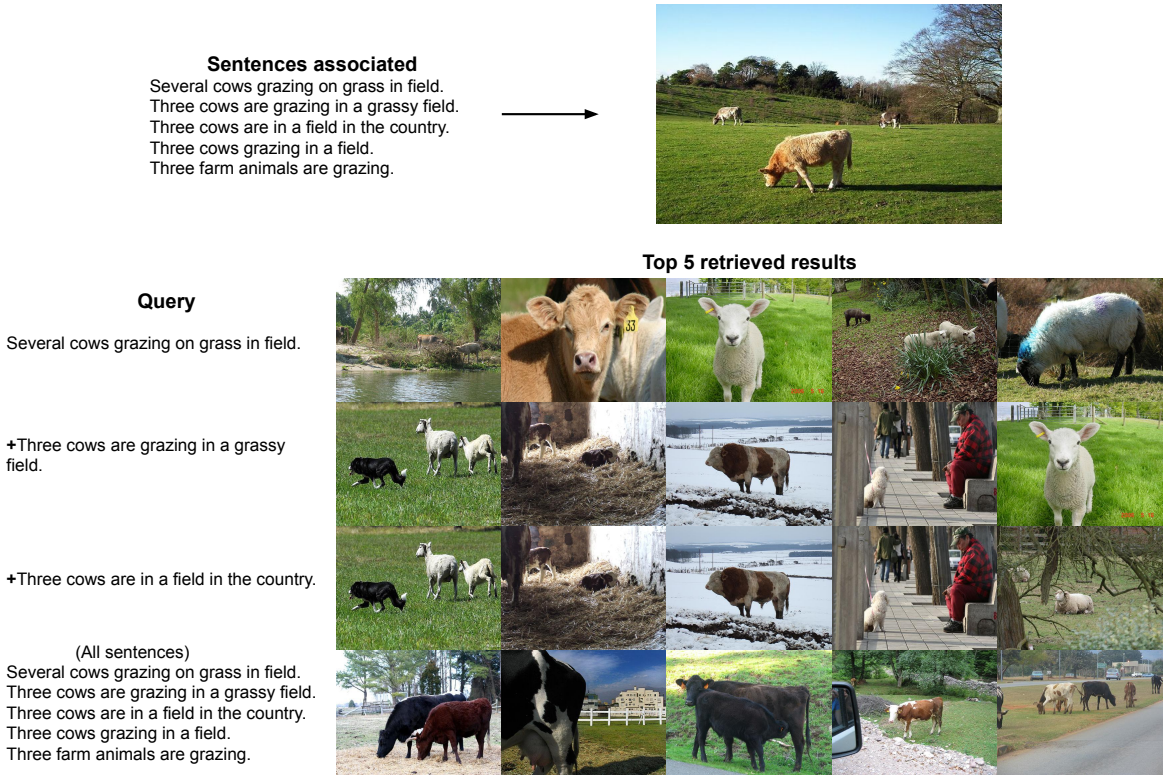



Figure 4.6: Retrieval examples from the Pascal sentences database: at the top, the original sample with its sentences and associated image, at the bottom, retrieved images by gradually adding more sentences to the text query.

answering architectures, this approach can help in learning combined representations to build effective cross-modal retrieval systems. Results indicate that these architectures can be retrained for cross-modal search without the need for special layers or additional model developments, making them ideal as a baseline for future multimodal indexing research. The results of our experiments underline two positive aspects of the evaluated model. First, performance does not degrade after combining the two modalities (one of the modalities can be noisier than the other). Second, our goal was to investigate the potential of a single model to retrieve relevant documents in cross-modal tasks without being optimized exclusively for any one of them. We performed an extended analysis of our approach, and without the need to fine-tune the model or results for every single retrieval task, we achieved robust results in 3 databases for all tasks and even reached state-of-the-art performance in some of them.

Having these promising results, we propose to adapt the system to perform a much more specific

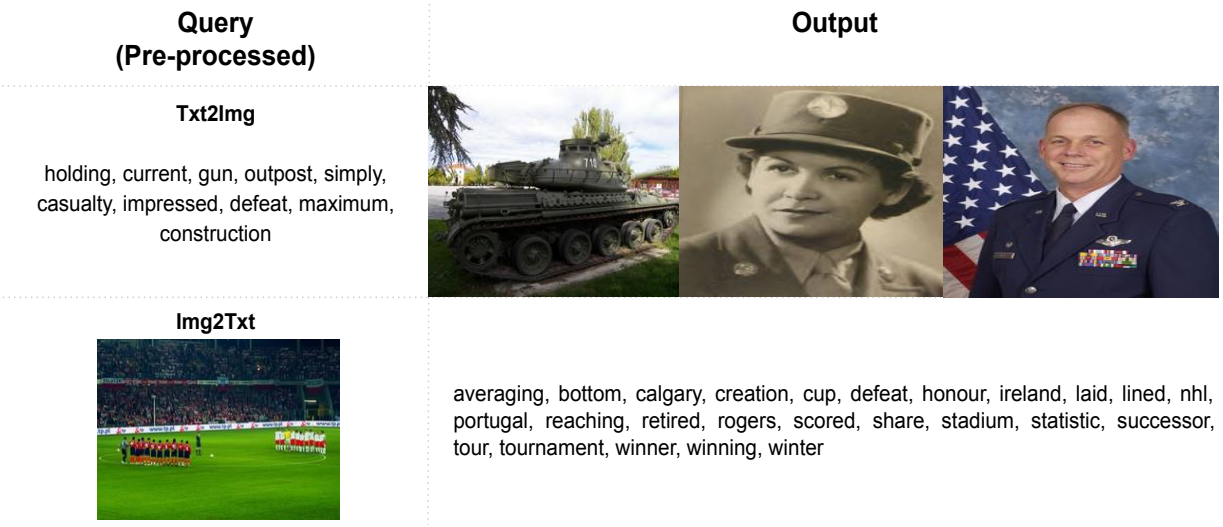
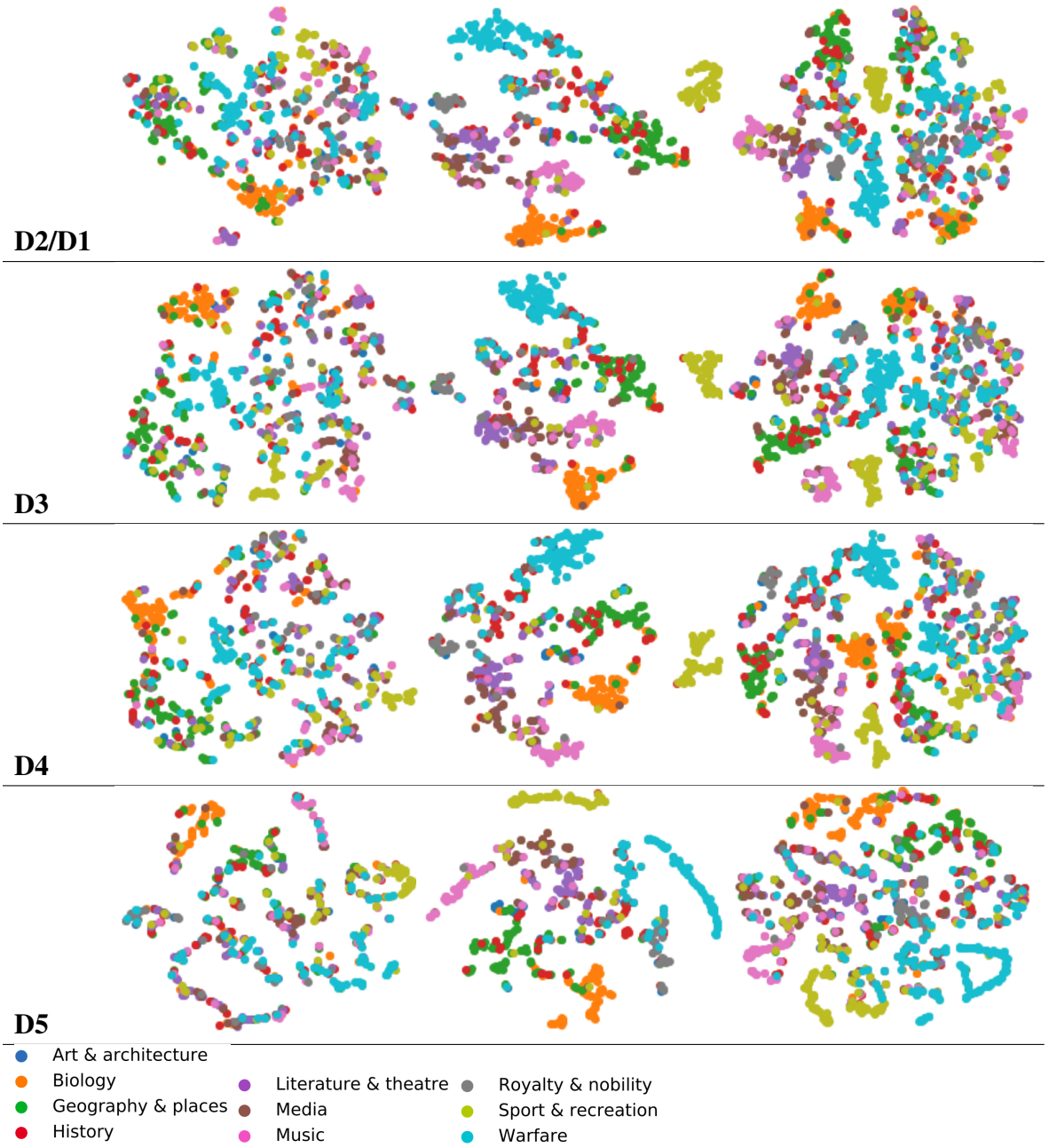



Figure 4.7: Retrieval examples for Wikipedia Retrieval database: when text|image is the input query, the cross-modal retrieval system aims to retrieve data from a different modality (i.e. image|text). In the first case, the text query shows the resulting tokens after the pre-processing step.

task between images and textual data. The task of scene-text visual question answering (ST-VQA) has been recently proposed as a new challenge in the context of multimodal content description. The aim is to teach traditional VQA models to read the text contained in natural images. Here, we need to perform a semantic analysis between the visual content and the textual information contained in associated questions to give the correct answer. Hence, we found that this task highlights the importance of exploiting high-level semantic information (text) present in images. This allows us to study the relationships between images and textual data for which we have the security of their existence. In the next chapter, we present the developments and evaluations proposed with this new task.


Figure 4.8: Sample of queries in a multimodal retrieval system for Pascal Sentences database: when text|image is the input query, the cross-modal retrieval system aims to retrieve data from a different modality (i.e. image|text).

Table 4.9: T-SNE visualizations of embeddings for different layers. From left to right: visual, textual and multimodal embeddings from Wikipedia retrieval database. The colors represent the different classes associated with each sample image-text pair. D and its corresponding numbers refer to the layer used to create the embeddings. We can see how the embeddings are better separated per class when deeper multimodal layers are used to create them.



# CHAPTER 5

# Semantic Text Recognition via Visual Question Answering

## 5.1 Motivation

In this chapter, our goal is to evaluate the robustness of the model proposed in Chapter 4 for a more recent task called scene-text visual question answering (ST-VQA). This task is based on two separated tasks pursuing different goals. First, the task of scene text recognition only seeks to recognize text in wild images, i.e, images taken from different scenes such as streets. And secondly, the VQA task where strong reasoning is required about some target visual information. This information is addressed by asking open-ended questions with some possible and acceptable variations in the answer. Current VQA models fail when the required target visual information is within textual data. This joint task (ST-VQA) has not received the required attention, due to the lack of databases targeting it, as well as all the additional challenges posed by the separate tasks of text recognition and VQA. ST-VQA requires that the model recognizes and interpretes the textual information in a scene (text contained in signs, posters, or ads in the image) so as to give the correct answer. Although deep learning has been used with acceptable

accuracy results of $\approx$ *70%* for traditional VQA tasks [289], when solving the ST-VQA problem the accuracy drops to $\approx$ *27%* [207] demonstrating the challenge at hand. For example, in Figure 5.1 - F, in a traditional VQA system, the question could be "What is the object in the image?", and to answer it, the system would not require to read any text. In an ST-VQA task, the system must read the correct textual data in the image, so as to answer the question, as in the example "What is the brand name of the toothpaste?". Several challenges arise in the context of this task. For example, understanding the type of question is required in order to filter the set of possible answers. Figure 5.1 presents samples of triplets *(Image, Question and Answers)* from TextVQA database [207]. Each sample contains an image, a question associated, and the ground truth list of answers given by 10 human annotators. We present examples for unclear annotation cases *(A-E)*, where, for instance, the answer given by the annotator is not correct (*case C*) or falls into the "Yes"/"No" category *(case D)*. In the correct cases *(case F)*, the answer is, as expected, provided by text present in the image.

Also, the representation of the answer space becomes critical because it can contain unlimited words in any possible language. This makes infeasible the establishment of a fixed pool of answers and yields the "out-of-vocabulary" (OOV) problem (words not contained in the pool of answers). Other challenges are related to the detection and recognition of the text present in the visual data in the wild or from natural scenes, which remains a challenge for current models facing this problem because of all the variations they present [290]. One of the most difficult tasks, even in traditional VQA systems [291] is the reasoning required to resolve spatial and visual references that involves understanding the question and the visual information at the same time. Recapitulating, this task represents a suitable recent challenge that combines images and text and that demands a high level of reasoning and semantic analysis. Also, being our previous model inspired by VQA architectures, it fits the main requirements for this task. For such a semantic description task, many related issues need to be addressed.

As we mentioned before, this task was recently introduced in the document analysis and recognition community which is why the state of the art and the available databases, are limited. The problem of having limited databases makes the evaluation and model comparison harder. Then, the first type of works in the state of the art addresses the problem of lack of databases

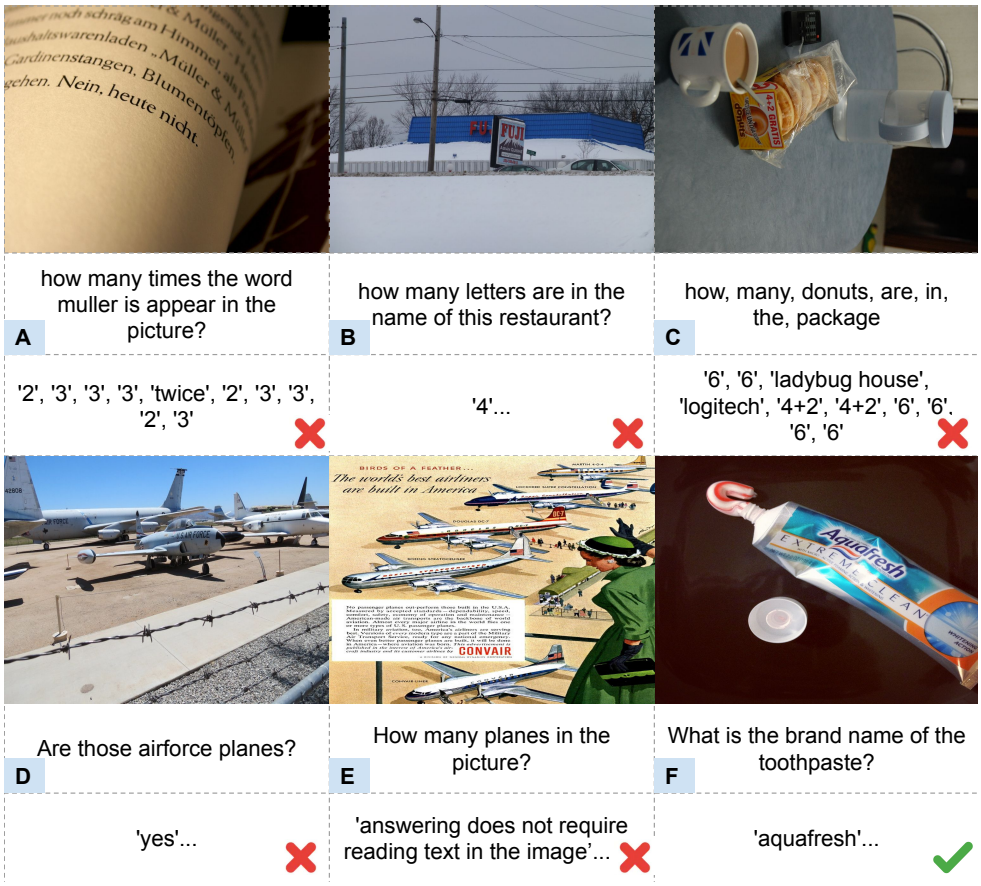



Figure 5.1: Samples of triplets (Image, Question, List of answers given by 10 annotators) from TextVQA database of wrong (A-E) and correct (F) annotations to address the problem of ST-VQA.

by proposing data collections that follow the requirements {image + question (targeting some textual information) + answer} of the ST-VQA task. The two first and most popular databases were then proposed by Biten et al. [211] with the ST-VQA database and second, by Singh et al. [207] with the TextVQA database. Several other databases have been proposed with more specific purposes such as images of book covers [208] and diverse sets of wild scenes [58, 209].

We will now introduce the existing approaches to this task. Singh et al. [207] proposed a framework called LoRRA based on three networks: the image, the question, and the OCRs extracted from the image using another model. Contrary to our work, the multimodal component is nonexistent. They replace it with a classifier layer that connects the independent modality networks. Their main proposition is based on a module that extracts the answer from the avail-

able set of OCRs recognized in the image (the copy module). The decision is based on the predicted scores for the original set of answers. We take into account its proposition in our evaluations and propose other representations for the answer space. This way, we do not rely only on OCRs extracted from the image, but instead, take advantage and used them as complementary information (following our previous propositions of using additional information that may help to improve the learning of the model). Many methods are based on the performance of current OCR engines. For example, in [208] the proposal is based on Tesseract, a well know OCR engine for text documents, and a VGG which is a well-known neural network define in [86]. The evaluation is however led on a very specific database of book covers with accuracy performances of 0.58%. Other approaches to the task involve attention mechanisms [219, 220]; Let's recall that attention mechanisms have shown to improve performance in computer vision models based on deep architectures by learning to focus on the regions of the image that are salient. Hang et al. [219] implement an attention mechanism that works by feeding the model with visual features extracted from the OCR boxes (not the text). The OCR boxes are computed using a specialized OCR model. In our strategy, we also implement attention by including not only the OCR boxes but combinations of complementary information that may help the model to attend to some parts of the image.

In summary, the state of the art proposes frameworks composed of successful feature extractors and attention mechanisms. We notice a lack of proposals concerning the representation of the answer space. In general, these are based on traditional bag of words [215], or improved versions of it including a set of dynamic spaces [207], i.e., to add additional dimensions (spaces), different for each sample, to those of the bag of words.

As this task is very recent, our first experiments were to perform ablation studies to evaluate the impact of the different components in the framework. Therefore, in Section 5.2, we present an analysis of feature extraction methods with a focus on the question embeddings. We consider the impact of the representations of the critical data. The images are represented with successful neural networks such as ResNet [87]. Due to the type of task (reading some textual data in the image), understanding the type of question (yes / no, or specific text contained in the image), as well as the target information required in it, further analysis are required to find suitable repre-

sentations. For this reason, to represent the questions, we propose to compare context-based and context-free textual embedding models. To represent the answers, we propose to evaluate an n-gram based representation. This n-gram representation is more flexible in representing words outside the English vocabulary.For our analysis, we rely on the two most popular databases: ST-VQA and TextVQA. In the first evaluation presented in Section 5.2, we only use the ST-VQA database and focus on the specifications of the challenge described in [292].

In the second part (see Section 5.3), we complement our ablation feature extraction study by the inclusion of a complementary module that feeds the model with additional information (such as available OCR text). We evaluate the performance of including additional data to train the system in the form of a complementary network representing embeddings from textual and visual data. Notice that we are using the OCRs as complementary information that may help to improve the learning of the model. On the contrary, it is the model that learns to recognize the target spatial references where some text may be located. We also extend our analysis by changing the answer representation to a bag-of-word representation with the option of augmenting its dimensionality with a dynamic set of spaces. We expose drawbacks related to the copy module which is the state-of-the-art solution, and we propose to use a second metric to compute the scores for the dynamic spaces so that the copy module can take advantage of texts that are not perfectly recognized by the OCR system. In Section 5.3), we make use of the second most popular database, the TextVQA database. For this database (unlike the ST-VQA database), there are state-of-the-art results for the validation set. Therefore, we can make a fair comparison with the performance obtained by our method.

## 5.2 Initial framework

As we mentioned before, our first approximation consists of performing ablation studies to evaluate the impact of the different components in the framework. Therefore, in this section, we present an analysis of feature extraction methods with a focus on the question embeddings and the representation of the answer space using n-grams. For the images, we make use of successful neural networks such as ResNet [87]. For the questions, we compare context-based and context-free textual embedding models. For the answers, we propose an n-gram-based representation.

We make use of the ST-VQA database to evaluate the model performance. Next, we present a general overview of the framework followed by our experimental evaluation.

### 5.2.1 Modules

The framework consist of fundamental modules that pre-process the information in a ST-VQA task. It is as follows: a textual network that processes data expressed as questions and that tells the system what the required information is, a visual network that processes data expressed as images and that contains the target information, a multimodal central module that fuses both sources of information and outputs the answer. Figure 5.2 presents the proposed architecture. We describe the modules next. The input data are visual (image) and textual features (question), while the target output is an n-gram vector representing the answer.

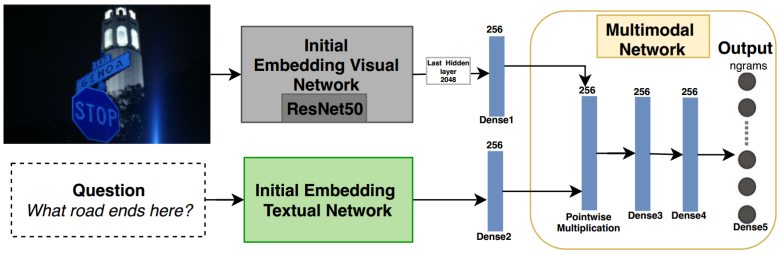



Figure 5.2: Overview of the first framework: The visual features are extracted using a pre-trained CNN, while textual features are extracted using a sequential model. Then, a fusion layer is applied to connect with a central module. The central module is composed of the fusion layer and fully connected layers. For the representation of the answer space, we use a vector containing n-grams extracted from the set of possible answers in the database ($\approx$ 550 n-grams). These n-grams are set to 0 or 1 representing their absence or presence in the word answer.

#### Visual embedding

The visual network is in charge of encoding the images by extracting the most relevant information from them. The main task here is to get good initial features for the images, therefore, instead of re-training a whole model, we leverage the learned properties from a bigger and well-tested architecture, by using a pre-trained model - ResNet50 [87] - in millions of images. This network was trained using the same type of natural images, present in our evaluation database, which makes it suitable to compute the initial representations. We take the last hidden layer with

a 2,048-dim. This 2048-dim vector is the visual input to the system.

**Question embedding**

In the context of natural language processing, there exist, powerful models to embed a text with or without context. We test both scenarios, to help us understand which one is the most suitable for our needs, and to analyze if, in the case of the ST-VQA task, the inclusion of context-free vs context-based embedding models is helpful. We evaluated two text embedding models: GloVe [76] and BERT [79]. GloVe is a well-known context-free unsupervised learning method for obtaining vector representations for words. It provides a global representation for each word present in the question database. It is based on statistics of word occurrences in a corpus. The model used was trained over a set of million of Wikipedia articles. On the other hand, BERT is one of the most recent context-based state-of-the-art models to compute textual embeddings and it provides us with different levels of representation. It is a deep network bidirectional model, trained from unlabeled text by jointly conditioning on both left and right context (it remembers the history). The pre-trained model used to compute the embeddings was trained using a large database (Wikipedia + BookCorpus [293]). Context-based models generate representations for each word that are based on the other words in the sentence, making the embeddings more robust. With BERT, we can test two different scenarios for generating the embeddings, we called these word-sentence embedding and sentence-embedding. The word-sentence embedding consists of generating an embedding for each token in the question. This means that the embedding for the question for one sample will be of dimensions $Sequence\ length \times 768$ (768 being the dimension of the BERT embeddings). The second scenario called sentence-embedding refers to applying a pooling strategy from some layers of the model, for generating the final embedding vector for the question as a whole, instead of doing it for each token in the question. As a pooling strategy, we are using the reduce mean strategy, which consists in taking the average of the outputs of the last hidden layer of the model.

**Central module**

The central module is the network in charge of fusing and learning the relations between the information coming from both modalities, textual and visual, and outputs target information

that we can interpret and finally, obtain an answer. We use a multiplication layer to fuse both networks (see Equation 5.1), followed by three fully connected layers with the last one seen as our n-gram representation layer (the answer transformed in the n-gram vector representation).

$$(V \circ T)_i = (V)_i (T)_i \; for \; all \; 0 \leq i \leq n \; , \tag{5.1}$$

where V = Image feature, T = Question feature, n = dim(V) = dim(T).

**Answer representation & Prediction**

To create the answer representation, we propose to use a concatenation of n-gram attributes (see Figure 5.3). We believe this is a suitable representation of the answer for two main reasons: the representation is flexible in the sense that it is not restricted to only learn exact sequences of text. On the contrary, it aims at finding smaller similarities, the data have some common n-grams. Because an image can contain several n-grams, yielding multiple output neurons to be 1, the task can be seen as a multi-label classification, for this, we use a Sigmoid activation function:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{5.2}$$

which is applied to every element of the output vector. We use a binary-cross-entropy loss function (see Equation 5.3) and train the model in an end-to-end configuration.

$$BCE(p_x) = -(y_x \log(p_x) + (1 - y_x) \log(1 - p_x)) \tag{5.3}$$

where $p_x$ is the prediction of the model for the input $x$, and $y_x$ is the ground truth label of the sample.

**Prediction.** We follow the evaluation protocol described in the website of the challenge described in [292]. Once we have our model trained, we can compute the n-gram representation for each image-question pair sample. Figure 5.4, Page 92 presents the complete protocol followed for the calculation of the score using the predicted embeddings.

We compute the distances between the predicted n-gram representation for our input sample and

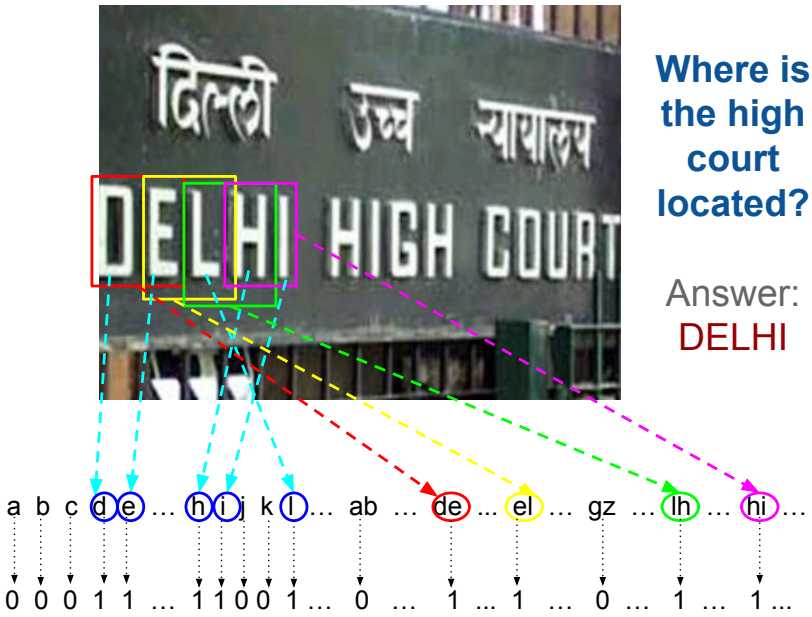



Figure 5.3: Example of the construction of the answer vector for an image. The uni-grams and bi-grams contained in the answer are used to activate the positions in the vector representation. The other positions remain as 0. The n-grams are extracted from the set of possible answers in the database (words can be from different languages, however, the set of answers belongs more to the English vocabulary). If an answer contains more than one word, the n-grams are extracted using all of them. For example, if the answer in the example was "DELHI HIGH", both words are used to compute the n-gram representation. All the words are lower case first, then the uni-grams (0-1,a-z) and bi-grams are extracted and filtered. The final size of the vector is 550-dim.

all the n-gram representations from all the words present in our answers database. This answer database, as we mentioned before, contains all the words present in the image plus some extra words acting as distractors (words non-appearing in the image). After computing the distances, we use the word associated with the most similar n-gram representation as our output answer. Then, we can calculate the evaluation score based on Levenshtein distance, as follows:

$$score = 1 - (\frac{Levenshtein(out, ans))}{max(len(out), len(ans))}) \tag{5.4}$$

where, $out$ is our predicted answer, and $ans$ is the ground truth answer. The results are also reported by using the following rule for the score (provided by the challenge):

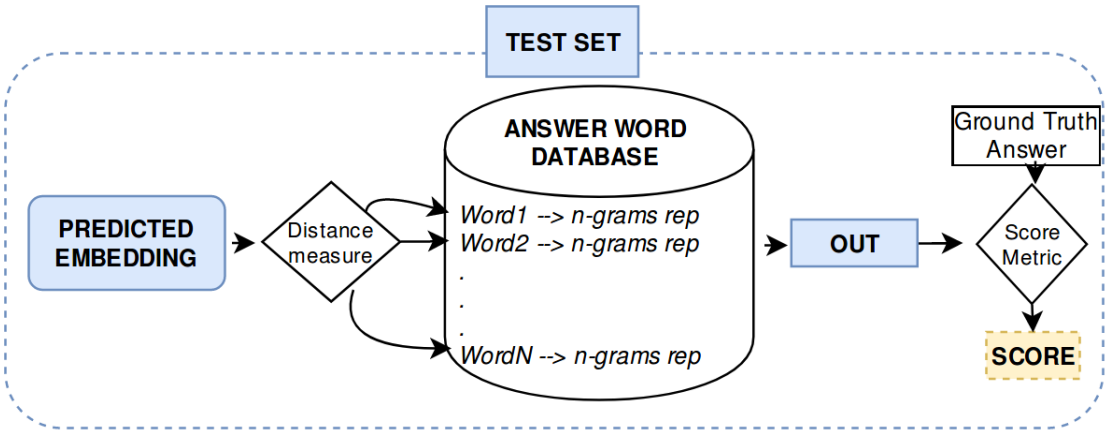



Figure 5.4: Protocol for the calculation of the final score: having the predicted n-gram representation for a sample, we can retrieve the string for the answer "OUT", by calculating the distances between the predicted embedding and all the n-gram representations from strings in the answer database. For example if there are 1,000 different answers in the database, there will be 1,000 n-gram embeddings, and therefore 1000 distances computed. The calculation is performed by using three distance metrics: Euclidean, cosine, and correlation. Then, we can get the string associated with the most similar n-gram representation. After we have the "OUT" string, we can compute the score based on the Levenshtein distance (see Equations 5.4 and 5.5) using our output answer string and the ground truth answer string.

$$trimmed\ score = \begin{cases} score\ if\ score \geq 0.5 \\ 0 \qquad if\ score < 0.5 \end{cases} \tag{5.5}$$

### 5.2.2 Experimental evaluation

In this section, we describe the database used in our explorations, followed by our implementation details and results obtained using the proposed approach.

#### ST-VQA database

As a case study, we use the database provided for the challenge "ICDAR 2019 Robust Reading Challenge on Scene Text Visual Question Answering" [292]. This is a recent database for ST-VQA, which contains 23,000 images with up to three questions/answer pairs per image. The images present in the database, are those normally found in everyday human activity, such as making a purchase, using public transportation, finding a place in the city, etc. The train set

consists of 19,000 images with 26,000 questions while the test set consists of 3000 images with 4,000 questions per task. We use the train set and separate it into two subsets: train ( 90%) and test ( 10%). We follow the evaluation protocol described by the organizers on their website, in the "Strongly Contextualized task". For this task, we have access to the bag-of-words appearing in each image plus some extra words acting as distractors, all of them conforms the answer database.

**Implementation details**

We used the ST-VQA database for this first set of experiments. For the training set, there are more than 16,345 different answers, 95% of them with 1, 2, or 3 tokens. As there are many different answers, the problem becomes more challenging, since, for any model to learn to discriminate among all the answers, it is required to pass a good set of different samples related with each one and with some variations, thus, the model learns to generalize as well. To tackle this, we use data augmentation only in the images (because it is the modality containing the target information) by applying a set of transformations in which the textual data contained in the image is not highly affected. We use transformations such as rotations (not bigger than 180 degrees), as well as cropping and morphological operations that can help to emphasize borders and therefore, the textual data. We did not use transformations such as mirroring because those types of transformations can reverse the order of the characters in the text data contained in the original image. Samples of transformations are shown in Figure 5.5, Page 94. We apply standard pre-processings to clean the question database such as tokenization, stemming and lemmatization. The minimum number of tokens for all the questions is 2, the maximum 25, and the average sequence length is 15 tokens, which we use to set as the maximum number of tokens. Around 98% of the questions start with a question "WH", 85% of them with the question word "WHAT". As we mentioned in Section 5.2, we test three different textual embedding scenarios: for the first one, we use the GloVe embeddings with a dimension of 300, the dimension of the embedding matrix is $15 \times 300$. For BERT, the dimension of embedding vectors is 768. Thus, for the first case, the word-sentence scenario, the dimension is $15 \times 768$, while for the sentence-embedding, it is 768 as the single dimension. For the calculation of the BERT embeddings, we use the tool bert-as-a-service [285]. For the answers, we apply the following process: we

obtained the set of unique answers ($\sim 16345$) and extract the uni-gram and bi-gram levels from all of them. There are 36 uni-grams taking into account only letters in the English alphabet and numbers. To select a good set of bi-grams to use, we select those with frequencies in the range [20, 1,000]. This selection contains 74.82% of all bi-grams, which balances the proportion of n-grams from the whole database vs the dimension of the final representation. Based on previous works, such as [294] whose representation is also based on n-grams, it is better to keep their number small. For this reason, we do not select more n-grams as three-grams because it increases the dimensionality 5 times. For example, a good set with frequencies between 10 and 100 leaves a final set of 2,387 three grams. Therefore, this option is not suitable. On average, all the samples have 10,92 n-grams, with a 553-dim as the final answer representation.

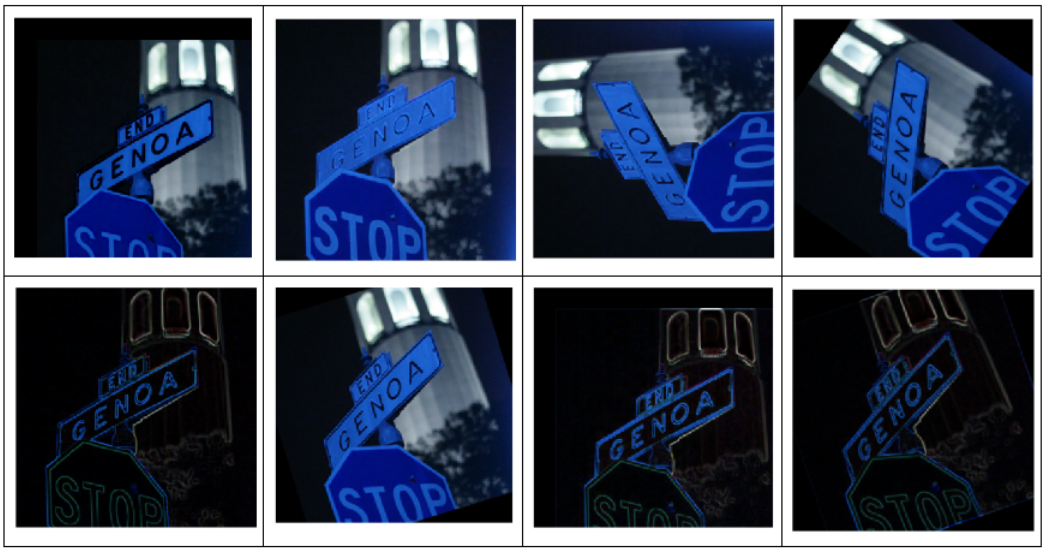



Figure 5.5: Examples of the transformations used for data augmentation.

**Results**

We describe the set of experiments performed. All the results are computed in the test set by using the evaluation metric presented in Equations 5.4 and 5.5. First, we present the results for the comparison of question embeddings. Table 5.1 presents the results for three different scenarios for question embeddings: using GloVe embeddings (context-free) and using the BERT model to compute word-sentence embeddings and sentence-embeddings (context-based). In this experiment, we notice that the first embedding method, GloVe, performs slightly better than the ones using the context. This result is explained in two ways: first, because the average number

of tokens per question is not very big. Second, because for most of the questions, the structure can be very similar, starting with "WH" interrogative words, as well as the surrounding words, and the target which is more related to finding the answer word. For example, for many samples, the question can start with *"What does the text written...?"* or *"What is written...?"*. This makes it more difficult to represent such small singularities when we consider that the questions have a very similar context.

Table 5.1: Comparison of textual embeddings for the questions.

| **Embedding** | **Score** | **Trimmed Score** |
|---|---|---|
| GloVe | 0,23093 | 0,120581 |
| Bert_per_word | 0,21814 | 0,117965 |
| Bert_per_sentence | 0,216221 | 0,114302 |

Having fixed the representation of the question by using GloVe vectors, we can continue with the model training. Figure 5.6 presents the training curves for the loss and accuracy in the ST-VQA database. Although both curves do not achieve optimal values, they converge smoothly and fast. The accuracy in training data shows the challenge ahead when working with very noisy databases on a task that requires extracting high-level semantic knowledge.

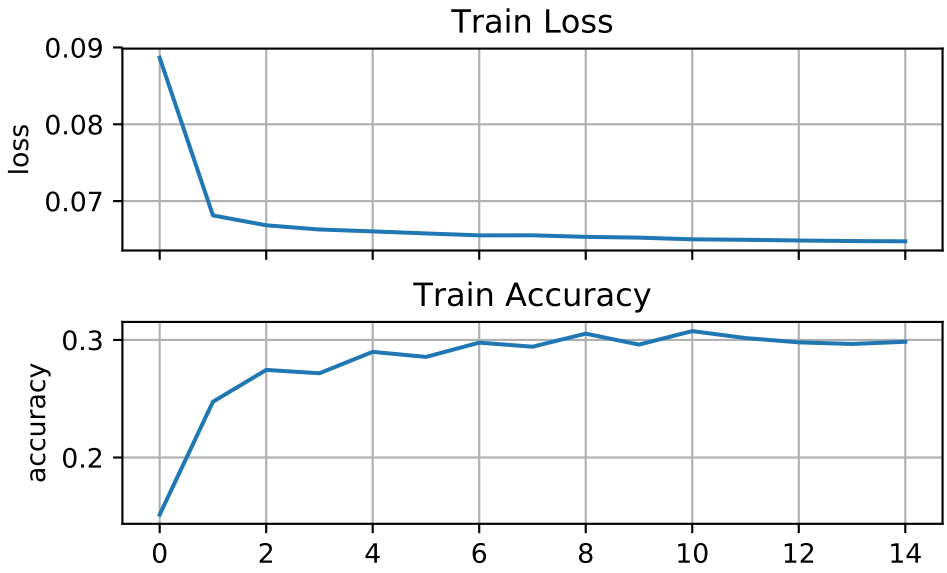



Figure 5.6: Learning curves for the loss and accuracy in training for the ST-VQA database.

As it is shown in Figure 5.4, Page 92, once the model is trained, we need to compute the distances

between the current predicted n-gram embedding and the n-gram vector representation from all the words in our answer database. For this reason, we evaluated 3 distance metrics that help us find the most similar vector representation from the vector representations in our answer database, and therefore, be able to obtain a final answer string to compute the score. Table 5.2 presents results for 3 different distance metrics: Euclidean, cosine and correlation. We use the Euclidean distance which has turned out to be the best metric.

Table 5.2: Comparison of distance metrics for calculation of output answer.

| **Distance** | **Score** | **Trimmed Score** |
|---|---|---|
| euclidean | 0,23093 | 0,120581 |
| cosine | 0,211453 | 0,097326 |
| correlation | 0,203372 | 0,070523 |

As our first approximation to address this task, and by having an overall performance of 23%, it is clear the system is not performing as we expected (see also Figure 5.7 for visual example results). Because no previous works are addressing this specific task, in the same database, and with many variables in hand, it is very difficult to determine the main cause of this low performance. These models also require to be trained with many different variations of each input pair and using thousands of iterations. Similar models addressing other tasks in the context of VQA train their models for many days using millions of data, which makes it harder to perform larger explorations. In the systems addressing ST-VQA tasks, all these aspects need to be considered plus the ones concerning the target task. By analyzing the quality and pertinence of the database, we found images in which, even for humans, determining what the textual answer required is almost an impossible task, due to the quality and resolution of the areas where the required text is located. We also found other cases, where the required text is not even present in the image. Passing these samples to our model reduces its performance score. For this case, the solution could be to clean the data manually, as well as to train a different model that detects these special cases, but this would be a whole other task. The best alternative is to strengthen the model to handle this type of noise. We also believe that one of the main causes for our current performance is the number of data available and the implemented techniques for data augmentation that we are applying to the images. Although these seem to work for images in which the text is very accentuated, most natural images appear with variations difficult to

determine, in which the text inside is very noisy.


Figure 5.7: Example of model predictions. In the first two samples, the questions target textual data that is visually well defined. In the third question, the answer prediction mechanism is distracted by the representation of the predicted response. Both answers, GT and predicted, contain the same number of characters in the first word [bedford st, debrand] and share more than half of the characters [b, e, d, d, r]. In the last question, the model also predicts the wrong answer. The main reason is that the text is not visually well defined, even for human performances.

Finalizing this first approximation, we present a qualitative analysis of the performance of the model for a set of particular samples, seen from the semantic standpoint. These samples are selected because of their answer frequency, or their visual content. For example, Table 5.3 presents different images that contain the word "STOP", with the distances between the predicted embedding of a sample in which the image-question pair have the ground-truth answer "STOP", and the vector representations of the words in the answer database. In this case, the retrieved answer will be "stop". This could be a special case, because the frequency of stop in the test set is bigger than 1, and also, because the visual data related with these samples can be very similar

(red signs with the same shape), which helps in finding similar content in the database.

Table 5.3: Distance values between the predicted embedding of sample with GT answer "stop" and all the remaining n-gram vector representations for the words in the answer database.

| **Answer** | **Distance** |
|---|---|
| stop | 0,0 |
| alto | 1,925972 |
| code search | 1,940344 |
| bnp | 1,940409 |
| aon | 1,947848 |
| sptc | 1,949048 |
| ami | 1,951303 |
| ... | ... |

One of the main goals in the ST-VQA task is that the proposed model be capable to discriminate between semantic information and visual information. In Figure 5.8, we wanted to analyze if the model is indeed learning to recognize semantically the content in the image, or if it is learning based only on visual objects or patterns. In the sample, the most similar results are also numbers, in most of the cases related with the answer "215". However, it can be biased for the representation, the length, and even the visual content, which is very similar, as well as other variables that require further analysis. Learning to discriminate semantically is very challenging. In some cases, like the one in Figure 5.8, the textual information is very tenuous, which makes it harder to recognize this text even for humans.

In this section, we presented the first approximation to tackle the problem of ST-VQA. The two main evaluations are: 1) the analysis of using context-free and context-based embeddings to represent the questions, and 2) the representation of the answer space by using an n-grams-based model. The results of our explorations show the challenge ahead when solving this task. We also notice that the low quality of the database has a huge impact on the learning of the model. In the next section, we present an extended and improved version of the proposed framework. We explore feature extraction models along with suitable embeddings not only for the questions but also for the images and complementary data. Concerning the answer space, we also evaluate a state-of-the-art technique to handle the problem of out-of-vocabulary (OOV) words. We make use of another popular database for the ST-VQA task called TextVQA [207].

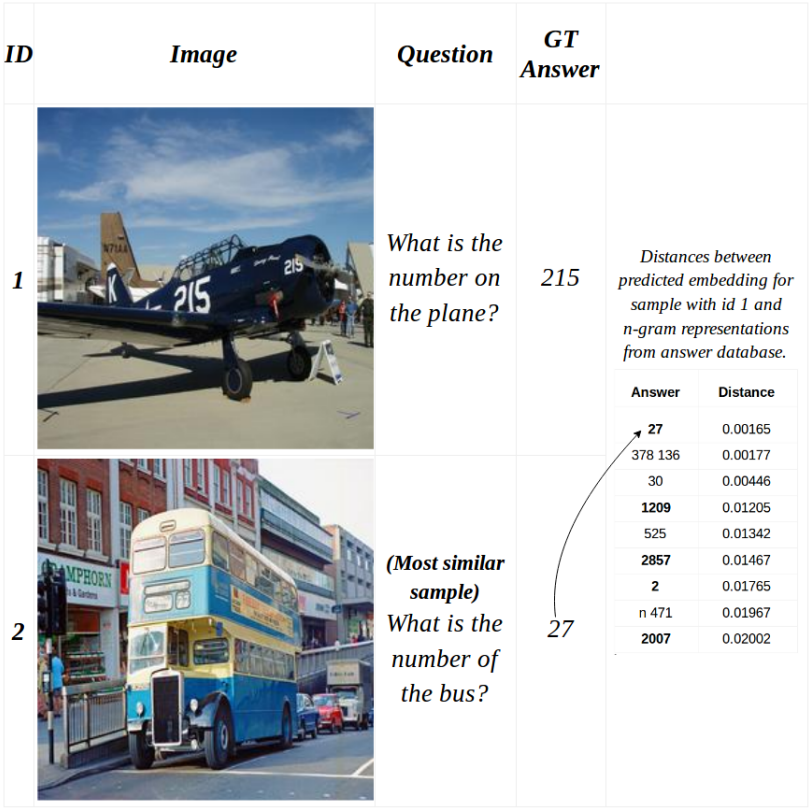



Figure 5.8: ID 1) Represents the input image with its associated question and ground truth answer "215". Once the system predicts the representation of the answer (n-gram) for the pair of data image-question, it is compared against all n-grams representations in the database. The most similar n-gram representation found belongs to the answer "27", where ID 2) is its associated image-question pair.

## 5.3 Model improvements: auxiliary modules + copy module for the answer space

In this section, we analyse an improved version of our proposed framework by including additional data to train the system. This additional data is sent to the model in the form of a complementary network representing embeddings from textual and visual data. We use visual and textual embeddings from OCRs as complementary information that may help to improve the learning of the model. However, it is the model that learns to recognize the target spatial references where some text may be located. We also evaluate a bag-of-words representation with the option of augmenting its dimensionality with a dynamic set of spaces to represent the

answer space. We conclude our analysis by exposing drawbacks related to the copy module which is the state-of-the-art solution. We make use of a second metric to compute the scores for the dynamic spaces so that the copy module can take advantage of texts that are not perfectly recognized by the OCR system. In summary, the input data contains image features, questions embeddings, multiple OCRs embeddings, and the expected output is the answer to the question. In this section, we make use of the second most popular database, the TextVQA database.

### 5.3.1 Modules

In this section, we describe the modules proposed in our second framework (see Figure 5.9). Apart from the improvements and modifications in the basic modules, we also include modules that have been shown to help increase the overall performance of the model.

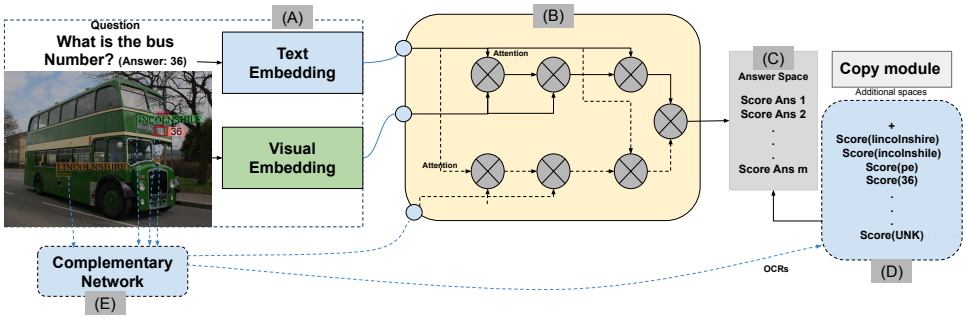



Figure 5.9: Modules included in ST-VQA frameworks. Modules A, B, and C represent the basic modules. Modules D and E are auxiliary modules added as strategies to improve performance. A) Embedding module for input data of different modalities (Images/Questions/OCRs); B) Central Module in charge of fusing input data; C) Answer space in charge of processing the answers and it is represented as a bag-of-words; D) Copy Module in charge of handling additional spaces for the answers (OOV); E) Complementary network in charge of input complementary or additional data.

#### The embeddings module

The embeddings module is in charge of computing input features for the visual and textual modalities, including other possible data such as OCRs, and localized features (see Figure 5.9 - A). For this reason, different specialized models for each modality can be studied. For example, different networks based on CNN are appropriate to compute the embeddings of the images,

while for text, several possible context-free and context-based models can be used as we have shown in Section 5.2.2. Section 5.3.2 provides a detailed description of the models tested during the experimentation phase.

**Central module**

The central module represents the component in which the data is combined (see Figure 5.9 - B). The module receives the features extracted by using module A and uses them to train the network and give the correct answer. We make use of a similar attention mechanism as the one presented in [201] that is used for image captioning and traditional VQA systems. Its attention mechanism is called top-down attention, and it consists in concatenating the previously generated word and the visual feature, with the aim of sending as much context as possible to the textual model that is used to generate the final caption. We modified it so the attention mechanism is directed from the question network to the visual network and the complementary network. Attention mechanisms were created as a mean of enhancing the performance of the artificial neural network by concentrating on the relevant things while ignoring others. In this case, the attention is increased on the question of what information truly needs to be filtered.

**Answer space module**

The answer module is in charge of the representation of the target vector relying on an answer space (see Figure 5.9 - C). We evaluated the usage of a fixed answer space commonly known as a boolean bag of words (BoW), in which the score of each space in the final vector will indicate the presence or absence of the word.

**Copy Module**

The copy module works as a mechanism to handle the OOV problem (see Figure 5.9 - D). This is especially required because the dimension of the answer space can grow indefinitely. For example, in the TextVQA database [207], more than 50% of the answers are unique (more than 30,000 distinct answers) creating a problem of representation. The copy module then works by adding a set of additional spaces to the fixed answer space (module C), filled with scores computed by using the OCRs recognized in the image. Thus, the final dimension of the answer space will be the one fixed by the set of selected answers from the training data + the set of

dynamic words with a fixed number of spaces representing the OCRs.

Because we have the predicted scores for the original set of answers given by the model, now we need then to compute the scores for OCRs recognized in the image. The ones that are located in the dynamic spaces. We propose to use two different metrics. First, the *Human Score* metric proposed in [207] computed as follows:

$$HS(\mathbf{ans}) = min(\frac{\#\, humans\, that\, said\, \mathbf{ans}}{3}, 1) \tag{5.6}$$

It means, each OCR will be taken as the possible *"ans"* to compute the score. This means that for *"ans"* to get a $HS = 1$, *"ans"* should be present in the set of answers given by the annotators at least three times (taking into account that we have access to the set of possible answers from the training set).

Figure 5.10 shows an example of calculation of scores using Equation 5.6 for an image with two different questions associated. For the first question *Qi*, the answer is composed of two words, (*eddie, izzard*), which are outside the fixed answer space. The copy module could help to use the OCRs extracted from the image as an advantage, however, in this case, the *Human Score, Equation 5.6*, will be zero for all the OCRs because it seeks a perfect match between the ground truth answer *"eddie izzard"* and each one of the OCRs in a separate way *["eddie", "izzard"]*, leading to a zero vector as the target representation for this sample. For the second question associated *Qj*, it works as expected, as there exists an exact match between the ground truth answer and the OCRs.

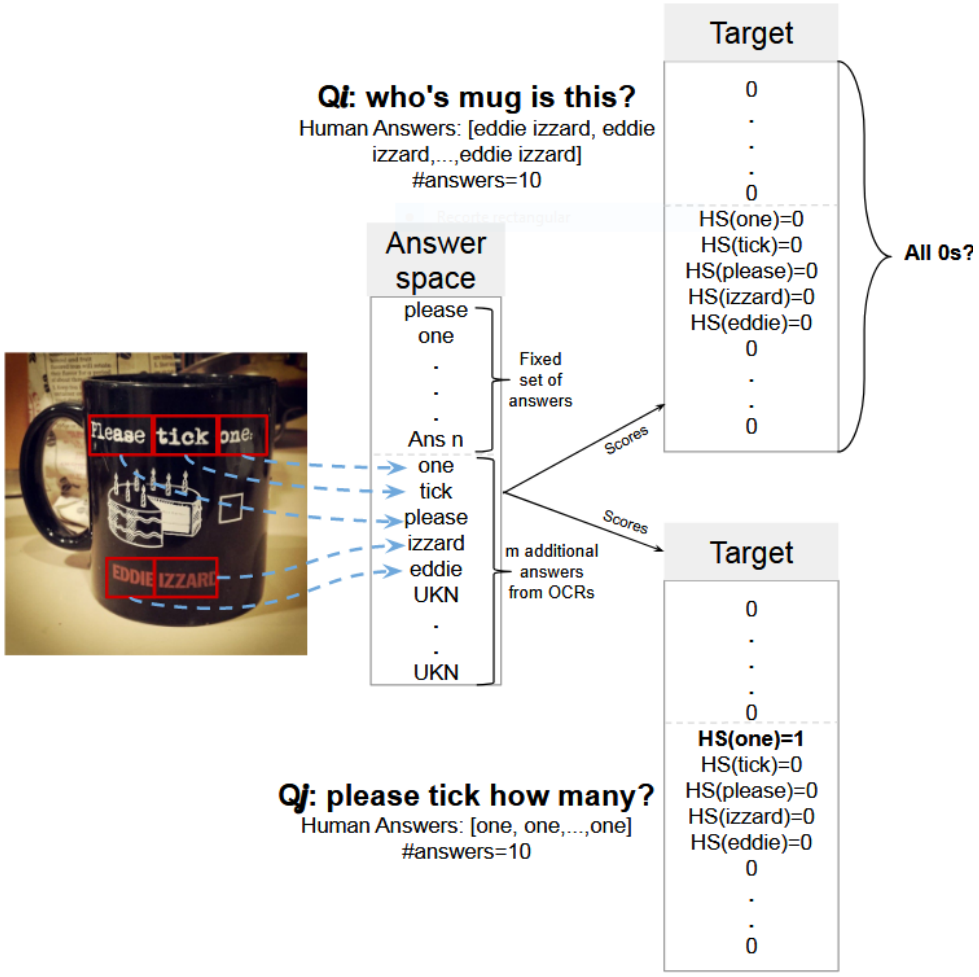



Figure 5.10: Sample of assignation of scores using Human Score in the positive and negative cases of the match between ground-truth answers and the set of OCRs of the image. In the first case, the OCRs that compose the answer contain two words recognized separately, therefore each one will have one score (< 1). In the second case, the answer is only one word and the score will be 1.

Another example of the use of the copy module is when the set of human answers contains more than one answer for the same question, as it is expected to be the same for all the 10 annotators, but in some cases they can differ and give a different answer. As an example, let us assume that the set of human answers is *[stop, emergency stop, emergency stop, stop, stop, emergency stop, emergency stop, it is an emergency stop, emergency stop, unanswerable]*, and the set of OCRs recognized is *[stop, emergency]*. Even if the majority of answers in the ground truth is *"emergency stop"*, it also contains the ground truth answer *"stop"*, which results conveniently in

setting a score greater than 0 for the OCR *"stop"*. As in the two previous cases, there are other cases in which the copy module may or may not work because it requires to have texts that are 100% well recognized in the images (*"one" is completely different from "one:" when computing the Human Score*), or when the answers contain more than one token.

To improve the calculation of scores when partial matches are found in the OCRs of the image, we propose to use a second metric based on the Average Normalized Levenshtein Similarity (ANLS), computed as follows:

$$ANLS\ Score(ocr) = \frac{1}{M}\sum_{i=0}^{M-1}(1 - NL(ans_i, ocr)) \tag{5.7}$$

where M = 10 is the set of answers given by 10 human annotators and NL is the Normalized Levenshtein Similarity. Thus, for the previous example in Figure 5.10, for *Qi*, the new scores for the target OCRs will be *Score("izzard") = 0.4166* and *Score("eddie") = 0.5*

There are also limitations related to the OCR system used, such as recognizing a single word separated in each one of their characters, or not recognizing the target text (i.e., the ground truth word from the human answer set that truly appears in the image). In Section 5.3.2, we discuss the advantages and disadvantages of training models that use the copy module as the main strategy to solve the OOV problem, as well as the dubious results obtained when evaluating the performance.

**Complementary Network**

This module represents the inclusion of networks that input additional data into the framework (see Figure 5.9 - E, Page 100). We tested three different setups: first, Fasttext embeddings [210] from OCRs recognized in the images (we use the OCRs available in the database) as in [207]. However, we do not concatenate the order of the OCRs, and we input the average of the embeddings from all the OCRs available without weighing them. Second, global features extracted from the visual boxes containing the text recognized in the image; For training, we used the ground truth answers to filter the boxes with text matching in at least 30% of the answers, while for validation, we used the entire image. We also tested the scenario when both, Fasttext and global features were sent into the framework together.

### 5.3.2 Experimental evaluation

In this section, we describe the database used in our explorations, followed by the evaluation metrics, the baselines, and ablations proposed together with the results and concluding with an analysis of the proposed approach.

#### TextVQA database

To explore different evaluation scenarios, we are using the TextVQA database [207], as it provides baseline results for the validation set. This database contains 34,602 training samples and 5,000 validation samples, with almost 50% of the answers being unique. This shows the difficulty of using a fixed set of words in the answer space.

#### Evaluation metrics

Two main metrics are used to evaluate the performance in this task. The accuracy performance (see Appendix A.1.4) and in the cases where the copy module is used, the calculation of the accuracy changes by using the Human Score accuracy (Equation 5.6). Then, the predicted answer is obtained by getting the index of the max value in the output vector. If the index is in the first part of the prediction (fixed space), the answer will be one of the fixed spaces shared among all the samples. But, if the index is in the additional/dynamic part of the output vector, the answer will be one of the sets of OCRs recognized in the image at the index position.

The second performance metric is the Average Normalized Levenshtein Similarity (ANLS) computed as follows:

$$
\begin{aligned}
ANLS &= \frac{1}{N}\sum_{i=0}^{N}\left(max_{(j-M)}\, s(ans_{ij}, o_{qi})\right) \\
s(ans_{ij}, o_{qi}) &= \left\{\begin{array}{cc} (1 - NL(ans_{ij}, o_{qi})) & if\ NL(ans_{ij}, o_{qi}) < \tau \\ 0 & if\ NL(ans_{ij}, o_{qi}) \geq \tau \end{array}\right\}
\end{aligned}
\tag{5.8}
$$

where N is the total number of questions, M is the total number of ground-truth answers per question, $ans_{ij}$ is the ground truth answers where i = 0, ..., N, and j = 0, ..., M, $o_{qi}$ is the network's answer for the i-th question $q_i$, and $\tau$ is the threshold. This threshold determines if the

answer has been correctly selected but not properly recognized, or on the contrary, the output is a wrong text selected from the options and given as an answer [211].

For this task, the second metric, ANLS (Equation 5.8), could be more convenient, as the system can find partial matches among the set of words in the answer space. On the contrary, evaluating the accuracy (see Appendix A.1.4) imposes a huge penalty if the model does not find perfect matches for the answers. This is directly related to the answer module and the OOV strategy used. However, considering the value of $\tau$ for the calculation of ANLS score that penalizes predictions matching in less than 50% of the characters, the performance for both metrics is expected to be similar. As this is the parameter established for the results of the baseline, we leave the exploration of different values of $\tau$ as future work.

**Baselines and Ablations**

We perform several ablation studies for the modules described in Section 5.3, where we aim to analyze the performance, drawbacks, and future improvements when targeting this task. We describe the ablations performed, to see if adding/ changing/ replacing key modules of the system would lead to obtain better results. For the first 5 models, we wanted to analyze the embedding module (see module A of Figure 5.9, Page 100), for the image and question data, and select the best one to test the rest of the evaluation scenarios. The components included in this set of studies from Figure 5.9 are modules A, B, and C. For the answer space, we use the set of 3,997 most frequent answers (Small Set SS, where the answers selected are those with frequencies $\geqslant 2$) in the training database. As representative embeddings, we compare two models for the images, ResNet101[87] and Faster R-CNN (bottom-up (BU) attention) [201] with final representations of *2,048-dim* and *36 (features per image with a 2,048-dim each one)* respectively. And two embedding models for the text, GloVe [76] and BERT [79], with final representations of *300-dim* and *768-dim* respectively, for both models, we use a set of 15 tokens as the maximum length. The vocabulary size extracted from the questions is 9,312 unique words. Therefore, the scenarios evaluated are: GloVe + ResNet, GloVe + BU, BERT + ResNet, BERT + BU.

After selecting the best set of embeddings based on the previous results, i.e., GloVe embeddings for the question, bottom-up features (BU) for the images, and having fixed a small answer space (SS), we wanted to evaluate if the performance improves by increasing the size of the answer

space to a large space (LS). The last row in Table 5.4 presents the result by using a larger set for the answer space of the 7999 most frequent answers in the training database. Table 5.4 presents the results obtained for these first 5 models for validation samples in which the answer is contained in the selected fixed set of answers, i.e., for the answer space SS, the number of samples gets reduced to 18,516 for training and 2,214 samples in validation. For the answer space LS, the number of samples gets reduced to 21,183 for training and 2,290 samples in validation. Thus, to make a fair comparison, results are reported over these validation subsets. We compare our results against the LoRRA model [207] as it is the most similar to ours. The major difference in the results may be explained by the complex and expensive training process carried out for LoRRRA ($\approx$ 24000 iterations, making use of $\approx$ 8 GPUs). Due to resource limitations, we only trained our model with $\approx$ 200 epochs and a maximum of 2 GPUs.

Table 5.4: Performance for representative embedding models for visual and textual data in ST-VQA systems, with a fixed set of words in the answer space. Validation results are reported over the set of samples which answers are contained in a small fixed set SS or a larger set LS.

| Model | Acc | ANLS | AVG |
|---|---|---|---|
| **LoRRA** + SS[207] | 0,2656 | - | - |
| **GloVe + ResNet101 + SS** | 0,1853 | 0,2274 | 0,2065 |
| **GloVe + BU + SS** | **0,2005** | **0,2474** | **0,2240** |
| **BERT + ResNet101 + SS** | 0,1910 | 0,2319 | 0,2115 |
| **BERT + BU + SS** | 0,1978 | 0,2366 | 0,2172 |
| **GloVe + BU + LS** | 0,1860 | 0,2279 | 0,2069 |

The second set of evaluation scenarios aims to analyze the inclusion of the copy module, based on results from Table 5.4. The components included from Figure 5.9, Page 100 are modules A, B, C, and D. We wanted to evaluate the appropriate number of additional spaces, for this, we experimented with three different numbers. First, 50 spaces following the work [207], second, the average of the number of OCRs of all training samples ($\approx$ 9.8) * 2, i.e., 20 spaces, and finally, the average of the number of OCRs from all training samples, i.e., 10 spaces. In this case, the data sets contain all the samples (34,602 for training and 5,000 for validation).

As we discussed in Section 5.3.1, the assignation of scores using Equation 5.6, does not take advantage of text that is not perfectly recognized, leaving many samples with zero score vectors.

In this case, we wanted to change the assignation of scores by using the average ANLS score (see Equation 5.7) over the set of human answers. The last row in Table 5.5 changes the assignation of scores using ANLS score metric. Table 5.5 presents the results for this set of evaluation scenarios.

Table 5.5: ST-VQA performance with the inclusion of the copy module with the assignation of scores using Human Score metric and by exploring the number of additional spaces for the OCRs to 50, 20 and 10. For the last result presented, the scoring method was changed to the ANLS score metric.

| Model | Acc | ANLS | AVG |
|---|---|---|---|
| **50 spaces + Human Score** | **0,1854** | **0,1835** | **0,1844** |
| **20 spaces + Human Score** | 0,1778 | 0,1799 | 0,1788 |
| **10 spaces + Human Score** | 0,1792 | 0,1817 | 0,1804 |
| **50 spaces + ANLS Score** | 0,1705 | 0,1816 | 0,1761 |

To evaluate whether the inclusion of more information into the central module would help the performance, we test the alternative inclusion of three complementary data: 1) the average of Fasttext embeddings [210] from OCRs recognized in the images, similar as in [207], but without the addition of order and weighted information, 2) a concatenation of global descriptors extracted from boxes containing target text, and 3) both of them. For this evaluation scenario, the components included are modules A, B, C, D and E from Figure 5.9, Page 100. We use the best model from Table 5.5, adding top-down attention in the central module from the question towards the complementary network data. Table 5.6 presents the results obtained for this set of experiments.

Table 5.6: ST-VQA performance when complementary data is sent into the VQA module. Three types of complementary data were evaluated, Fasttext embeddings from OCRs recognized in the image, Global features extracted from the box containing the target text data and, finally, their combination.

| Model | Acc | ANLS | AVG |
|---|---|---|---|
| **OCR Fasttext** | **0,1848** | **0,1942** | **0,1895** |
| **Global Visual features (GVF)** | 0,1756 | 0,1797 | 0,1776 |
| **OCR Fasttext + GVF** | 0,1843 | 0,1932 | 0,1887 |

**Analysis**

The best results from Table 5.4, Page 107 are obtained by using GloVe vectors + Fast R-CNN (or bottom-up BU) features. The slightly better performance of GloVe over BERT can be attributed to the fact that the structure and meaning of the words in the questions for this database are shared, and therefore the context does not play an important role in the discrimination of different samples. Also, as the last result in the Table showed, increasing the set of possible answers does not necessarily imply an improvement of the performance (see also results of small set SA vs large set LA at "Table 2: Evaluation on TextVQA" [207] that confirm our result). This is because the set of possible answers can contain any combination of characters in different languages that are found in natural images, while in the case of the TextVQA database, there are more than 19,000 different answers among 34,000 samples. This makes the establishment of a manageable fixed set of words as the answer space unfeasible and raises the need to handle the OOV problem.

For Tables 5.5 and 5.6, the copy module was included as a strategy to handle the OOV problem. The best number of additional spaces to include in the answer space for this database was 50. This means that the performance improves as more text data recognized in the image is provided to the system. On the contrary, the last result in the Table that evaluates the ANLS Score did not show an improvement in the performance. This is related to the fact that for most of the samples in the database with at least one OCR recognized, their associated answers are composed of only one token. Therefore, the scores will be similar (for only 8.9% of the samples in TextVQA, answers contain more than one token).

Regarding the inclusion of additional data, Fasttext embeddings showed a small performance improvement. On the contrary, the inclusion of the global visual descriptors with the target textual data did not show any relevance. This could be due to the attention mechanism used, as it was the same one for both embeddings.

Finally, Figure 5.11 presents the learning curves for loss accuracy in training for the TextVQA database. Both training curves converge smoothly and do not require a large number of epochs. Although the accuracy is $\approx 70\%$, when compared to the results of the ST-VQA database which only reached 30% (see Figure 5.6, page 95) this database contains more quality annotations.

This result shows how the proposed modules have a direct positive impact on model learning.

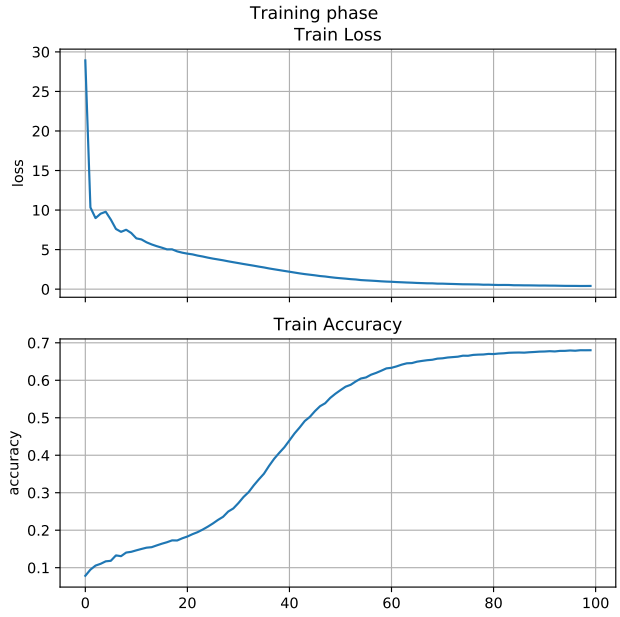



Figure 5.11: Learning curves for the loss and accuracy in training for the TextVQA database.

***Is the copy mechanism solving the OOV problem in a suitable way?*** We wanted to give final comments regarding the convenience of using the copy module as a strategy for the OOV problem. Although the copy module partially solves the OOV problem, each item in the dynamic space could represent as many different words as exist in the OCR space of all samples. In the end, the prediction of the correct answer over these values becomes almost a random choice that depends on the position of the OCR. Better solutions to handle OOV are required because many tasks in the state of the art are facing the same problem. The n-gram representation for the answer space could be a solution as with it, a larger set of answers can be represented by a fixed and manageable set of n-grams (as we showed in Section 5.2).

## 5.4 Discussion

We presented an incremental and extended study for the task of ST-VQA by performing an analysis of the modules required in any framework addressing this task. Our contributions are as follows. Our proposed strategies are very flexible in each one of its components and were tested

for the two principal databases ST-VQA and TextVQA. We evaluated the impact of several basic and helper modules under different strategies. We explored and evaluated the quality of different embeddings or representations for the data involved in the task, including the target images, the questions, and the answers. We evaluated the relevance of the dimension when a fixed set of words (BoW solution) is used as the answer space (which for this problem turned out to have little impact). We also evaluated the performance of the model when using the copy module under two different metrics for the calculation of the scores, both ended up with similar performance, as the majority of data contains answers with only one token. Our final evaluation concerned the performance when including complementary data to train the system in the form of an additional network, resulting in a slight performance improvement. Finally, we exposed some of the main drawbacks of current solutions, especially when handling the OOV problem showing us the need for better and more robust strategies. Being part of the first works to address this task, we presented several experiments and results for the two principal databases. We believe these support future works to advance research in this challenging task.

During the development of these approaches, we found images in which, even for humans, determining the required textual answer is almost impossible, due to the quality and resolution of the areas where the text is located. We also found other cases, where the text is not even present in the image, reducing the performance of our approaches. This demonstrates how data is one of the most important factors when aiming to reach good performances. Therefore, we determine that one important step is to understand how to properly clean and analyze data and to create strategies able to leverage it. Our final approaches aim to address problems around the data, such as noise, imbalance, and insufficiently labeled data. To evaluate our strategies, we consider the problem of attribute learning, a clearly defined task fostering a lot of recent interest in the area of explainable artificial intelligence. Attribute learning can complement class-level recognition and therefore improve the degree to which machines perceive visual objects. In the next chapters, we present the strategies and experiments performed around this task.

# CHAPTER 6

# Multi-Attribute Learning With Highly Imbalanced Data

## 6.1 Motivation

In this chapter, our goal is to analyze and explore alternative solutions for the problem of learning with imbalanced data under the specific scenario of multi-attribute learning. We note that one of the most important factors determining whether a model can perform well is the data itself. In previous chapters (4 and 5), we encountered two main problems associated with the supervision data that was available. The first one is determined by subjective and in some cases erroneous annotations, while the second is the absence of annotations. These two problems lead to an imbalance scenario. We find this problem especially in the application of ST-VQA, where the answer set presents an extreme imbalance (an answer can only be associated to one sample). This makes harder the learning of features from samples belonging to less represented or rare classes. The simplest case is a binary classification task, where the majority of samples belong to one of the two classes. Therefore, this leads to the "accuracy paradox" problem, because in the most frequent case, when the accuracy level is high, it is only reflecting the underlying class

distribution [295, 296]. In this case, the model is only learning to predict the most represented class in the database. Hence, imbalanced learning is a real problem that occurs in several applications in the context of machine learning. For the previously addressed applications, many components can affect the final performance differently, and therefore we have several points of failure. These failures can come from the feature extraction component, the textual data used as supervision, the fusion component in charge of merging the different data modalities, the handling of the answer space, among others. Then the complexity of determining the component that affects performance the most becomes more difficult. Especially when we are evaluating the impact on the model performance for our proposed solutions to the problem of imbalance. For this reason, we have chosen an application for which it is feasible to determine the causes and if our proposals have a direct effect. This is the task of attribute learning. Also, we have chosen this application because the imbalance problem is very common due to the structure of the databases that require collecting a huge amount of information. Attributes are a richer, intermediate data representation. They may represent an intermediate representation, which enables parameter sharing between classes [297]. Moreover, they generally correspond to the semantic representation of experts. The best scenario then would be to have all attributes well represented to get a well-trained system. Unfortunately, this is not the case for most of the real databases where the training database is highly imbalanced with attributes almost unrepresented (see Figure 6.1 for an example image and its associated attributes). In these cases, state-of-the-art works may assume ideal conditions. Some of them expect the combination of attributes to be the same for all samples belonging to a class [240]. Real databases may present variations at the sample-level. For example, not all images display the same set of attributes even when belonging to the same class. It causes also imbalance problems, where classical imbalance learning strategies are not adequate or adapted. Hence, attributes are important in many applications of the state of the art, and even when the main objective is different from the one pursued in this thesis, we believe this is a relevant task.


Figure 6.1: Example image and its associated attributes. Some of these attributes such as "Breast white" may be assigned to many classes of birds, while others such as "Belly orange" are less common. Samples extracted from the Caltech-UCSD Birds-200-2011 (CUB).

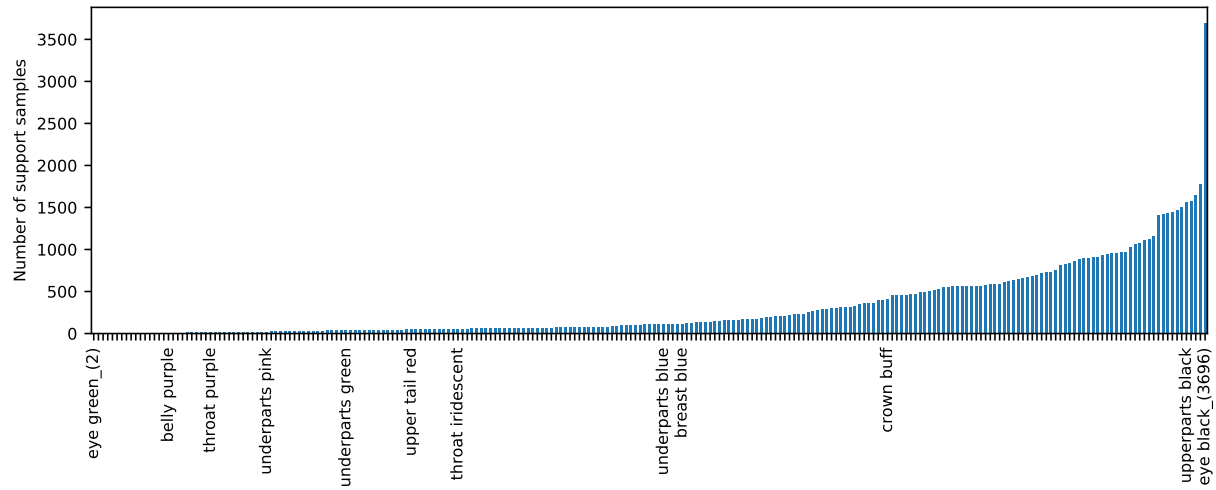
This contrast in the objective also imposes an obstacle, because the comparison against the state-of-the-art methods is not evident. In the state of the art, the main application that incorporates attribute data is classification at the category level. For example, the application of pedestrian identification where the goal is to identify a person based on a set of attribute data [243]. One of the closest works to the one proposed here is the work of Demirel at al. [240]. They propose to use attributes for zero-shot learning (ZSL) of classes. Their goal is to predict the unseen class name based on combinations of attribute names. They first learn an intermediate representation of the image based on its attributes (our goal), but they only use it as an input for a model that learns a word embedding. The strategy then is to find a common word embedding for representations of attribute-based representations and class names. Notice that works based on ZSL target have a different goal, i.e., the classification of unseen classes. Other works use expensive annotations such as part localization for the learning of these intermediate representations [298]. In short, attributes are considered a form of auxiliary data [299]. In our case, the focus is multi-attribute learning. So the results are at the attribute level. For example, if we want to learn color variations for multiple parts of a bird, such as the wings, the head, or the tail, etc. We also do not use annotations such as part localization that are very expensive to get. As we mentioned before, our target problem of attribute learning under an imbalance scenario is very specific and it is not well studied in the state of the art. For this, we found difficult the direct and fair comparison with other methods in the state of the art. Thus, to provide an easy comparison for future works, we selected easily reproducible standard machine learning methods such as

SVM, Logistic Regression, among others.

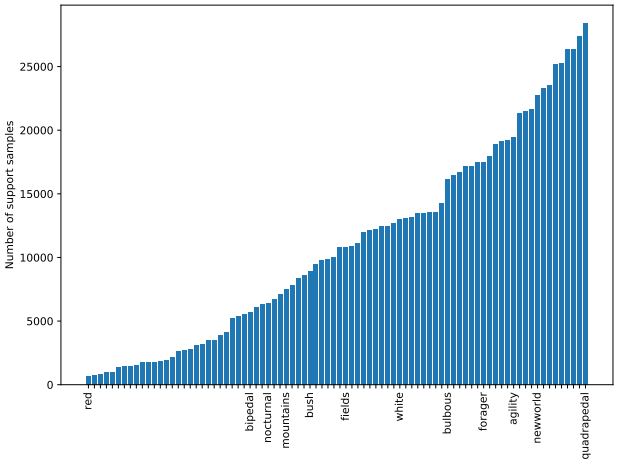
Our strategy is to do incremental analysis along with proper evaluation settings. We begin by evaluating the two principal techniques for imbalance in the independent setting. These are sampling and cost-sensitive learning [262]. In this scenario, each attribute is learned using a separate model, and these learning techniques are easy to implement. To carry out extensive exploration, we select the most general attribute. Once we find the most suitable scenario, we apply it to the remaining attributes. This provides us with a baseline result computed as the average performance for all attributes. Then we can proceed to analyze the problem when all the attributes are learned using a unique model. For this scenario, we propose two different multi-attribute frameworks. As we said earlier, our objective is focused on this scenario. Mainly, because we found three problems when using classic imbalanced learning strategies applied to multi-attribute models. These are derived from a characteristic of imbalance that manifests itself in different ways. To address them, we propose three strategies that allow us to train our models by leveraging high-level data such as class labels. We finish with the results of the average performance of all the attributes against the state of the art.

For our experimental evaluation, we selected three popular attribute databases. The first, CUB [232], is a database of different kinds of birds. Although this database is used more in the state of the art of fine-grained image classification [1], we chose it as our case study due to its high level of imbalance. The other two databases are AwA2 [234] and CelebA[235]. Unlike the CUB database, these two databases are not imbalanced at the attribute level. AwA2 also contains images of a diverse set of animals, while CelebA contains images of persons' faces. Then these last two databases allow us to compare the performance of imbalanced and balanced databases so we can carry out our experimental evaluation. Figure 6.2 shows the distribution of samples per attribute for the three selected databases. Notice that for some attributes in the CUB database, there are very few support samples.

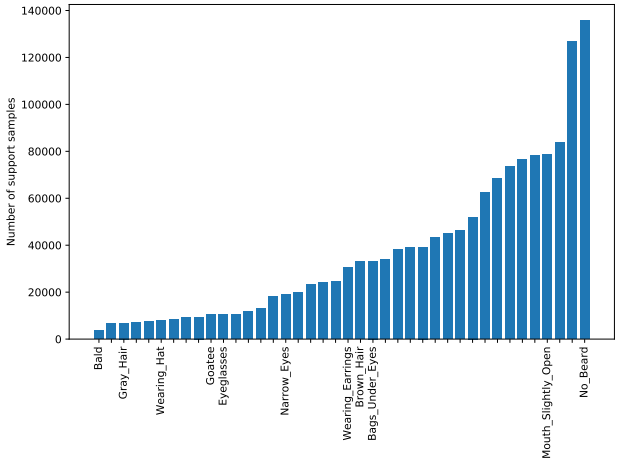
[1]The task of fine-grained image classification focuses on differentiating between classes that are very difficult to discriminate, such as species of birds, flowers or animals, etc.




a) CUB




b) AwA2




c) CelebA

Figure 6.2: Distribution of attributes for each database.

To develop our strategies, we propose to use deep learning. Deep learning has become the main tool to address many artificial intelligence problems because of its success in solving complex tasks (see Chapter 3 for deep learning related work), specially in computer vision problems. These systems have demonstrated their ability to extract and learn high-quality features from visual data [300]. In addition to this, using those systems now is an easy task for many people, like scholars or expert users. They generally use these models merely as black boxes, without

understanding how they work and why results are produced. Then, the focus tends to be on the data. Generally, having "good data" leads to a cleaner training process, and therefore, better performances [14].

In summary, in this Chapter, we evaluate the specific task of fine-grained attribute classification in an imbalanced scenario. Throughout the experiments, we show that one of the main causes of bad performance comes from these problems, more than from the actual ability of the model itself. We study the effect of applying classical imbalance learning strategies to straightforward and successful deep learning models seen as black boxes. Specifically, we will evaluate the use of sampling and cost-sensitive learning strategies. Sampling refers to the techniques used to order the training data to better balance the class distribution. On the other hand, cost-sensitive learning involves explicitly defining and using suitable costs when training. This has a direct impact on those imbalanced classes [301]. We propose adaptations to these two classical imbalanced learning strategies that cannot be directly applied when using multi-attribute frameworks. Our strategies are designed to be used for multi-attribute deep learning models, i.e., multi-task or multi-label architectures with competitive performance in real databases (see Appendix B.1.1 for a brief definition of these types of learning paradigms). We expose a certain number of problems derived from highly imbalanced databases, usually ignored in the state of the art. The problems we found are described next. First, *the number of attributes per image is different*. Some images can have many attributes available in the training set while others only have 1 or 2. This makes the target representation very sparse and harder to learn. This is a problem of imbalance at the level of the number of attributes per sample. Second, *the problem of unrepresented classes or "attribute-value" combinations*. For example, learning color variations in birds for two specific attributes *"Primary"* and *"Eye"* from 15 different variations can be difficult for some of them. To find birds within the whole range of representative - *"Primary"* - colors is very common, while for the second attribute, *"Eye"*, the most common value is *"Black"*. Therefore, an attribute combination such as *"Eye-Pink"* is very rare and difficult to learn. Finally, the third problem is related to the *inconsistency of labels/classes pairs given by annotators which are very subjective and can differ and be incorrect*. This leads to a lack of uniqueness and discrimination between classes. Formally, this corresponds to a large intra-class variance and a low inter-class variance. These problems require deciding which one of these values is the most reliable to train

the model, and how to feed the model with all the possible variations.

Our two main contributions in this chapter are as follows: 1) We validate the effectiveness of classical imbalance learning approaches applied to straightforward multi-attribute approaches, 2) we expose different problems derived from imbalanced databases in the context of very fine-grained multi-attribute learning, some of which are ignored in the state of the art. Subsequently, we present our proposals to tackle each one of these problems.

## 6.2 Databases

The imbalance learning problem is extensively studied by using a representative attribute database as our main use case. We also report results in the multi-label scenario for another two attribute databases that have a better data distribution (see Table 6.1, Page 120). We describe these databases next.

***Caltech-UCSD Birds-200-2011 (CUB).*** This database contains images of 200 bird species. All images are annotated with bounding boxes, part locations, and attribute labels. We only use the main classes as auxiliary data, along with the attribute labels for the target task (see Figure 6.1 for an example image and its associated attributes). Only attributes based on color information are selected for the experimentation (16 out of 28 attributes are related to the color information) [232].

***AwA2.*** This database contains 50 animal classes with 85 different attributes. We use the same partitions as proposed in [234]. The partitions were proposed by distributing the classes, 40 to train and 10 to test (standard partitions). The main difference with this database is that samples belonging to the same class share the same set of attributes. This aspect balances the database at the level of attributes, allowing us to compare it with a well-distributed database (see Figure 6.3 for an example image and its associated attributes).

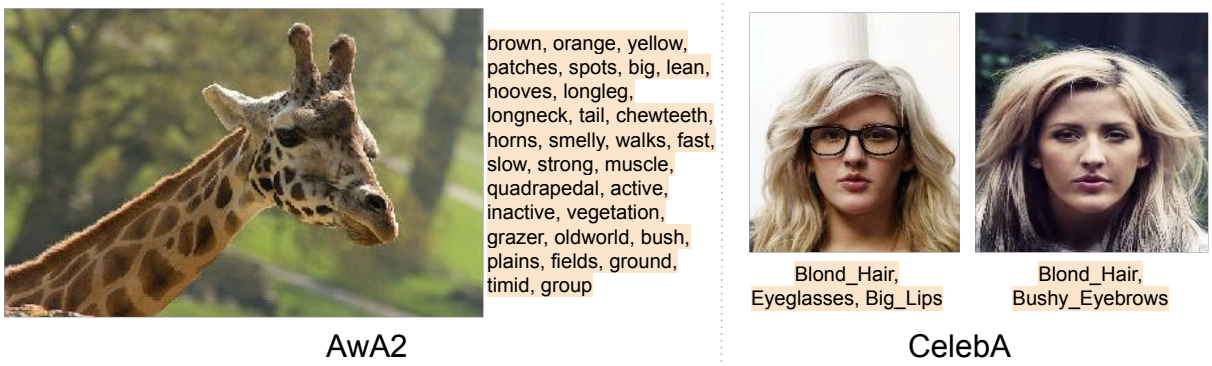



Figure 6.3: Example images from AwA2 and CelebA database and their associated attributes.

***celebA.*** This database [235] contains data for different identities, with landmark locations (which we do not use) and binary attributes. It is widely used in the context of Face recognition. This database presents an imbalance at the attribute level with a minimum and a maximum number of samples per attribute of 4,547 and 169,158. Samples from the same class do not share the same set of attributes because of the nature of the relation between classes (identities) and attributes ("sunglasses", "mustache", etc). For example, if we have two images of the same identity in which the attribute "sunglasses" is present in only one of them (see Figure 6.3 for example images and their associated attributes).

Table 6.1: Databases distribution.

| **Database** | **# classes** | **# attributes** | **# Train** | **# Test** |
|---|---|---|---|---|
| CUB | 200 | 312 (239 used) | 5,994 | 5,794 |
| AwA2 | 50 | 85 | 29,409 | 7,913 |
| celebA | 10,177 | 40 | 162,770 | 19,962 |

As stated before, our main goal is to study the abilities of straightforward but successful deep learning models when used as black boxes by expert end-users. Our main objective is to determine if these are capable of learning attribute data at the finest level, despite being used in an imbalanced context, and to retrieve content sought by scholars or experts. We evaluate the task of fine-grained attribute classification in birds, for color variations for a set of different attributes. We make use of the previously described CUB database because of its characteristic of containing different levels of imbalance. We compare the performance vs other well-known machine

learning methods for other widely known databases in different domains, animals (AwA2), and face recognition (celebA). These last two databases are well structured and distributed; therefore, we can make the comparison under better circumstances. In the next sections, we present the methodology followed to pursue our objectives along with the subsequent experimental evaluation.

## 6.3 Proposed attribute independent framework

In this section, we perform an incremental and extensive evaluation of attribute modeling in the independent scenario, i.e., when each attribute is trained using separated models. For example, classify the attribute *"Primary""* in its 15 different color variations such as [white, blue, black,...color15]; then, train another classifier for the second attribute *"Wing"* for its 15 color variations as well, etc. In this first evaluation, we selected the most representative attribute to be able to explore many learning scenarios. Once we find the best setting, we present the results for all the remaining attributes.

### 6.3.1 Proposed framework

The model used to train each independent attribute is a ResNet50 [87] in which the weights are initialized by using a pre-trained network on ImageNet (see Figure 6.4). ImageNet is also composed of natural images with several classes of animals. This is the reason why this network trained with this database represents a good feature extraction model for our data. The dimension of the classifier layer is adjusted according to the number of classes for each attribute. Therefore, in this scenario, we train as many networks as attributes are present in the database.

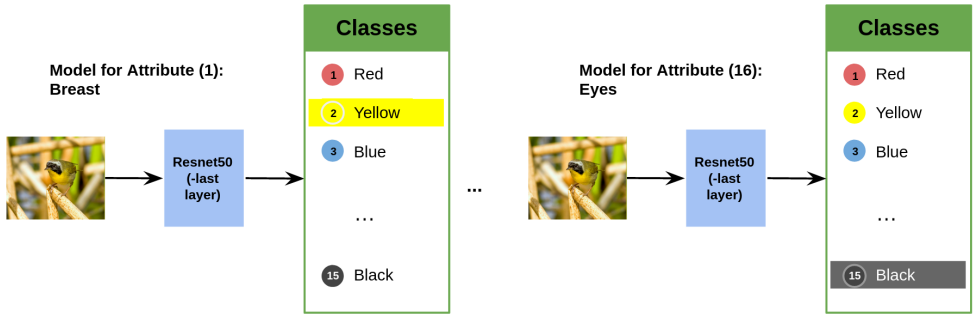



Figure 6.4: Multi-label model for the independent case. We train one model for each one of the 16 attributes. Each attribute contains different colors (up to 15 different colors).

Next, we describe the classical imbalanced learning strategies implemented. Two main strategies are well-known when training a model with an imbalanced database [261]:

- **Sampling.** It consists of sample elements from the training data by establishing a set of probabilities or weights. This affects directly the method to create the mini-batches fed to the model. Two different forms to compute the weights are tested. First, by taking into account the number of samples per class (different weights), where the weight for a sample that belongs to the class $i$ is computed as $1/Ni$, where $Ni = number\ of\ samples\ in\ class\ i$. The second approach is to give equal weight to each sample ($1/N\ classes$). Another option when using sampling, also called Weighted random sampler (WRS), is to replace samples. This option allows us to continue passing (repeating) samples of rare classes in the creation of mini-batches once all the samples belonging to them are already fed to the model.

- **Cost-sensitive learning.** It concerns the update of the loss function by using weights given to each class (Weighted Loss). The weights are computed as $1/Ni$ , where $Ni = number\ of\ samples\ in\ class\ i$. Then, the definition of the loss functions for the regular Cross-Entropy loss and the weighted version is as follows:

$$\begin{aligned} CE(p_x) &= -y_x * log(p_x) \\ Weighted\ CE(p_x) &= -W_{class} * y_x * log(p_x)) \end{aligned} \tag{6.1}$$

where $p_x$ is the prediction of the model for the input $x$, $y_x$ is the ground truth label of the

sample, and $W_x$ is the weight assigned to the associated class.

### 6.3.2 Experimental evaluation

In this section, we describe the set of experiments to carry on in the independent case. To be able to test different evaluation scenarios in the training phase, the most general attribute is used to select the configuration with the best performance. The *"Primary"* attribute is the most representative attribute (from all the 16 part attributes with color information) as it does not refer to any specific part of the bird. It refers to the most representative color of the bird (see Figure 6.5 for example images and the value of the *"Primary"* attribute).

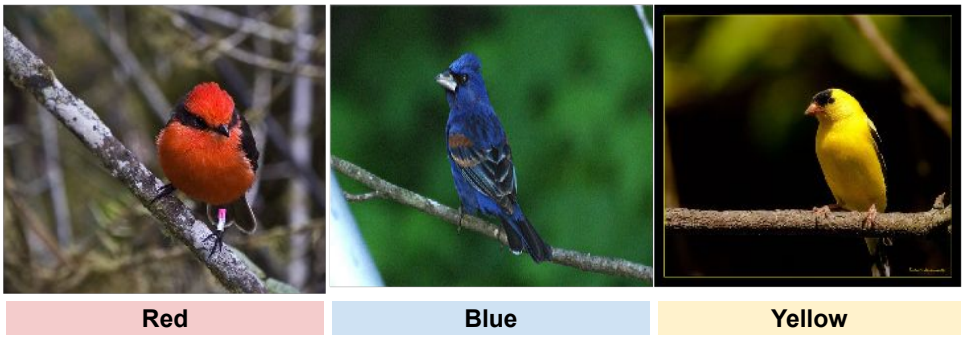



Figure 6.5: Example images from the CUB database exhibiting the attribute "primary" with different values.

Table 6.2 presents results for an incremental evaluation making use of the two strategies described in the previous section: sampling and cost-sensitive learning. All settings are computed by using a learning rate (LR) of 0,0001, a batch size (BS) of 24, and 120 epochs. In the first row, none of the strategies are applied. In row 2, standard data augmentation is applied to obtain better results in the test set such as rotations, horizontal flips, and perspective changes. Rows 3 and 4 evaluate the use of a sampling strategy (WRS, without replacement) using different and equal weight values, with the last one performing better. Row 5 evaluates the use of the *Replacement* option with the best setup (WRS + equal weights) showing a small improvement. The last two rows evaluate the use of using a cost-sensitive loss strategy with data augmentation (row 6) and when combined with the WRS strategy (row 7) getting decay in the performance. The rest of the experiments in the independent case are reported using the best setting found, that is, by

applying data augmentation and using a sampling strategy with equal probability weights for all samples and with replacement.

Table 6.2: Incremental analysis for the performance of the "Primary" attribute when using Sampling and cost-sensitive learning strategies.

| Id | Setting | Accuracy (%) |
|---|---|---|
| 1 | None strategy | 72,78 |
| 2 | Data augmentation | **73,06** |
| 3 | Weighted random sampler with different weights + no Replacement | 72,57 |
| 4 | Weighted random sampler with equal weights + no replacement | **73,21** |
| 5 | Weighted random sampler with equal weights + replacement | **73,24** |
| 6 | Data augmentation + weighted loss | 66,98 |
| 7 | Weighted random sampler with equal weights + replacement + weighted loss | 65,70 |

Based on the results for the "Primary" attribute, we use the setting found for the training of the remaining attributes. Therefore, we have the performance for each attribute in the independent case to compare against the multi-attribute case. Table 6.3 presents the results for all attributes trained independently. The average accuracy is $\approx$ 68,41%. This value is the baseline and expected value to reach with our multi-attribute models.

Table 6.3: Results for all attributes trained independently.

| Attribute | Accuracy (%) |
|---|---|
| back | 68,30 |
| belly | 74,49 |
| bill | 57,35 |
| breast | 72,50 |
| crown | 67,16 |
| eye | 90,69 |
| forehead | 66,41 |
| leg | 50,12 |
| nape | 66,76 |
| primary | 73,16 |
| throat | 70,50 |
| under tail | 61,46 |
| underparts | 74,51 |
| upper tail | 61,09 |
| upper parts | 69,67 |
| wing | 70,35 |
| **Average** | **68,41** |

Figures 6.6, 6.7 and 6.8 present the training loss and accuracy in test for the attribute 'Primary' (our case study), followed for the attribute with the best performance ('Eye') and with the lowest performances ('Leg'). The performance of the attribute 'Eye' presents a high imbalance level that explains the non-smoothly and also high convergence as most of the samples belong to the class 'Eye - Black' ($> 3000$) while for the class 'Eye - Green' there only 2 samples. On the contrary, the training curves for the attribute 'Primary' have a soft convergence because all the classes are well represented ($averagenumberofsamplesperclassis \approx 400$) and second, the attribute is indeed easier to learn. The learning of the 'Leg' attribute may represent a challenge as well, because of its localization. However, there there are fewer unrepresented classes and therefore, the training loss and accuracy in the test has as well a smooth convergence.


Figure 6.6: Training loss and accuracy in the test for the attribute: 'Primary'. Having a well class distribution and being 'easy' to learn, its associated learning curves converge smoothly.

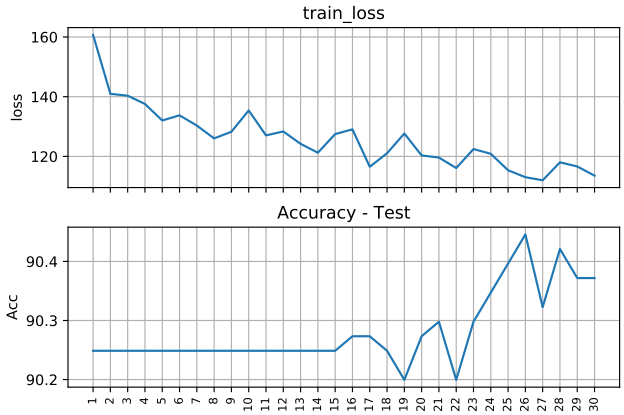



Figure 6.7: Training loss and accuracy in the test set for the attribute with the highest performance: Eye. Being the attribute with the highest level of imbalance, its corresponding learning curves have a hash convergence.

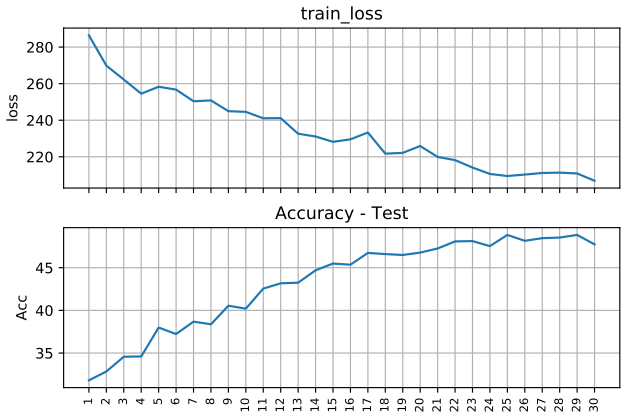



Figure 6.8: Training loss and accuracy in the test set for the attribute with the lowest performance: Leg. Although its accuracy in test, the learning of the attribute represents a challenge because of its visual localization. However, the learning curves are smooth due to its well class distribution.

In the next section, we first present the problems found in the multi-attribute scenario along with strategies for their optimal use, which represents one of our main contributions. We follow these experiments with the description of our proposed multi-attribute frameworks (multi-task and multi-label models) and their respective performances, concluding with a comparison with the the selected machine learning methods.

## 6.4 Problems encountered in the multi-attribute scenario and our adaptations

In this section, we present problems associated with different levels of imbalance that severely affect the final performance of the model, and that arise in the multi-attribute scenario making its use non-optimal. Next, we describe these key problems together with our proposals.

1. **The sampling strategy is not easily applicable.** The sampling strategy provides a "probability value" for a sample to be selected. However, for a sampling method to determine when an image belongs to a rare class, in a multi-attribute model, is more difficult. First, consider the most difficult case, the multi-task model, with only two tasks, "Primary" (main color of the bird) and Eye" (color of the eye), each one of them with 15 different

classes (color variations). In a multi-task model (see models in Section 6.5), the image is passed through a set of shared layers and forwarded to all the branches each representing one of the tasks. For the task "Primary", classes such as "White", "Blue" and "Red" are very frequent. However, for the task "Eye", these classes are considered very rare. Thus, the complexity of defining which classes are rarer increases with the number of tasks (there are 16 different attributes/tasks). A similar problem is present in the multi-label model. In our experiments, we found that the best strategy is to assign the same probability to all samples regardless of their classes and to use replacement[2]. This helps rare classes to be passed after all samples have been already selected. Figure 6.9 shows the process of the creation of mini-batches during a training epoch by using this strategy.

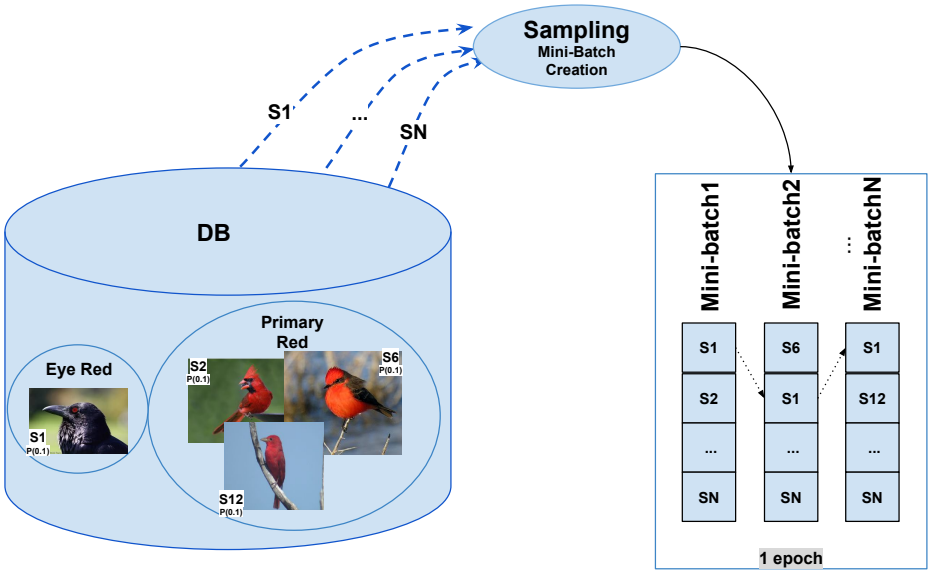



Figure 6.9: Creation of the mini-batches during an epoch. The sampling strategy is implemented to tackle the problem of having very rare classes. All the samples in the database have the same probability of being selected. By using replacement, samples of very rare classes (eye red) are included several times.

2. **Very few samples have annotations for all attributes.** The ideal and most common case is when images belonging to a class share the same set of attributes and values in the domain of the database. In our use case database, CUB, this feature is absent. Although

[2]Replacement: inclusion of the sample in several mini-batches (repetition of the sample)

images do have class information, two images that belong to the same type of bird may have a different set of attribute-color information. For example, we found two images *i* and *j* belonging to the class *"Pigeon Guillemot"*; image *i* contains the set of attribute-values ["Primary White", "Wing Gray", "Crown Gray"], while image *j* contains the set of attribute-values ["Primary Gray", "Wing Gray"]. Notice that some values are different for the same attribute and that there are even some absent attributes in images belonging to the same class. This represents a problem in the multi-task model when updating the loss for those absent classes for some input images. For this problem, two strategies are evaluated: **A)** updating the loss function using a mask value that indicates which values will not be used in the computation, and **B)**, computing the most common value for all samples for each one of the attributes according to the class (there exist 200 different classes). If a sample does not have an annotation for an attribute, the most common value is transferred to be used as ground truth during the training phase (partial transfer). In the previous sample, if the most frequent value for the attribute *"Crown"* in the class *"Pigeon Guillemot"* is *"Gray"*, this is used to complete the set of attribute-values for the image *j* as ["Primary Gray"", "Wing Gray", "Crown Gray"]. Also, we can only use frequent values for all samples belonging to the same class (total transfer). Figure 6.10 depicts the two proposed strategies.

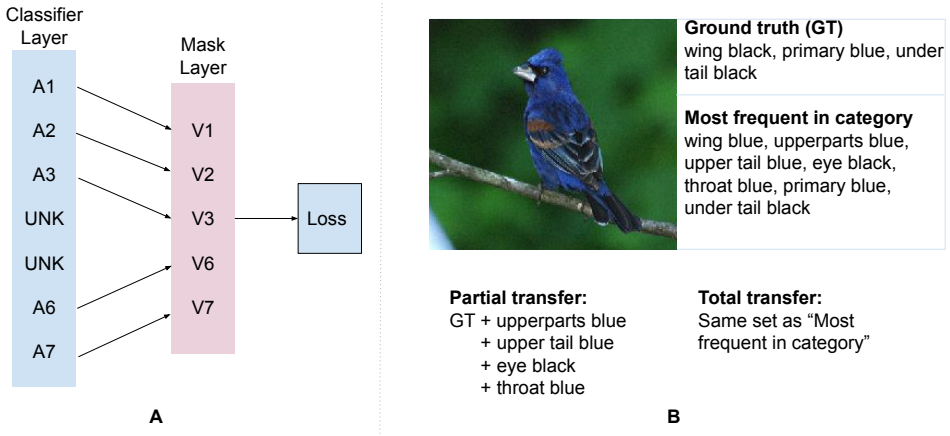



Figure 6.10: Strategies proposed to address the problem of "Very few samples have annotations for all attributes": A) By updating the loss function using a mask value that indicates which values won't be used in the computation. B) By transferring partially or totally attributes from the most frequent set of attributes for the respective class.

3. **The attribute-value per sample becomes a combination of all different values per all different attributes.** In the independent case, the performance was better for the attribute "Primary" by using all ground truth values in the training phase for the classification task. We consider each combination as independent and therefore, we pass the image as many times as it has different values. In the multi-task model, we must take into account all possible combinations per attribute and among all attributes, because the image is forwarded throughout all the tasks at the same time. This extremely increases the complexity of the sampler when selecting images. For example, if an image *i* has the set of values for the attribute "Primary" = *["Blue", "White"]* and the set of values for the attribute "Wing" = *["Blue", "White", "Black"]*, this gives us a total number of 6 possible combinations. This means the image is passed 6 times to the model if we use the same strategy as in the independent case. This is only for two different attributes with few variations. Thus, for the real case, the combination of 16 attributes with all their variations is not efficient and can be very large. To solve this problem and take into account all values during the training, the proposed solution is as follows: if an attribute has more than one value, only one of them is randomly chosen at each iteration. In the previous example, *image i* with two attributes, in the first iteration the combination passed to the model can be composed as *"Primary-Blue", "Wing-White"*. In a second iteration, the same *image i* can be passed with the combination *"Primary-White", "Wing-Black"*, and so on (see Figure 6.11).

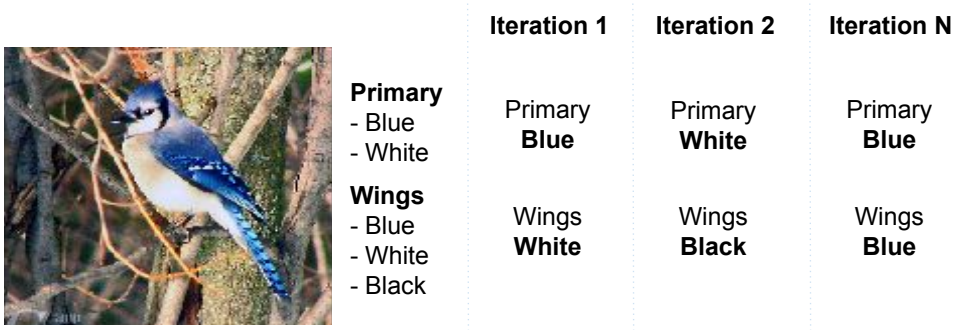



Figure 6.11: Choosing the attribute-values combination from all the ground truth set. In each iteration, only one value per attribute is randomly selected to train the model.

To analyze the performance in the multi-attribute case, we make use of two architectures that change drastically at the last layers: first, a multi-task model with two variations at the last layers,

in which the number of tasks is equal to the number of selected attributes (16 attributes for the use case database CUB). Second, a multi-label architecture in which the target representation for each image is a one-hot vector with dimension = number of attributes indicating the absence or presence of each one of them. We present these approaches in the following sections.

## 6.5 First multi-attribute framework: Multi-task models

This is the first multi-attribute model proposed. We proposed two variants, the first one to explore if only one classifier layer is sufficient to discriminate among all the attributes compared with a second one that includes more independent layers per attribute.

### 6.5.1 Proposed framework

This model along with its two variants is depicted in Figures 6.12 and 6.13, Page 132. The backbone structure in both variations is a ResNet50. For the first variation (Figure 6.12), the last layer of a ResNet50 is removed and a classifier layer is added with a dimension equal to the number of classes for each task (15 colors except for the attribute "Eye" with only 14 colors, for use case database CUB). This first variation is closer to the independent case. The only difference lies in the last layer, i.e., there is only 1 classifier with all attribute-color variations vs the 16 classifiers in the previous case, each one for one attribute.

For the second variation, the last two layers - the last convolutional block and the classifier layer - of the ResNet50 are removed. For each task (attribute), the same convolutional block is added along with a classifier layer with the respective dimension (Figure 6.13, Page 132). In this case, we wanted to determine if by having more layers for each task, the discrimination ability among them improves.

### 6.5.2 Experimental evaluation

For the multi-task models, two architectures are proposed. Table 6.4, Page 133 presents the summary results for models MT1 and MT2 with their variations. Experiments (1) and (2) present results for models MT1 and MT2 by using a mask for absence values when computing the loss

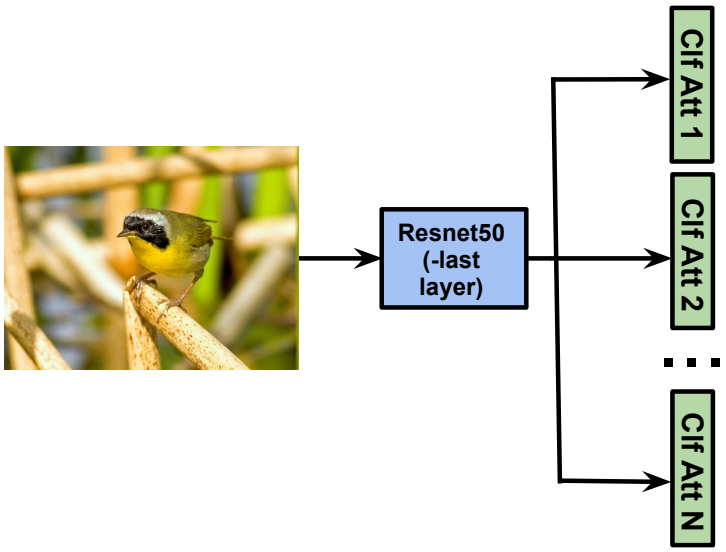



Figure 6.12: Multi-task model - MT1. In this scenario, the backbone is a ResNet50 with N different simple classifier output layers equal to the number of attributes in the database.

(see Section 6.4 - Strategy 1). The model MT1 performs slightly better and therefore, it is chosen for the following experiments. Experiments (3) and (4) present the results when transferring the most frequent values using the class information (see Section 6.4 - Strategy 2). In (3) a partial transfer is implemented, i.e., only the absent attributes of the sample are transferred. In (4), a total transfer is implemented, i.e., all samples belonging to the same class have the same attribute-values (see Figure 6.11, Page 129). The results demonstrate that the best solution is a partial transfer because the model is aware of all the possible annotations (colors) present in each sample.

To address the problem of data imbalance in the multi-task model, and because the sampling strategy is not directly applicable in this scenario (as stated in Section 6.4), more effort is required. We propose to do this through the loss function, which is easier to manipulate. In [302], they propose a new loss function for improving performance in single-stage object detectors. This function may also help in the problem of highly imbalanced classes by training the model with a focus on the hard negative examples instead of overlooking them and getting an arbitrary accuracy. This strategy is a more developed version of the cost-sensitive technique explained in Section 6.3 where weights are given to each class during the calculation of the loss. The loss function proposed in that work is called *Focal Loss* ($FL$) and it is defined as follows:

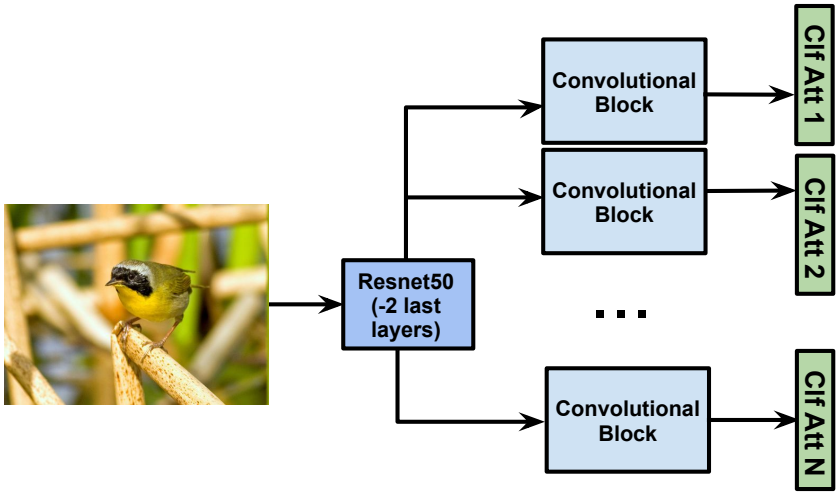



Figure 6.13: Multi-task model - MT2. This network is an adaptation of MT1. We added convolution blocks to each classifier layer to analyze if the discrimination among them increases.

$$\begin{aligned} CE(p_x) &= -log(p_x) \\ FL(p_x) &= -(1-p_x)^{\gamma} log(p_x) \end{aligned} \tag{6.2}$$

where $p_x$ is the model prediction for the input $x$, $CE$ is the traditional Cross-Entropy loss, and $FL$ is the Focal loss function. The key difference is the scaling factor ($(1-p_x)^{\gamma}$) that is added to the original definition of the Cross-Entropy loss ($CE$). This scaling factor decreases (or decays towards 0) as the confidence in a prediction goes up. The aim then is that by setting $\gamma > 0$, it reduces the relative loss for well-classified examples (i.e., $p_x > .5$), and then, the focus is on hard mis-classified examples.

Experiments 5 and 6 present the results when applying this method to the multi-task model MT1 and using different scaling factors ($\lambda = 2$ and $\lambda = 5$ respectively), with the first one having a better performance. To ease the evaluation of this strategy, the attribute "eye" is removed because this is the only attribute with a different number of classes. Table 6.5 presents results per attribute for all settings listed in Table 6.4.

Table 6.4: Results for multi-task models. Experiments (1) and (2) display the results for models MT1 and MT2 by using a mask for absence values when computing the loss. Experiments (3) - partial and (4) - total display the results when transferring the most frequent values using the class information. Experiments (5) and (6) display the results of applying different scaling factors for the Focal Loss function ($\lambda = 22$ and $\lambda = 5$ respectively).

| **Setting** | **Accuracy (%)** |
|---|---|
| 1 | 39,16 |
| 2 | 38,87 |
| 3 | **39,23** |
| 4 | 38,46 |
| 5 | 35,01 |
| 6 | 34,52 |

Table 6.5: Best results per attribute for multi-task models variations evaluated for the 16 attributes used from the database CUB.

| **Attribute** | **Acc (%)** | | | | | |
|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** |
| back | 31,77 | 31,44 | 32,17 | 30,96 | 31,20 | 31,08 |
| belly | 43,24 | 43,65 | 43,26 | 43,30 | 42,46 | 41,85 |
| bill | 35,90 | 35,88 | 35,74 | 35,74 | 35,76 | 35,68 |
| breast | 43,63 | 43,49 | 43,49 | 43,04 | 42,72 | 42,34 |
| crown | 38,68 | 38,60 | 38,79 | 38,02 | 37,98 | 37,31 |
| eye | 73,92 | 73,92 | 73,92 | 73,92 | - | - |
| forehead | 39,03 | 38,48 | 39,33 | 37,90 | 34,27 | 33,70 |
| leg | 22,13 | 21,78 | 21,92 | 21,56 | 19,48 | 18,78 |
| nape | 38,54 | 37,86 | 38,54 | 37,17 | 35,94 | 35,22 |
| primary | 50,58 | 50,08 | 50,67 | 48,67 | 49,33 | 48,41 |
| throat | 43,04 | 42,94 | 43,32 | 42,86 | 41,13 | 40,06 |
| under tail | 25,80 | 25,57 | 26,04 | 25,21 | 23,28 | 22,87 |
| underparts | 44,19 | 44,82 | 44,37 | 44,33 | 41,91 | 40,76 |
| upper tail | 22,17 | 22,05 | 22,55 | 21,96 | 18,31 | 17,73 |
| upperparts | 38,44 | 37,98 | 38,73 | 37,78 | 37,76 | 37,47 |
| wing | 40,02 | 40,04 | 40,14 | 38,60 | 39,49 | 39,65 |
| **Average** | 39,44 | 39,29 | **39,56** | 38,81 | 35,40 | 34,86 |

The results are coherent with the independent results. Those attributes with higher results in the independent case also have higher results in the multi-task case. However, the general performance is decaying to about $50\%$. The focal loss function is not well adapted, as explained in

Section 6.4 - Problems encountered, "very few samples have annotations for all attributes". In other words, it is very difficult to determine which are the rare classes among all tasks.

Figures 6.14, 6.15 and 6.16 present the overall learning curves for the best multi-task model (setting 3 in Table 6.5), followed by the specific learning curves for the attribute 'Primary', and the attribute with the lowest performance 'Leg', respectively. The overall learning curves converge after 60 epochs. This represents a saving in resource consumption in comparison with the independent case. Following similar behavior than in the independent scenario. The 'Primary' and 'Leg' attributes have a smooth convergence for both learning curves.

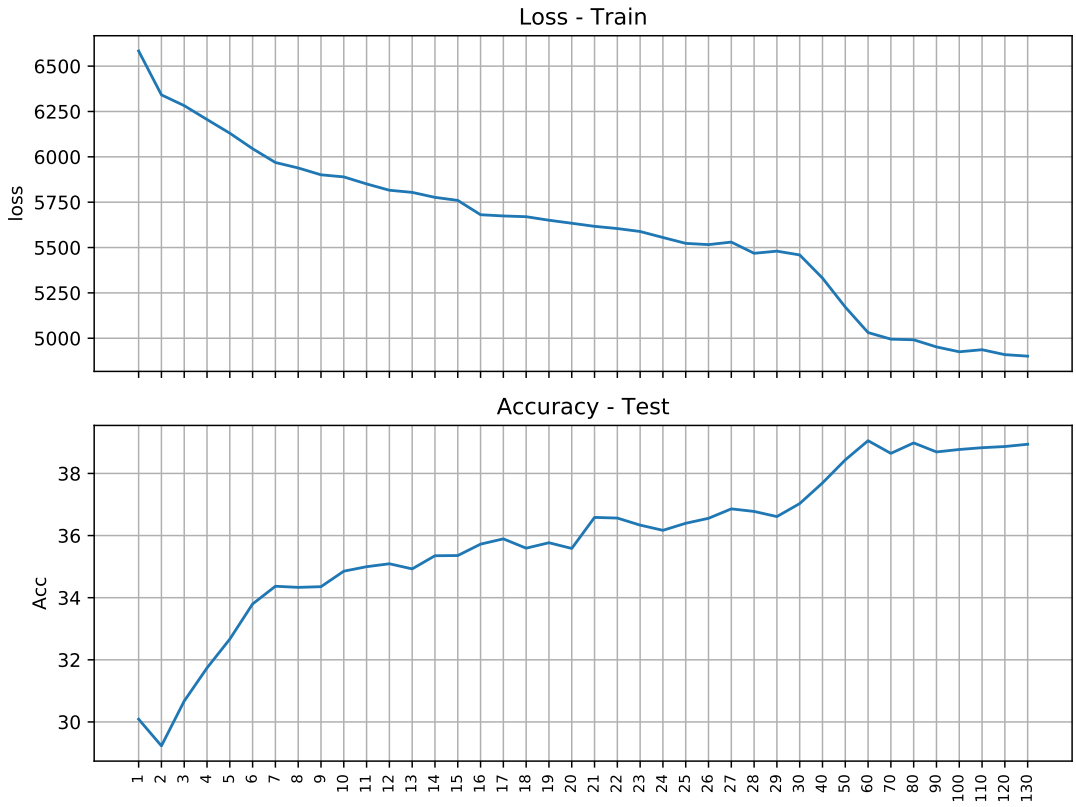



Figure 6.14: Overall learning curves for the best setting found in the multi-task scenario. The training loss and the accuracy in the test set converges after 60 epochs.

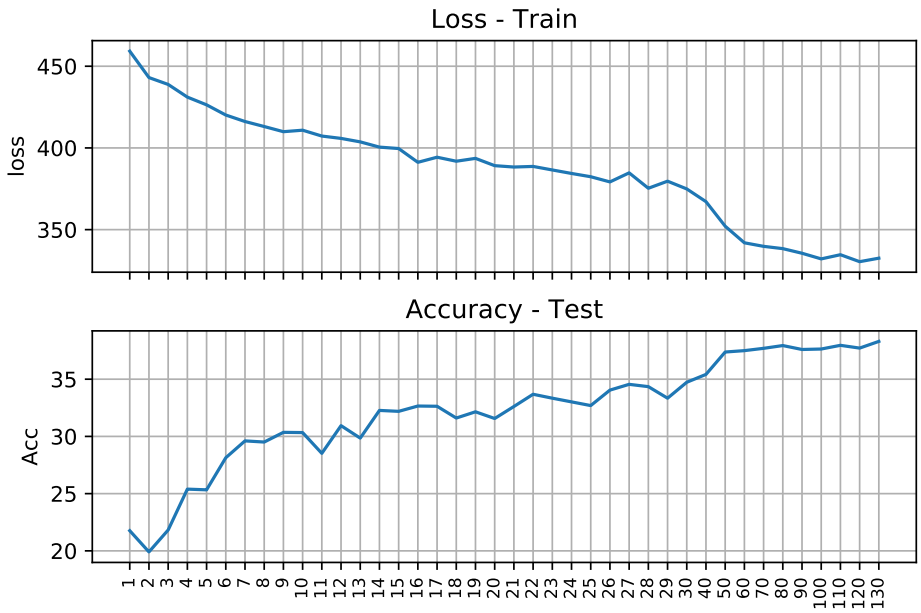



Figure 6.15: Learning curves for the 'Primary' attribute in the multi-task scenario. Similar than in the independent case, the convergence is smooth.

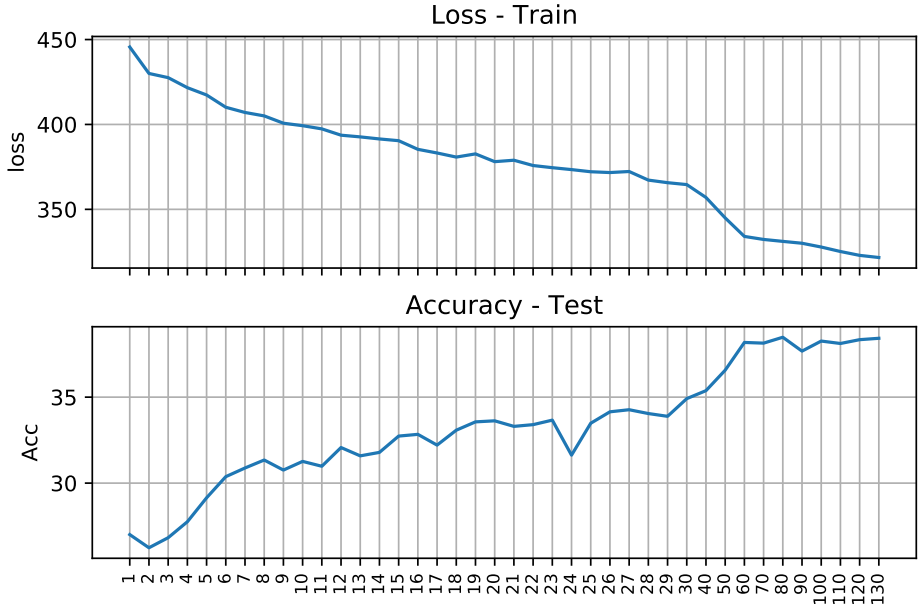



Figure 6.16: Learning curves for the 'Leg' attribute in the multi-task scenario. Similar to in the independent case, this attribute has the lowest performance but the learning curves have a smooth convergence.

## 6.6 Second multi-attribute framework: Multi-label model

This is the second multi-attribute model proposed. The multi-label setting for this use case is as follows: the dimension of the output is equal to the number of different attribute-values (labels) in the database (239 different labels). Next, we present the proposed framework along with the experimental evaluation.

### 6.6.1 Proposed framework

Similar to the independent scenario, the ResNet50 model is used with the weights initialized using a pre-trained network on ImageNet. The dimension of the output is the number of different attribute-value in all the databases. For example, in the CUB database the type of attributes is as follows: *"Primary-Red", "Primary-Blue", ...+ "Black-Red", "Black-Blue", ...+ "N_attribute-Color1", "N_attribute-Color2", etc* with a total number of 239 attributes related to color information (see Figure 6.17). Because this is the most straightforward model, we use this architecture to compare it against traditional machine learning models. We report results for all three databases.

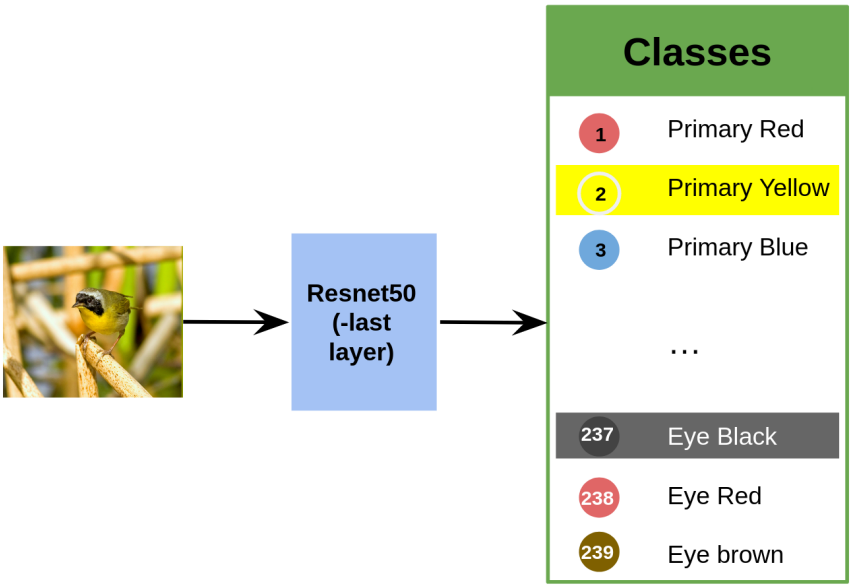



Figure 6.17: Multi-label model. In this setting, only one model is trained for all the attributes used from the CUB database that refers to color information. In total, there are 239 attribute-color possible outputs.

The dimension of the output is equal to the number of different attribute-value (labels) in the database (239 different labels). Therefore, the target is a one-hot vector with 0s and 1s representing the absence or presence of each attribute in the image. We evaluate different settings that aim to address the imbalance problem either from the loss function or from the sampling method or combined solutions. Similar to the independent case, we also test sampling techniques with weights computed in different ways. More precisely, we use a partial and total transfer of attribute data, and we make a comparison of loss functions including cross-entropy, the F1 score based loss function, and a focal loss function [302] (the latter was presented in Section 6.5).

### 6.6.2 Experimental evaluation

The best results were obtained by using a partial transfer and a focal loss function (see Table 6.6 - Deep Multilabel). One of the main problems in this scenario is described in Section 6.4 - Strategy 2: "Very few samples have annotations for all attributes". Some images have only 1 label, while others may have many labels with different values associated. This makes some representations very sparse, affecting the learning of features mainly for rare classes. By doing a closer analysis of the results, we see the loss function directly affecting the number of true positives (TP), false positives (FP), and false negatives (FN). When using the F1 score based on the loss function, the amount of FP increases in large amounts, while by using a focal loss Function, the model opts for not assigning a label (increasing FN) if the confidence in the prediction is not high enough (decreasing FP). With weighted F1-score the result is of 23,97% for the first settings explored to 41,42% with focal loss. The increase in performance is about 42%. Although for the use case database, the performance is still quite low, this result points us towards the best direction, with the use of the focal loss function (or an optimal function) for further research. Figure 6.18 presents the overall learning curves for the multi-label scenario for the CUB database. Although the performance is low (see Table 6.6 - Deep Multilabel)), the convergence for the training loss and the f1-score in test is very smooth.

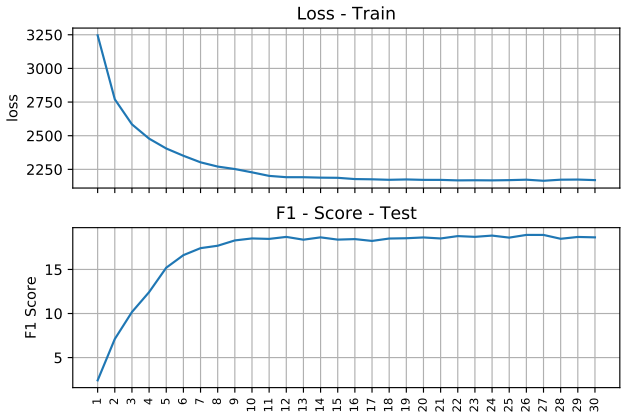



Figure 6.18: Overall learning curves (all attributes learnt with one model) in the multi-label scenario for the CUB database. The curves have a smooth convergence.

### 6.6.3 Performance comparison with other machine learning methods

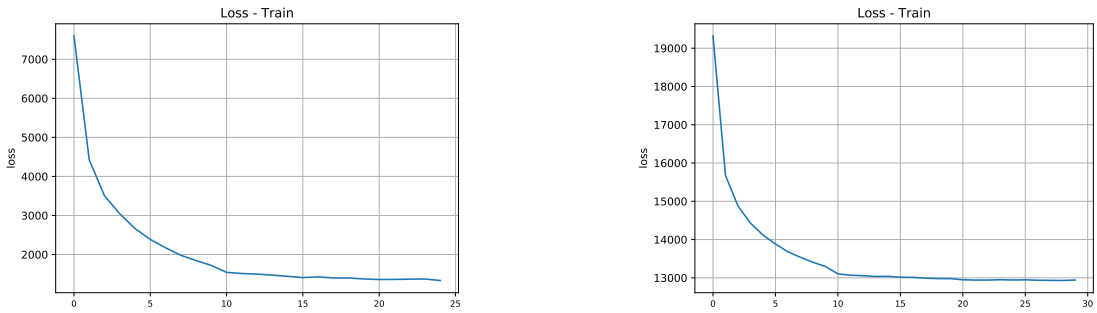
This section presents the comparison vs well-known machine learning methods for the three selected databases, in the task of multi-labeling. We selected standard models, easily reproducible, and widely used in the context of imbalanced learning according to the survey work presented in [257]. The selected methods are Support Vector Machines (SVM) [303], MLKNN [304], Logistic Regression (Log Reg), Random Forest (RandomF) [305], Decision Trees (DecisionT) [306], Extreme Learning Machine (ELM) [307], and Gaussian Naive Bayes (GaussianNB) [308]. We use 5-fold cross-validation to find the best set of parameters per database for each model. Table 6.6 presents the results of the comparison against traditional machine learning models for multi-label learning, for the three databases selected. A non-computer science expert can opt for using one of these models if the performance required is good enough, especially, in well-structured databases (such as AwA2 and celebA). However, there is a clear improvement in the performance when using deep learning models. For the multi-label context, more adequate measures are reported (precision, recall, and F1-score). The metrics reported include macro average (averaging the unweighted mean per label) and weighted average (averaging the support-weighted mean per label).

The main reason for the bad performance in the use case database CUB still seems to be the imbalanced problem (see results for the other two databases). And, like multi-task results, the

Table 6.6: Comparison against well-known machine learning models in the task of multi-label.

| | | CUB | | | AwA2 | | | CelebA | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Model** | Precision | Recall | F1-score | Precision | Recall | F1-score | Precision | Recall | F1-score |
| **Macro Avg** | SVM | 24,15 | 27,52 | **22,22** | 58,24 | 59,84 | 51,17 | 50,66 | **79,26** | 59,50 |
| | MLKNN | 13,04 | 12,28 | 12,48 | 53,82 | 55,71 | 47,02 | 41,60 | 41,25 | 40,61 |
| | Log Reg | 29,59 | 10,71 | 14,21 | 58,08 | 60,46 | 51,47 | 65,16 | 46,57 | 51,65 |
| | RandomF | **79,31** | 1,28 | 1,92 | 64,30 | 51,88 | 46,33 | 58,15 | 23,42 | 27,58 |
| | DecisionT | 10,29 | 7,15 | 8,08 | 51,60 | 52,92 | 44,43 | 42,73 | 28,49 | 32,09 |
| | ELM | 7,71 | **32,63** | 10,34 | 40,38 | 56,68 | 41,06 | 31,52 | 45,52 | 34,73 |
| | GaussianNB | 30,30 | 14,54 | 17,04 | 55,32 | 54,08 | 45,95 | 44,32 | 53,98 | 46,53 |
| | **Deep Multilabel (ours)** | 23,01 | 17,99 | 18,71 | **77,67** | **70,61** | **73,31** | **78,02** | 70,50 | **73,15** |
| **Weighted Avg** | SVM | 35,49 | **51,10** | 41,12 | 72,87 | 66,80 | 67,14 | 67,02 | **80,56** | 71,42 |
| | MLKNN | 26,84 | 26,70 | 26,67 | 69,06 | 63,69 | 63,47 | 55,23 | 58,52 | 56,48 |
| | Log Reg | 47,06 | 21,43 | 26,94 | 73,26 | 66,98 | 67,41 | 72,55 | 62,61 | 65,09 |
| | RandomF | **50,61** | 7,49 | 9,29 | 74,17 | 63,17 | 62,47 | 65,42 | 45,78 | 49,20 |
| | DecisionT | 23,35 | 18,65 | 20,62 | 67,85 | 61,26 | 61,56 | 58,63 | 49,77 | 51,93 |
| | ELM | 21,98 | 37,15 | 25,86 | 61,34 | 56,96 | 57,16 | 52,36 | 56,51 | 53,32 |
| | GaussianNB | 41,49 | 26,49 | 30,87 | 69,91 | 62,55 | 62,53 | 57,48 | 64,66 | 59,38 |
| | **Deep Multilabel (ours)** | 46,00 | 40,73 | **41,42** | **83,19** | **78,36** | **80,05** | **84,35** | 77,12 | **79,67** |

strategies for multi-class problems do not perform well in a multi-label scenario. The labels with more support samples are still the only ones adding to global metrics. Notice that the macro average metric does not consider the number of samples of support that the metric uses to compute the performance. While the weighted metric does use this parameter, this allows us to better understand the results obtained: the more examples of support, the better the performance. For well-structured databases, AwA2, and CelebA, the performance for deep learning models is higher. Figure 6.19 presents the training loss in the multi-label model for the databases of AwA2 and CelebA. The learning curves have a very smooth and rapid convergence showing the capability of the proposed models under well-distributed databases.




AwA2 CelebA

Figure 6.19: Training loss for AwA2 and CelebA databases in the multi-label scenario. For both databases, the training loss have a very smooth and rapid convergence.

## 6.7 Discussion

In this chapter, we explored the problem of fine-grained classification from the point of view of non-computer science experts. Non-computer science experts may struggle to use highly complex deep learning models, which may offer performance gains, unfortunately with results that are surprising and hard to understand. Therefore, in our work, we see these models as black boxes and analyze the data in the specific case of very fine-grained attribute classification with an imbalanced database as our main use case. Many recent works using attribute data assume ideal conditions. For example, they expect the combination of attributes related to a class to be present in all images annotated [240]. In real databases, such as our selected use case database CUB, this cannot be assumed. The subset of attributes for images belonging to the same class can also be different, making the problem even more difficult. After performing a large exploration of classical imbalance learning strategies in the independent scenario, the results indicated that these strategies were not adapted to multi-task or multi-label tasks. These are adequate for the general problem of classification, under ideal conditions present in the database. Therefore, we presented three different strategies for the optimal use of multi-attribute frameworks. To evaluate our strategies, we proposed two multi-attribute frameworks: multi-task and multi-label. The latter one allowed us to make a fair comparison with the selected methods for which we obtained competitive performance. We found that the strategy that presents the best improvement is concerning the loss function, in particular, a focal loss function that increases the number of false negatives and decreases the number of false positives, if the confidence in the prediction is insufficiently high.

In the next chapter, we present the strategies proposed to tackle another important problem concerning the data: an insufficient amount of annotated data. Specifically, we address the problem of propagating attribute annotations from classes to single images in a semi-supervised manner. Our approach allows us to extend the annotations available in an image collection with diverse vocabularies without having to annotate individual images manually.

# CHAPTER 7

# Attribute Discovery

## 7.1 Motivation

In previous chapters, we mentioned two main related problems that affect the model performance: learning with highly imbalanced databases, and learning when few annotated data is available. In Chapter 6, we addressed imbalanced learning. We analyzed and proposed strategies for the problem of imbalanced learning in the specific case of multi-attribute learning. In this chapter, our goal is to develop strategies around multi-attribute learning that take advantage of high-level information when little data is annotated. This chapter is principally exploratory, and with several experiments and strategies proposed, we aim to tackle the problem of having very few annotated data at the attribute level and for several attribute spaces. First, let us remind that multi-attribute refers to the learning of multiple types of attributes (or attribute spaces). For example, one attribute set may refer to the color of an animal's parts, while another attribute set may refer to the action displayed on a piece of visual data such as an image. The dominant paradigm for visual recognition is based on supervised machine learning, where a set of fully

annotated images in a fixed number of classes is required. The vision community has created several large-scale data sets with an increasing number of examples and classes [8], which have been critical for moving the field forward. In particular, the existence of such data sets has allowed training accurate deep learning models that are reused as feature extraction methods and transferred to other tasks and domains. Annotating data sets with class labels is widely adopted practice in many domains of computer vision, and the process is generally accepted as a first step to organize an image collection. Nonetheless, having more information helps in making more robust models. In the previous applications addressed in this thesis, we show how having extra information can help improve the performance of the methods. For example, in Chapter 4, we addressed the task of information retrieval, for which we had general descriptions about images. It would be advantageous to have more concrete textual descriptions about the images that help to discriminate among the different classes. In Chapter 5, we also made use of a complementary module that uses the available data, in that case, OCRs extracted from the image. Sending to the model complementary texts that describe the visual objects in the image may help in defining the scenario from which the image was taken. However, extending data set annotations with more structure and including attributes or multi-label tags increases the manual annotation complexity. This is because of combinatorial interactions among labels that need to be considered by manual annotators, resulting in inefficient or expensive processes.

In the previous chapter, we showed how attributes are important because they can complement class-level recognition and therefore improve the degree to which computers perceive visual concepts. Thus, we propose to develop a model that learns to assign attributes from multiple attribute spaces to single instances (single images or a group of images displaying the same visual concept). As we mentioned in the previous chapter, the state of the art in attribute learning has been explored extensively and for different purposes and applications (see Section 3.4). We showed that attribute data is largely used as a type of supervision to perform fine-grained classification or simply as an auxiliary piece of information. Another flaw when working with attribute data is the assumption of having well and sufficiently annotated databases. In reality, these conditions are rarely met in practice in these types of databases. To address this issue, several new research directions are being explored, including self-supervision [309], weak supervision [310] and few-shot learning [311], [312]. In particular, there is considerable research around zero-shot

learning algorithms to classify new, unseen object classes from attributes, which aim to leverage visually encoded properties of the objects at test time, instead of just the classes[1]. In the zero-shot learning paradigm, attributes are annotated by class, therefore, reducing the labeling cost significantly [271]. However, the attribute-level descriptions used for zero-shot learning are usually obtained from abstract vocabularies that make sense from the semantic perspective of classes and not for individual images. For example, these attributes are sometimes not necessarily visual properties of objects (e.g. *solitary*, *smelly*), and at other times they are not visible in every single image that belongs to the class (e.g. *eating*, *group*). Another problem under the zero-shot learning setup is that the testing instances are assumed to come from the unseen classes only. This problem setting is somewhat unrealistic. The ideal learning scenario is that the testing instances can come from both the seen and the unseen classes [276]. In summary, we found the following flaws in the state of the art (see also Figure 7.1): 1) attribute data is merely used as auxiliary information; 2) attribute databases are generally annotated only at the class level; 3) the attributes may refer to abstract descriptions instead of useful visual concepts; 4) and finally, the zero-shot learning paradigm is class-oriented, i.e., its objective is to annotate unseen classes instead of attributes. In our case, target annotations are for seen and unseen classes because our target annotations are at the sample level. Instead, we use class-level data as auxiliary information.

Taking this into account, our strategy is to develop a model which uses visual concepts as input and that predicts attributes from multiple heterogeneous spaces in a semi-supervised fashion. In Figure 7.2, we present a general overview of our proposed framework. Semi-supervised learning permits harnessing the large amounts of unlabeled data available. Typically, the semi-supervised feature is explained by the use of smaller sets of labeled data [65] (mainly unlabelled data and a small set of labelled data). This learning approach is getting more attention as there exists many data sets with little to no annotations. Similar to zero-shot learning, we exploit the relationships that classes have, given the attributes (see Figure 7.3, Page 145, for an example of an attribute in multiple classes), and use this knowledge to initialize annotations in the training set and to refine or improve them iteratively in a semi-supervised way. It means, at each iteration, we rely

[1] A shot refers to a single sample available for training. In the N-shot setting, N samples are available for training. For few-shot learning, the number of samples usually lies between zero and five. For zero-shot learning, there is no example.

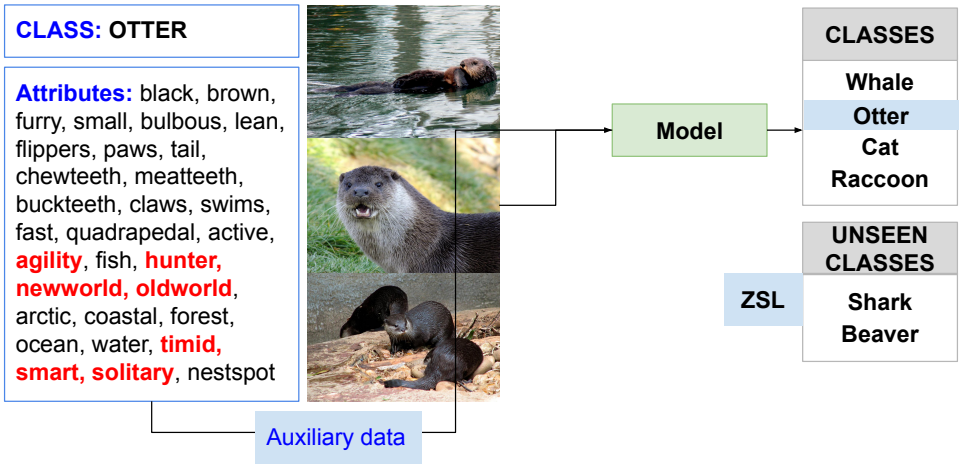



Figure 7.1: Examples of the use of attributes in the state of the art. Attributes are used as intermediate data. Attributes are defined only at the class level. Not all images in a class display the set of attributes. In red, attributes that are abstract and not visual descriptions of the visual content (images extracted from the database AwA2).

at some level on the model predictions. In this way, we propose to use the high-level information that we have at our disposal.

The main obstacle is the lack of rich and diverse databases annotated at the level of single instances for the training phase (as previously mentioned, this is the case of most attribute databases in the domain of person identification [243]). Therefore, we also propose a sufficient set to explore and analyze our strategies. This set is derived from the database AwA2 [234]. The problem with the AwA2 database is its abstract and non-visual set of original attributes. We can find attributes such as 'solitary', 'newworld', or 'oldworld' that do not describe the visual

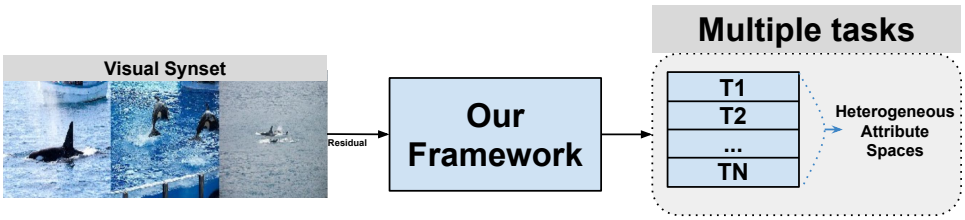



Figure 7.2: Our simplified framework. The model receives residuals which represent visual concepts. Then, multiple tasks associated to multiple attribute spaces are trained. The objective is to assign the optimal set of attributes to images from class-attribute relationships in a semi-supervised learning fashion.

Figure 7.3: Examples of the semantic attribute "Eating" for different classes.

concepts typically annotated for images. For this, we define a new and improved set of vocabularies of heterogeneous attribute spaces and tags that are associated with classes (see Figure 7.4 for a comparison of the original set of attributes in the AwA2 database vs attributes extracted from our proposed sets). We can use the original class-level information and then assume that all images in our training set have one class label ('Whale', 'Elephant', 'Gorilla', etc). With this assumption, we can define a set of optional properties that images of those classes may have. State-of-the-art proposals [252, 277] use heterogeneous attribute spaces as well. However, they rely on the important assumption that annotations are available in the training phase. In our case, without attribute annotations at the image level, our goal is to learn annotation models to propagate the attributes to images as in Figure 7.4. Both images belong to the same category 'Elephant', but the target sets of visual attributes displayed are different.

To summarize, the contributions and results in this chapter are the following:

- We formulate the problem of assigning attributes to images from class-attribute relationships as a semi-supervised learning problem using little (we manually annotated up to 5 samples per attribute) or no manual annotations. The proposed algorithm, which propagates attributes that are consistent among classes, is described in Section 7.2.

- During the training phase, our algorithm makes guesses of which attributes are likely to be assigned to images. We explored several strategies to improve the initial set of annotations

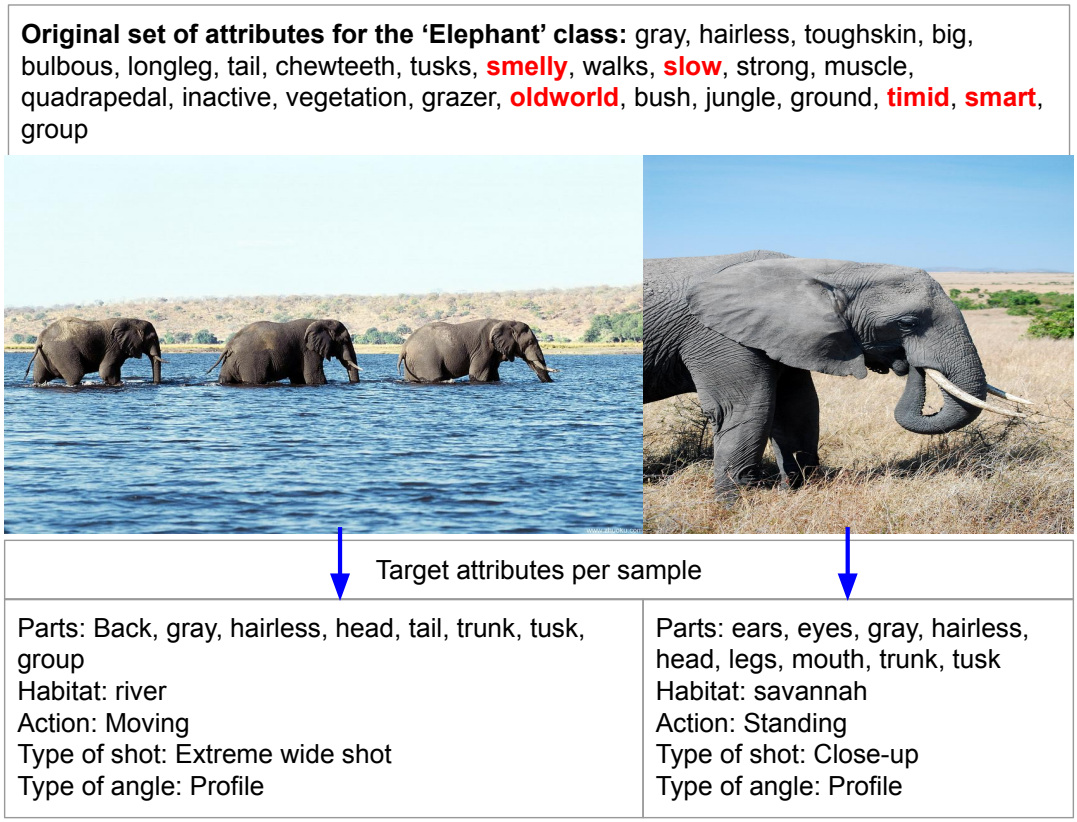



Figure 7.4: Comparison of non-visual attributes from the original set for the AwA2 database vs the set of attributes extracted from our proposed sets.

with none or less than 1% of manually annotated samples (Section 7.2.2).

- Our formulation allows us to incorporate attributes from multiple vocabularies, which can have mutually exclusive labels or multi-label tags. These attribute spaces allow for richer annotations and are modeled using multi-task learning (Section 7.2.4).

- Using the Animals with Attributes 2 (AwA2 [234]) data set, we created sufficient data sets to carry on a complete experimental evaluation. We defined five different attribute spaces (vocabularies) for 40 training classes (Section 7.3.1). Our methodology can facilitate the propagation of attributes to any image set organized in classes.

- Our results show that by using the information at the level of class-attribute relationships, we can learn models that recognize the correct set of attributes for test images. We found that the initialization strategy for semi-supervised learning is the most important step for improving performance and assigning attributes (Section 7.3.5).

## 7.2 Proposed approach

We first describe the general setting of our semi-supervised learning task and define the notations used throughout the section. Let us assume there exists the following sets of data: $X$ is a collection of images associated with a set $C$ of one-hot encoded class labels. Each image $X_j \in X$ is the associated embedding of an image computed with a convolutional neural network (ResNet50 [87] in our experiments). Also, let $A$ be the set of attributes spaces extracted from different vocabularies $V$. Figure 7.5 shows the relationships shared among all these entities.

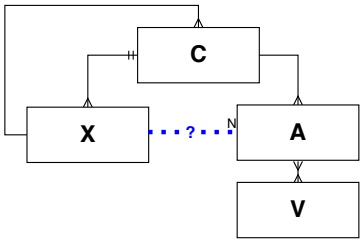



Figure 7.5: Data model: we assume a collection of image embeddings $X$, each one associated with one class from the set $C$. In turn, each class has associated a set of attribute vectors from $A$, which are extracted from different vocabularies $V$. We aim to transfer attribute annotations to individual images, a relationship represented in the diagram with a dotted blue line.

Our model relies on aggregated visual representations of semantic concepts that we call *visual synsets*[2]. We define a synset as the average representation of a group of images with similar attributes. The number of images in the synset can be one or many, and the number of shared attributes can also be one or several. According to the data model, we know the class labels associated with samples taken from $X$, thus, we can compute the synset representation of classes with $n_i$ as the number of samples belonging to a class, as follows:

$$\bar{X}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij} \qquad \forall X_{ij} \in X_i \quad i \in C \tag{7.1}$$

Now, we have the original image embeddings $X$ (divided into training $X_{train}$ and validation $X_{Val}$), and class synset representations $\bar{X}$ (one representation per class). We will estimate attributes for synset representations as described in the following sections. Our final goal is to

[2]Following the ImageNet terminology for groups of similar images

learn the relation $X \rightarrow A$, i.e, to find the best subset of attributes $A_s$ that describes images in $X$. To pursue this, we propose a model that can be trained in a semi-supervised fashion (see Figure 7.6 for a visual example of the relationships in our databases). Table 7.1 introduces the most important notations used in this work.

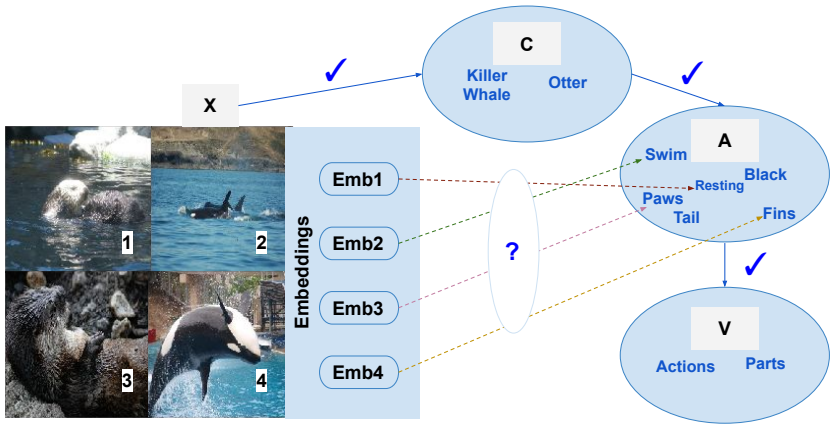



Figure 7.6: Data model with visual examples: we have a collection of images and their associated embeddings $X$, each one associated with one class from the set $C$. In the example, these are represented by two classes: Killer Whale and Otter. Each class has associated a set of attribute vectors from $A$, which are extracted from different vocabularies $V$. In the example, the two types of vocabularies and their associated attributes are Parts [paws, tail, fins] and Actions [swimming, resting]. We aim to transfer attribute annotations to individual images to find relationships (represented in the diagram with dotted lines) that are unknown.

| Notation | Description |
|---|---|
| $X$ | Collection of image embeddings |
| $X_{train}$ | Image embeddings for train |
| $X_{Val}$ | Image embeddings for Validation |
| $X$ | Synset of X |
| C | Class labels |
| A | Attribute spaces |
| $A_s$ | Subset of A |
| V | Vocabularies |
| $R$ | Residual between images and the synset of their classes |
| $\mathcal{N}$ | Neighborhood of the synset |
| MC | Multi-class |
| ML | Multi-label |
| p | Parts |
| h | Habitats |
| a | Actions |
| sc | Scales |
| an | Angles |
| n(GT) | Length of the ground truth set |
| $\bigcup$ | Union |
| $\Omega$ | Valid subset |
| $cos$ | Cosine similarity |
| $max$ | Maximum |
| $th_a$ | Threshold for attribute a |
| $\Lambda$ | AND |
| $E$ | # of epochs |
| $n_{it}$ | # of iterations |
| $L$ | Loss |
| $Init$ | Initialization strategy |
| $A1$ | Re-annotation strategy A1 |
| $A2$ | Re-annotation strategy A2 |
| $GT$ | Ground truth |
| $P$ | Predictions |
| $K$ | # of positive predictions to take in evaluation |
| $BS$ | Batch size |
| $LR$ | Learning rate |
| $SGD$ | Stochastic Gradient Descent |

Table 7.1: Specific notations used throughout the section.

### 7.2.1 Inputs and outputs

This section describes the input and output data required in our approach. In what follows, we assume that attributes are associated with more than one class (see Figure 7.3, Page 145). Attributes that are only associated with a single class are not useful in our approach, because they can be considered as class synonyms that do not bring any additional information concerning a

class label. Also, we assume that images of a class do not necessarily display all the attributes associated with the class (i.e., we assume that an example image of a class only has a subset of the attributes of that class).

**Inputs.** Our goal is to transfer attribute labels from different vocabularies to individuals or groups of images. We assume that the visual representation (features) of an image can be decomposed into two parts: 1) common features of the class and 2) image-specific features. We achieve this decomposition by computing simple *residual* vectors $R$ between images and the synset of their class:

$$R_i = \sum_{j \in \mathcal{N}} X_{ij} - \bar{X}_i \tag{7.2}$$

where $\mathcal{N}$ is the neighborhood of an arbitrary synset. The neighborhood is computed by using a similarity metric to find the closer embeddings. The main motivation for computing residual representations is that this allows us to separate class-specific features from other features that encode other aspects of the image. We hypothesize that the residual vectors encode class-independent features relevant to identify attributes that are shared among classes. Note that the residual representation can be computed for groups of images or for individual images (when $|\mathcal{N}| = 1$).

**Outputs.** The output is the set of attributes $A$. In our approach, they come from different vocabularies $V$. We design the model with a multitask architecture, which supports multi-class[3] (MC) outputs or multi-label[4] (ML) outputs. In our approach, we use the database AwA2 [234] as our main source of data and created 5 attribute spaces associated with different vocabularies. The name and the type of vocabularies for learning are as follows (# labels / task type): Animal parts (42 / ML), Habitats (18 / ML), Actions (5 / ML), Object scale (3 / MC), and Camera angles (6 / MC). These vocabularies represent a total of 67 independent tasks. For details on the construction of each attribute space, see Section 7.3.1 - Vocabularies.

[3]Exclusive, one-hot encoding outputs (for example, if the main action of the sample is 'Moving' among the six possible actions. Then the encoding is equal to [0,0,0,1,0,0], where the 4th space represents the action 'Moving').

[4]Multiple labels allowed, i.e. independent binary classifiers

### 7.2.2 Label initialization

As our main research problem is to find the relationship $X \rightarrow A$ as illustrated in Figure 7.5, Page 147, i.e., to automatically infer attributes to individual samples, we model this relationship as a function of the inputs to obtain the outputs. This function is parameterized by a neural network, as illustrated in Figure 7.10, Page 156, which is trained with a supervised loss function using stochastic gradient descent. Since our training data does not have attribute annotations for individual images (there are only defined at the class-level), we propose to initialize attribute annotations for training using different strategies. Note that attribute annotations can have three states: positive (an image has the attribute), negative (the image does not have the attribute), and unknown (the attribute belongs to the class but was not assigned).

**A. Partial Random.** For images in the set $X_{train}$, we randomly assign $p$ positive labels from each vocabulary to the $k$-th image (or synset):

$$A_k = \bigcup_{A \in V} random(\Omega(A, C_k), p) \tag{7.3}$$

where $\Omega(A, C_k)$ is the subset of attribute annotations for the class $C$ of the $k$-th image (i.e., those defined at least at the class-level), according to the class-attribute relationships defined in the data model. Valid attributes not selected as positive annotations at random are not considered negative annotations: we simply assume that they are unknown and do not penalize the multitask loss for those missing annotations. Also, invalid attributes, i.e., those not associated with the image class, are all considered to be negative.

**B. Using annotated samples per attribute.** We collect a small number of annotations for synsets in the subset $X'_{train}$ (as we will refer to this set in future mentions), which represent examples of each of the attributes of interest. In our case, we collected 5 examples from different classes for each of the 67 tasks of interest and used them as seed annotations for semi-supervised learning. These collected labels do not make a comprehensive set of attribute annotations for these training samples; therefore, any missing attribute label is considered unknown and not penalized during training.

**C. Pairing samples with the same class-level attribute.** As stated in [313], correlating attribute data among samples helps. Then, in addition to assigning positive attribute annotations for images in the subset $X''_{train}$ (either at random or with ground truth), we expand these annotations by finding similar pairs in other classes that may share the same attributes. In this case, we follow the principle of transitivity:

$$if\ (R_i \approx R_j)\ \ and\ \ (X_i \rightarrow A_{ik})\ \ then\ \ (X_j \rightarrow A_{ik}) \tag{7.4}$$

We assume that $X_i$ and $X_j$ are two images (or synsets) from two different classes $C_i$ and $C_j$ that satisfy the class-attribute relationship for attribute $k$ according to the data model: $|\Omega(A_k, C_i) \cap \Omega(A_k, C_j)| = 1$. For example, one sample that belongs to the class 'Elephant' has assigned the attribute 'group'. Then, we selected another sample (by using the cosine similarity) from one class that also contains the attribute 'group'. Then, we assign the same set of attributes for the two samples because we know they share visual and semantic similarities. The two images are similar according to the cosine similarity metric in the residual feature space ($cos(R_i, R_j)$). This strategy is useful to propagate labels in different classes that are known to share common attributes. The similarity metric ensures that the residual representations $R_i$ and $R_j$ have common class-independent features. The search for these pairs is performed with the example $X_i$ as a query in the subset of images $C_j$. The label propagation is performed using the transitivity rule with the nearest neighbor in that subset.

### 7.2.3 Prediction strategies

For each attribute space (where attributes $A_p$ = parts, $A_h$ = habitats, $A_a$ = actions, $A_{sc}$ = scales and, $A_{an}$ = angles), we can calculate the new annotations for each synset $j$ in two different ways:

1. **By using all attributes predicted as positive.** For the attribute spaces Parts, Habitats, and Actions that are binary tasks per attribute, we annotate the sample with the attribute $a$ if the model predicts this attribute as positive. For the other two attribute spaces Scales and Angles, we choose the index of the attribute with the larger prediction value. The latter is

learned by two multi-class tasks. This strategy is depicted in Figure 7.7.

2. **By applying thresholds.** For the attribute spaces Parts, Habitats, and Actions that are binary tasks per attribute, we annotate the sample with the attribute $a$ if the model predicts this attribute as positive, and if the prediction value $P_a > th_a$, where $th_a$ refers to the threshold of the attribute space $a$ in [p, h, a], for the attribute spaces Parts, Habitats, and Actions that are binary tasks. We computed the thresholds by taking the prediction values for all samples per attribute space predicted as positive (=1) and establishing a percentile value of 10%. With this setting, we can filter the set of predicted attributes as positive but with very small prediction values (inspired by the work of [278]). For the attribute spaces Scales and Angles, as in the previous strategy, we choose the index of the attribute with the larger prediction value. This strategy is depicted in Figure 7.8.

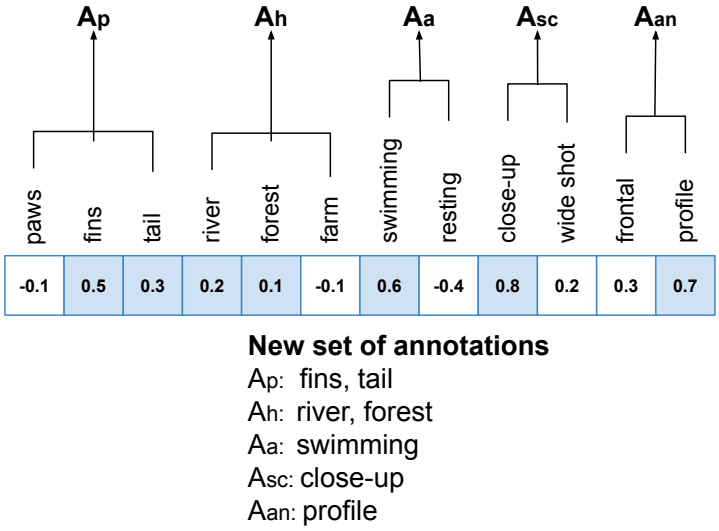



Figure 7.7: Example of the operation of the first prediction strategy. We annotate the sample with the attribute $a$ if the model prediction ($P_a$) for it is positive (=1), for the attribute spaces Parts, Habitats, and Actions (binary tasks). For the attribute spaces Scales and Angles, we choose the index of the attribute with the larger prediction value (multi-class task).

**Re-annotation strategy.** Figure 7.9, Page 155 presents the steps in our approach when we use a re-annotation strategy: we compute a set of initial annotations T1 using one of the strategies presented in Section 7.2.2 ("1 - by using all attributes predicted as positive" and "2 - by applying

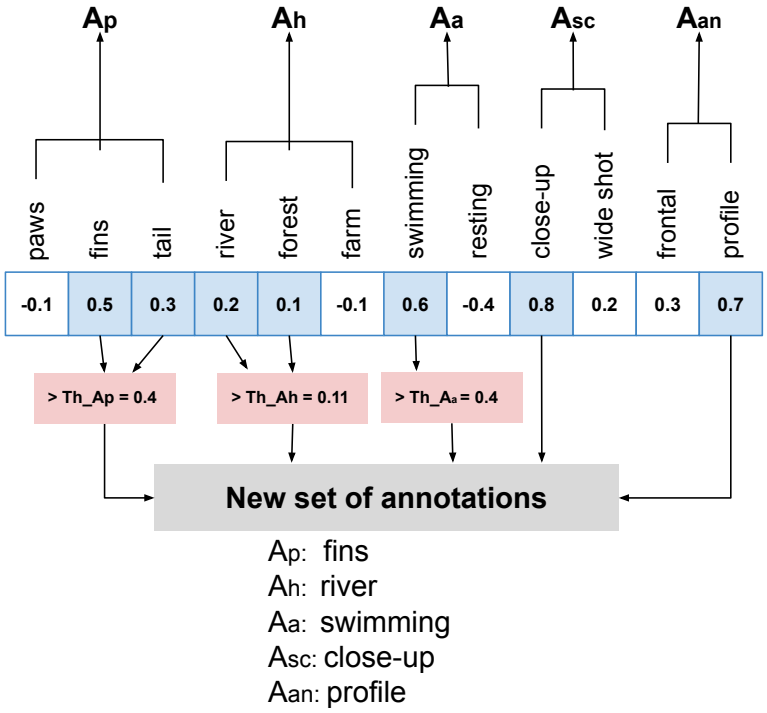



Figure 7.8: Example of the operation of the second prediction strategy. We annotate the sample with the attribute $a$ if the model prediction ($P_a$) for it is positive (=1), and if the prediction value $P_a > th_a$, where $th_a$ refers to the threshold of the attribute space $a$ in [p, h, a], for the attribute spaces Parts, Habitats, and Actions (binary tasks). For the attribute spaces Scales and Angles, we choose the index of the attribute with the larger prediction value as in the previous strategy (multi-class tasks).

thresholds"). Once we have our initial annotations, we train our model for a fixed number of epochs $E$. Once the model is trained, we can re-calculate the set of annotations, this time for all the samples in $X_{train}$ using strategies presented in *Section 3.4. (Prediction strategies)*, and re-train the model with them (process T2). We can repeat the process for a set of fixed iterations $n_{it}$ (it = iterations).

### 7.2.4 Architecture

Figure 7.10, Page 156 presents the architecture of the multi-attribute spaces model (*MultiAttS*). The encoder is composed of a set of layers with dimensions starting at 2,048 (dimension of the input embeddings) and ending with a dimension of 128 to be forwarded to each independent

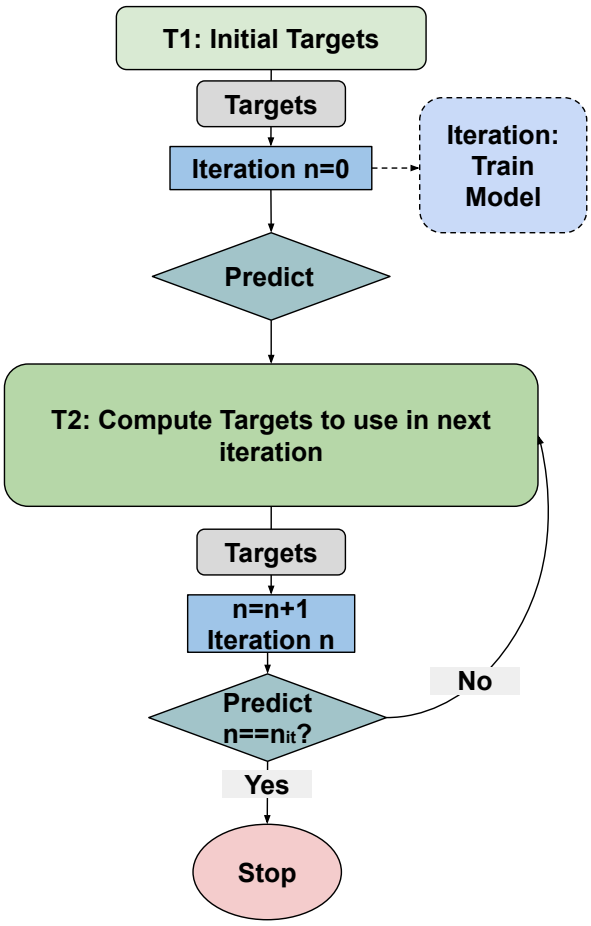



Figure 7.9: Re-training process: an initial set of annotations is inputted into the model. After iteration 0, we re-compute the annotations by using the predictions of the model. We re-train the model and repeat the process for a predefined set of iterations $n_{it}$.

task. The tasks are a combination of binary and multi-class learning.

**Loss function** The loss function is defined as follows:

$$L_t = -\frac{1}{m}\sum_{j=1}^{m}\sum_{i=0}^{C} a_j log(p_{i,j}),$$
$$L_T = \sum_{t=0}^{n} L_t \tag{7.5}$$

where $j$ is the current sample (or synset), $i$ is the class, $p_i$ is the prediction, $a$ is the set of annotations, $m$ the number of samples in the class, $t$ is the current task, and $n$ is the total number

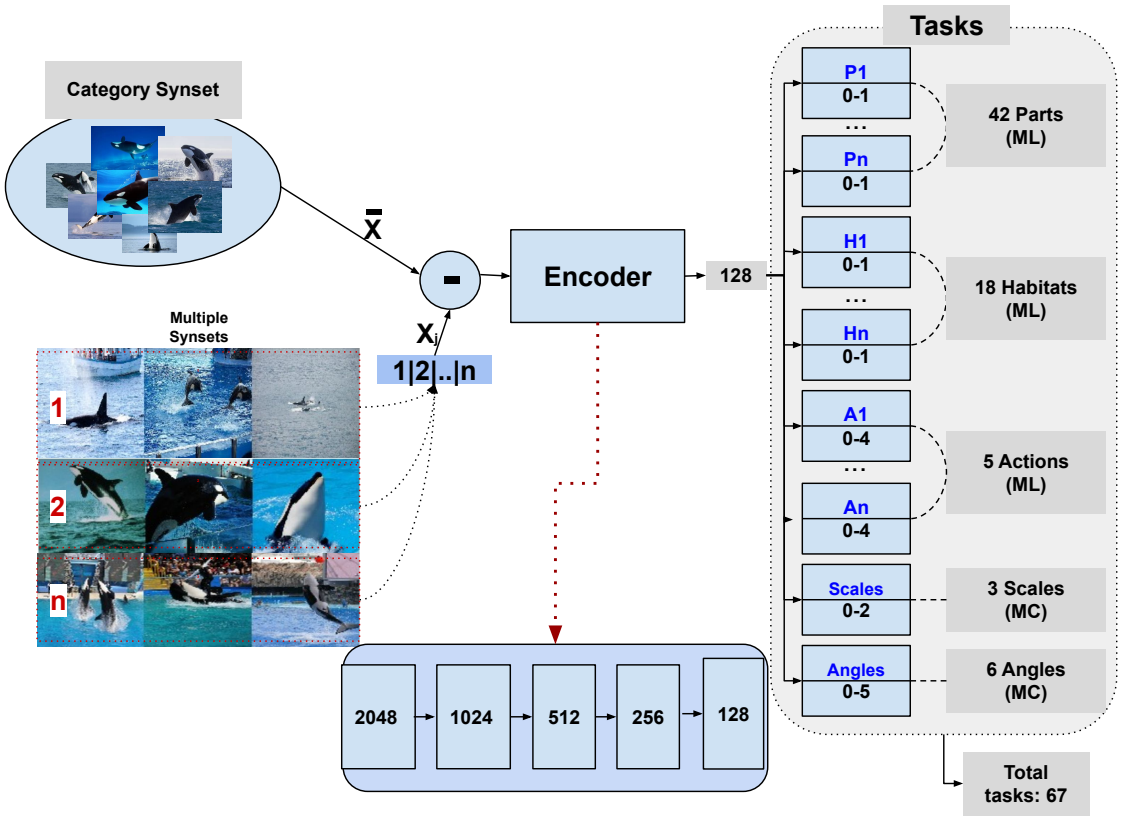



Figure 7.10: MultiAttS. The model receives residuals seen as the result of the operation $Xij - \bar{x}_i$, where $X_{ij}$ represents the synset $j$ computed from embeddings, $i$ its class, and $\bar{x}$ is the synset of the class $i$. The encoder is composed of a set of layers with the final one connecting to multiple Multi-Label (ML) or Multi-Class (MC) tasks. In turn, the tasks belong to different attribute spaces coming from different vocabularies $V$.

of tasks. $L_t$ is the loss for each multi-class task (65 binary tasks with two outputs for Parts, Habitats, Actions, and 2 multi-class tasks for Scales and Angles), and $L_T$ is the total sum of losses for all the tasks. For those attributes belonging to [Parts, Habitats] that are ground truth at the class level but that are not assigned in the initial iteration, the output does not contribute to the calculation of the loss. This way, we do not punish those attributes that are ground truth at the class level but that are not part of the initial annotations.

## 7.3 Evaluation

Here we describe the settings of our experiments. As we mentioned before, this problem is recently addressed and, as a consequence, adequate databases and methods are few in the literature. Therefore, we define the set of data proposed and required in our approach. Then we describe the initial annotation sets, the evaluation metric, the parameter tuning, and finally, the experimental results.

### 7.3.1 Data

In this section, we present the data sets used in our experiments. Tables 7.2 and 7.1 respectively describe statistics of the data sets and the notation used in this work. First, we present the description of the images source ('Data source'), followed by the description of the attribute spaces ('Vocabularies'). Then, we describe the data sets created for experimentation ('Class-level attributes', 'Validation set', 'Samples per attribute' and 'Pairing samples').

| Subset | Dimension |
|---|---|
| $X_{train}$ | 29,309 |
| $X'_{train}$ | 389 |
| $X''_{train}$ | 505 |
| $X_{Val}$ | 100 |
| C | 40 |
| V | 5 |

Table 7.2: Data dimensions. X stands for the collection of image embeddings and, $X_{train}$, $X'_{train}$, $X''_{train}$, $X_{Val}$ are different partitions created from it, C stands for the set of class labels, and V the number of different vocabularies created. For V, the proportions for each one is as follows: parts = 42, habitats = 18, actions = 5, scales = 3, angles = 6.

***Data source.*** We use as the main source the database AwA2 with the same partitions as proposed in [234]. This database contains 50 animal classes with 85 different attributes. We only use the training partition of 40 classes and divided it into different subsets explained later. Samples belonging to the same class share the same set of attributes. As with most attribute databases, the attribute information contained in it is not precise. Although attributes such as *"New World", "Old World", "Solitary", "Timid", "Smart"*, etc., give us some semantic clues, this

information is very limited and does not describe in the best way the semantic and visual concepts contained in the images. This flaw in attribute data creation is also mentioned in [314], where a poor classification performance is attributed to a human-based attribute set that is insufficiently informative. With these considerations, we propose to use this database as our main source and create databases with more specific and adequate information, that does not require an expensive creation process. We describe them next.

**Vocabularies.** We hereby analyze the different types of vocabulary $V$ that better describe the content of the target set of images.

- Parts. First, the most logical type of vocabulary concerns the animal parts. There are images in which the emphasis is on the head, while others are on the back part of the animal. Therefore, we collect a vocabulary and filter the most relevant ones by performing a manual visual analysis, with a final number of 42 different parts of animals.
- Habitats. The second feature that the images exhibit the most is the type of habitats. We collected 18 different types of habitats for animals.
- Actions. We can also describe the content by assigning an action to each image. We selected 5 actions that are represented in most parts of the images. These are *"Standing", "Resting", "Moving", "Eating" and "Swimming"*.
- Photography vocabulary: the last two types of vocabulary that describe the content regards the type of shots. For this, we selected two vocabularies that determine the scale or angle of the shot:
  - Scale shots. The shot size determines how large is the visible area within the frame. Among these, the distance between the camera and the subject varies. In general, there exist around 6 different types of shots regarding the scale. These shots are measured in general regarding human shots, and therefore, are easy to measure. When working with different types of images, this discrimination is not trivial. Therefore, among all the 6, we decide to use only 3 and classify them as follows: *Close up shot* (shows a character usually cut off across at some part of the body), *Wide Shot* (shows an entire character from head to toe) and finally, *Extreme Wide Shot* (includes broad

views of the surroundings around the character, distance, and geographic location).

- Angle shots. Another set of shots are identified by their camera angles. There is a large set of camera angles such as horizontal, vertical, aerial, etc., we selected only 6 to include all variations as follows: frontal or eye level, three-quarter front, profile, three-quarter rear, rear(r), high angle of far.

**Class-level attributes.** After we fixed the type and content of each attribute space, only for the attribute space *"Animal Parts"* and *"Habitats"*, we can create an assignment for each element in the set of classes $C$. Then, we manually annotated each class with a subset of these two attribute spaces. For the other attribute spaces *"Actions"*, *"Scale shots"* and *"Angle Shots"*, there is no possibility to assign specific attributes to each class, as these are quite generic and we can find examples in each class representing each one of these attributes. Therefore, we assume all attributes in these two spaces are ground truth attributes for all the classes.

**Validation set ($X_{Val}$).** As we do not have access to ground truth data, and to evaluate the performance of our model, we created a small validation set of 100 samples with manually annotated attributes for each type of vocabulary. To select the most representative 100 samples, we used the K-means method over all samples in $X_{train}$ and selected the sample closer to each center as the reference sample $j$. We then proceeded to annotate each one of them by assigning a subset of the most visible parts and habitats contained in the synset, one of the most representative actions (if there are more than one), one scale, and one angle attribute. We excluded this set of samples from all the explorations and training and leave it only for evaluation purposes.

**Samples per attribute ($X'_{train}$).** To improve the initial set of attributes, we annotated up to 5 samples for each attribute in $A$. This is to explore the semi-supervised training strategy when very few annotated samples are used to train the model at the first iteration (few-shot learning). We called this set $X'_{train}$ (see examples of the manually annotated images in Figure 7.11).

**Pairing samples ($X''_{train}$).** We randomly selected 100 samples from the set $X_{train}$ and for each one, we applied the process described in Section 7.2.2 – C.

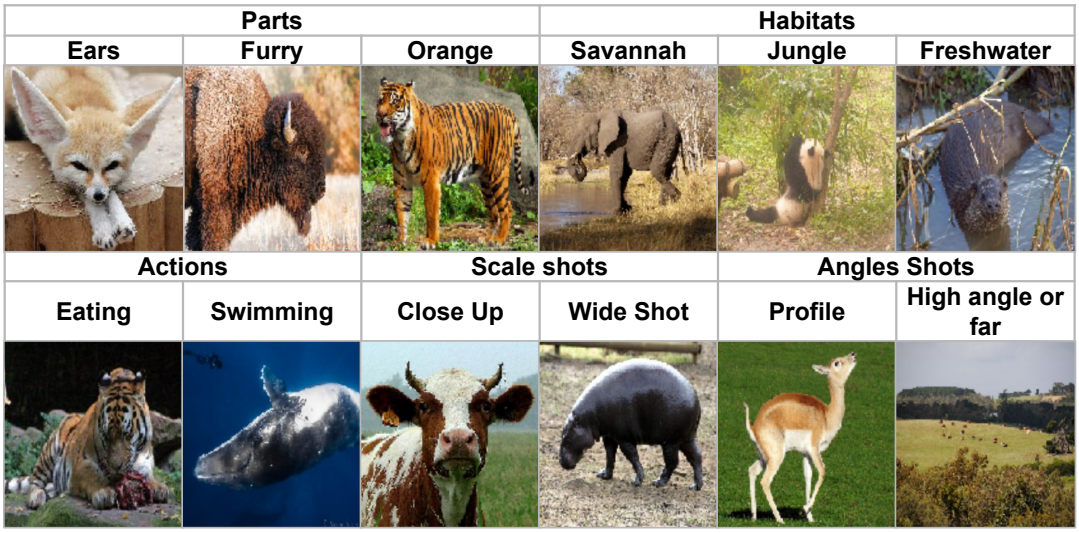



Figure 7.11: Example of images manually annotated from attributes from all attribute spaces.

### 7.3.2 Initial annotation sets

Here, we present the details for the initial set of annotations created based on the strategies presented in Section 7.2.2. We proposed 6 different initialization strategies whose objective is to evaluate the impact of improving the initial set of annotations and taking advantage of all the available information. These strategies comprise completely random initialization, followed by the inclusion of information at the class level, to finally more elaborate strategies. These last ones include weak supervision (a small number of samples annotated at the attribute level) and the usage of correlations between samples that share attribute data at the class level.

- **Init1.** In this strategy we simply choose a random subset of attributes completely class unaware for the set $X_{train}$. We fixed $p$ (following the metrics in Equation 7.3) for each attribute space as follows: parts = 4, habitats = 2, and for actions, scales and angles = 1. Those numbers are based on statistics of the set $X_{Val}$.

- **Init2.** This strategy is explained in Section 7.2.2 – A, where we simply choose a random subset of attributes belonging to the class for the set $X_{train}$. We fixed $p$ as in the Init1 setting.

**Using annotated samples per attributes.** In this setup, we use the set $X'_{train}$, for which we have three settings in which we gradually increment the number of annotations and samples to

use in training (Init3, Init4, and Init5):

- **Init3.** This strategy uses only the set $X'_{train}$ of manually annotated attributes (as explained in Section 7.2.2 – B).

- **Init4.** Complementing the Init3 setting with the number of attributes of the set $X'_{train}$ per attribute space with partial random attributes as in Equation 7.3. For example, if the sample is annotated for the attribute space *Parts* with only one attribute (e.g., "Head"), we could assign another 3 random attributes.

- **Init5.** Complementing the Init4 setting by completing the annotations for the remaining samples in $X_{train}$ as established in Equation 7.3.

- **Init6.** Finally, in this setting, we use the information regarding the attribute data at the level of classes (see Section 7.2.2, *Pairing samples with same class-level attribute*). For this initialization, we make use of Equation 7.4 to create the initial set of annotations for the set $X''_{train}$ and, we completed with random annotations for all the remaining samples in $X_{train}$ as in the Init1 setting.

### 7.3.3 Performance metric

To evaluate the performance of our proposed strategies, we define a metric that allows us to evaluate whether the top attributes predicted by the model are included in the set of ground truth attributes. We can compare this metric to the traditional recall as it considers the first K values predicted. This metric does not punish the position of the retrieved attribute if it is included in the top subset. For example, if the ground truth attributes are the set [tail, stripes, horns] and the first 4 attributes predicted are [horns, tail, fin, paws], then the relevance of attributes [horns, tail] should be the same as they both belong to the original set of ground truth attributes. In other words, the order does not have a big impact, and what matters most is to evaluate whether the ground truth attributes are in the top predicted by the model. In the following, we will refer to our customized recall metric as "c_recall". The formal formulation of the metric followed by an

example is provided next.

$$perf = \frac{1}{m}\sum_{j=0}^{100}\frac{1}{5}(\frac{n(GT_{jp} \cap P_{jp}^{K_p})}{n(GT_{jp})} + \frac{n(GT_{jh} \cap P_{jh}^{K_h})}{n(GT_{jh})} + \frac{n(GT_{ja} \cap P_{ja}^{K_a})}{n(GT_{ja})} + v_{sc} + v_{an})$$
$$\begin{cases} K_p, K_h, K_a = n(GT_{jp}), n(GT_{jh}), n(GT_{ja}), & \text{if } topK = n \\ K_p, K_h, K_a = topK, & \text{otherwise} \\ v_{sc} = (GT_{j\_sc} = P_{j\_sc}), v_{an} = (GT_{j\_an} = P_{j\_an}) \end{cases} \tag{7.6}$$

where j=100 is the number of validation samples, $GT$ is the set of ground truth annotations, $P$ is the set of predicted attributes for the synset $j$ for each attribute space [parts: $P_{jp}$, habitats: $P_{jh}$, actions: $P_{ja}$], $n$ represents the dimension of a subset, K represents the number of predictions to take into account for each attribute space $p, h, a$ (with *p:parts, h:habitats, a:actions*), $sc$ represents the performance of the attribute space *scales* and $an$ the one of *angles*. We report results for the top K with K=n(GT) and K=10 along with their average.

Let us explain the previous equation with an example. Assuming a sample $s1$ with the following ground truth attributes: parts [horns, tail, fur], habitats [savannah, forest], actions [eating], scales [close-up], and angles [profile], the variables will be: $n(GT_{jp})$ = 3, $n(GT_{jh}) = 2$, and $n(GT_{ja}) = 2$, and assume the model predicted the following attributes: parts [tail, leg, fur, head, horns], habitats [forest, river], actions [resting, swimming, eating], scales [close-up], and angles [frontal]. Then, the calculation of the score for the first case (with $K = n$) is as follows:

$$perf_{s1} = \tfrac{1}{5}(\tfrac{n([horns,tail,fur] \cap [tail,leg,fur])}{3} + \tfrac{n([savannah,forest] \cap [forest,river])}{2} + \tfrac{n([eating] \cap [resting])}{1}$$
$$+(closeup == closeup) + (profile == frontal))$$
$$perf_{s1} = \tfrac{1}{5}(\tfrac{n([tail,fur])}{3} + \tfrac{n([forest])}{2} + \tfrac{n([\_])}{1} + (true) + (false))$$
$$perf_{s1} = \tfrac{1}{5}(\tfrac{2}{3} + \tfrac{1}{2} + \tfrac{0}{1} + 1 + 0)$$
$$perf_{s1} = 0.43$$
$$perf_{s1} = 43\% \tag{7.7}$$

For the second case, let us assume K = 5, then the calculation of the score is as follows:

$$
\begin{aligned}
perf_{s1} = \tfrac{1}{5}\Big(&\tfrac{n([horns,tail,fur]\cap[tail,leg,fur,head,horns])}{3} + \tfrac{n([savannah,forest]\cap[forest,river])}{2} \\
&+ \tfrac{n([eating]\cap[resting,swimming,eating])}{1} \\
&+(closeup == closeup) + (profile == frontal)\Big) \\
perf_{s1} = \tfrac{1}{5}\Big(&\tfrac{n([tail,fur,horns])}{3} + \tfrac{n([forest])}{2} + \tfrac{n([eating])}{1} + (true) + (false)\Big) \\
perf_{s1} = \tfrac{1}{5}&(\tfrac{3}{3} + \tfrac{1}{2} + \tfrac{1}{1} + 1 + 0) \\
perf_{s1} &= 0.7 \\
perf_{s1} &= 70\%
\end{aligned}
\tag{7.8}
$$

### 7.3.4 Parameter setting

We implemented our approach using PyTorch, and explore the model architecture concerning the dimension of the batch size from 2 to 32 (with a final batch size of 4), the learning rates 0,0001, 0,001, and 0,01 (with a final value of 0,001), and the optimizer RMSPROP and SGD (with the final being SGD). Also, to help to balance the samples in each mini-batch, we make use of a random sampler with equal weights to all samples and with replacement. We obtained an increase of $\approx 4\%$ in the performance with this option. Concerning the dimension of the neighborhood $\mathcal{N}$ to compute each synset, we explored the impact of using different sizes of neighborhoods to calculate the synset between 1 to 15. The results showed that the model learns better to discriminate among different synsets with $\mathcal{N}$ = 10 (see Figure 7.12). Therefore, we chose this parameter for all experiments. Also, when using $\mathcal{N}$ = 1, i.e., only one image, there is not enough information to discriminate among different attributes. For example, in Figure 7.20, Page 172, the attribute "tongue" is present in a variety of positions of the class "Dog" which helps to discriminate the visual concept among different visual variations.

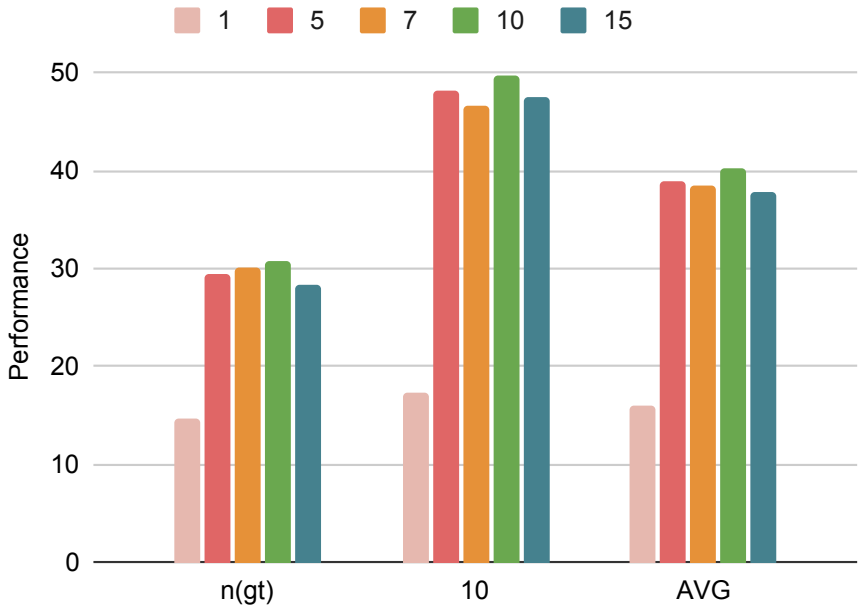



Figure 7.12: Performance of the model when modifying the number of samples in the neighborhood $\mathcal{N}$ to compute the input synset. Colors refer to the change in the parameter $\mathcal{N}$ in [1, 5, 7, 10, 15].

Concerning the input data passed to the model, we explored two options and evaluate the performance in each one. We explored two alternatives inputs to the model: $synset_j - \bar{x}_i$ and $synset_j$. Let us recall that $synset_j$ represents a set of images from the class $i$ containing the same visual concept, while $\bar{x}_i$ represents the synset of the class $i$ (the average of all the image features belonging to class $i$). We obtained an improvement of $2,96$ in the c_recall for the evaluation of the performance when using K=n(GT) and, $3,25$ in c_recall when using K=10. With these results, we can validate our hypothesis that residual representations allow us to separate class-specific features from other features that encode other aspects of the image. Table 7.3 presents the results for experiments concerning the parameter exploration with the batch size (BS), Learning rate (LR), optimizer, with Weighted Random Sampler (WRS), and testing the impact of the inputs.

| **Batch size** | **K = n(GT)** | **K = 10** |
|---|---|---|
| 2 | 20,13 | 22,75 |
| 4 | **29,52** | **45,69** |
| 8 | 28,56 | 43,90 |
| 16 | 26,42 | 41,86 |
| 32 | 27,41 | 41,69 |
| **Learning Rate** | | |
| 0,0001 | 27,88 | 41,41 |
| 0,001 | **29,52** | **45,69** |
| 0,01 | does not converge | |
| **Optimizer** | | |
| RMSPROP | 26,03 | 40,99 |
| SGD | **29,52** | **45,69** |
| **Others** | | |
| With WRS | **30,70** | **49,65** |
| Passing $synset_j$ | 27,74 | 46,40 |
| Passing $synset_j$-$\bar{x}_i$ | **30,70** | **49,65** |

Table 7.3: Parameter tuning.

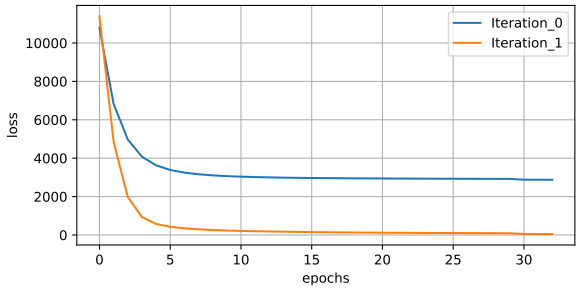
### 7.3.5 Experimental Results

In this section, we present the experiments proposed to evaluate our framework. We first provide learning curves for training loss and performance metrics. Then, we present the evaluation concerning the impact of different initialization strategies, followed by an analysis of the results for each attribute space, the comparison of the re-annotation strategy when performing multiple iterations, and finally a visual analysis along with several examples of our predictions.

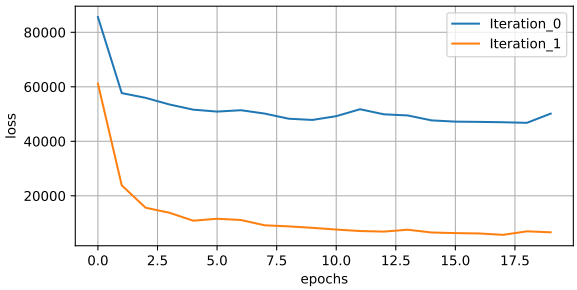
**Model learning.**

Figure 7.13 presents the curves of training loss for two iterations. The first graph shows the curve for training loss for the best setting found in Table 7.3 for the given batch size, learning rate, optimizer, and sampling (row 'With WRS'). The model converges with only less than 10 epochs and without over-fitting (no supervised data). Also, in the second iteration, the model converges faster while the loss achieves a lower value. The second graphic shows the impact on the loss function when the synset is passed instead of the residual (row: *Passing* $synset_j$). Although there is convergence in both iterations, the loss increases tenfold. This result shows

that feeding the model with the residual indeed has a positive impact on the learning process.




A) Fast model convergence under the setting: (Batch size = 45.69), (Learning rate = 0.001), (Optimizer = SGD), and using a weighted random sampler (WRS).




B) *Passing* $synset_j$ instead of residual.

Figure 7.13: Training loss for two iterations for A) the best parameter setting, B) when passing the $synset_j$ instead of residual.

In Figure 7.14, we show the standard performance metrics used in classification for attribute learning at the class level when they are well defined. However, the problem we addressed in our work is more challenging. It requires new learning strategies at the level of the training process as well as in how to leverage the available data to improve the initial set of annotations.

**Comparison of the performance of the proposed initialization strategies.**

Figure 7.15, Page 168 presents the results for each initialization strategy presented in Section 7.3.2. First, notice that the performance tends to increase in all strategies when comparing a K=n(GT) to K=10. Second, we can observe the impact of going fully random (Init1) compared to the case when only some of the class-attribute data is included in the random initialization (Init2). For initialization strategies (Init3, Init4, and Init5), the tendency is to increase when K=10. This shows the impact of improving the initial annotations with ground truth annotations. This result also shows that by having very few annotated samples ($\approx 1\%$ of the training data), the performance improves dramatically in comparison with a fully random setting (Init1 compared to Init3-Init5). The initialization strategy that has the biggest impact on performance (Init6) demonstrates the benefits of creating correlations among the samples based on attribute information extracted at the class level. This is indeed useful to propagate labels in different

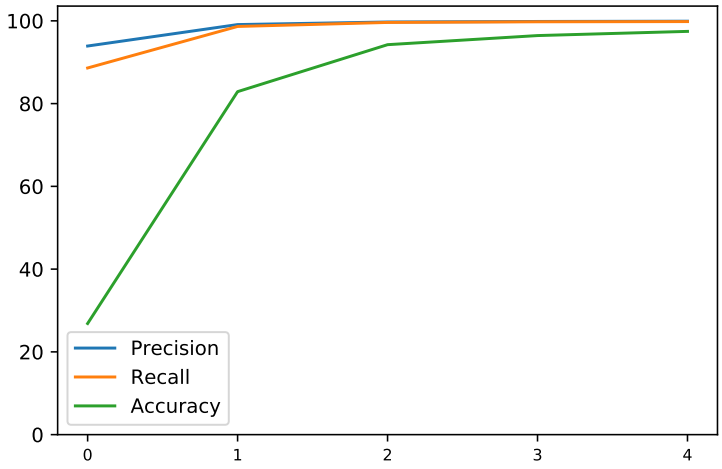



Figure 7.14: Standard performance metrics at the attribute-class level for the database aWa2 with the new set of attribute spaces defined: Parts & Habitats.

classes that are known to share common attributes. Figure 7.16, Page 168 presents the training loss for two iterations, for the best initialization strategy ('Init6') according to the average performance group displayed in Figure 7.15. Although the loss curves are not entirely smooth, they converge at $\approx 30$ epochs. The loss value reached is high. However, let us recall that during the training phase, the model has extremely weak supervision. Therefore, high loss values are expected.

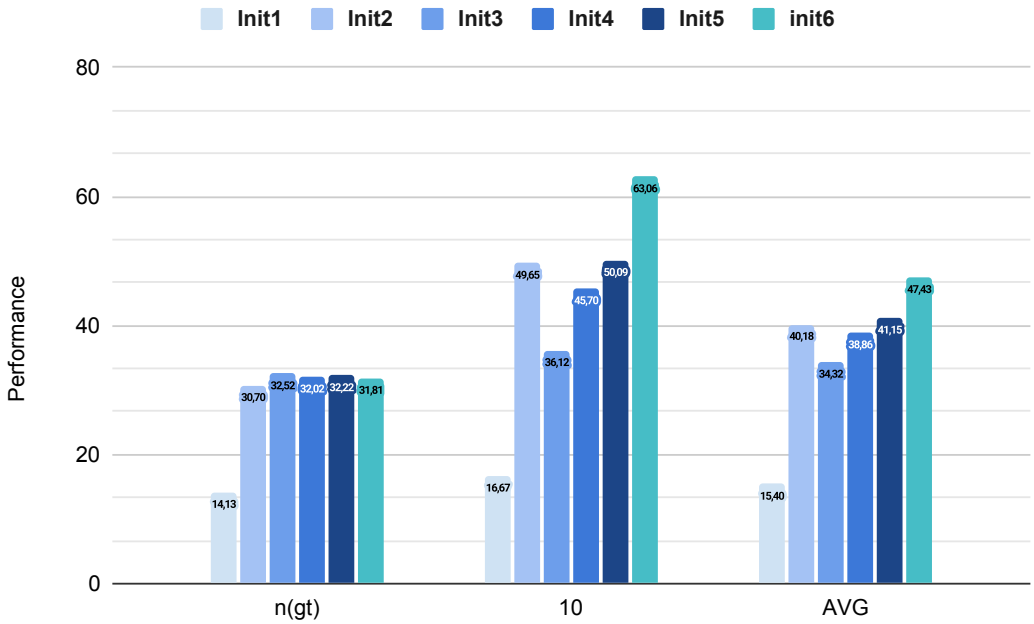



Figure 7.15: Comparison of the performance vs the initialization Strategies presented in Section 7.2.2. The results are calculated by using the top-K with K=n(GT), K=10 and their average. The initialization settings explored are as follows: Init1) Fully random; Init2) Using strategy A (partial random), Init3) Using strategy B (subset $X'_{train}$), Init4) Using a combination of strategies A and B for the set $X'_{train}$, Init5) Using a combination of strategies A and B for the set $X_{train}$, Init6) Using strategy C.

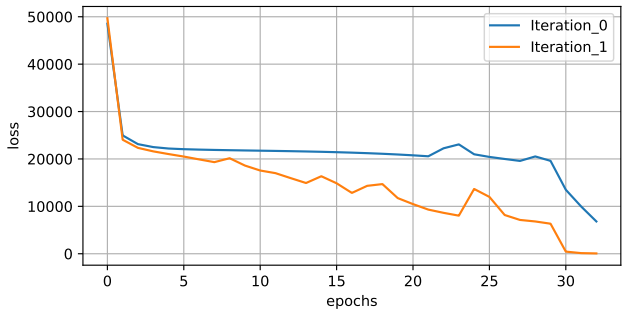



Figure 7.16: Training loss curves for two iterations for the best initialization strategy (Init6) with convergence at $\approx$ 30 epochs.

We also present an analysis of setting Init4 when we gradually augment the number of annotated samples per attribute from 1 to 5 (see Figure 7.17, Page 169). We can see that the performance gradually increases for the first 4 results. As we also increase the number of random attributes

per sample, the noisy attributes are bigger and therefore, the performance for the last experiment (5 samples per attribute) does not show an improvement. Let us recall that the model is trained only with the manually annotated samples (subset $X'_{train}$ that contains only $389$ samples vs original training set with $29,309$ samples).

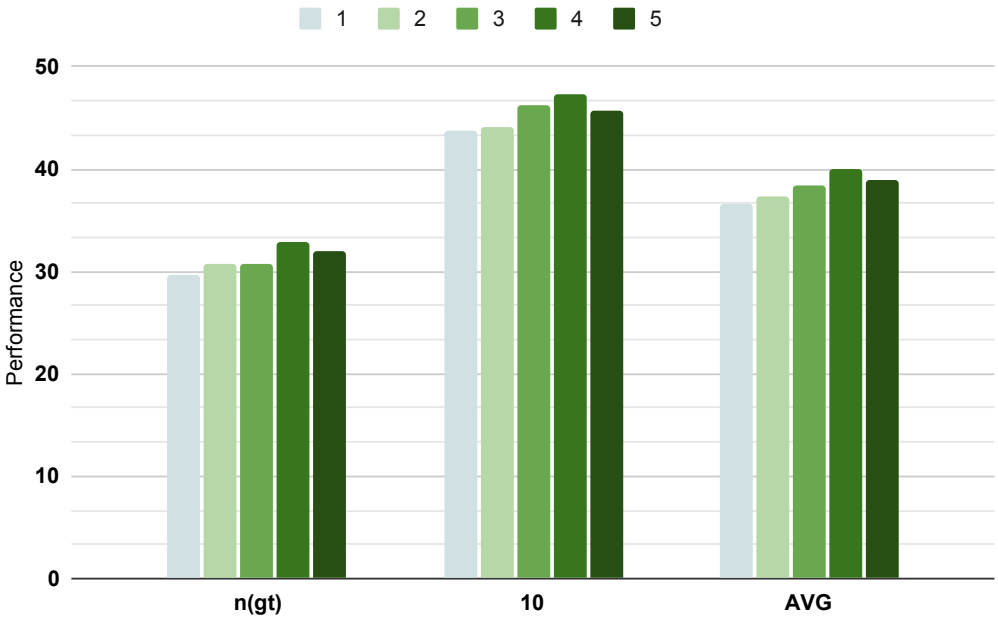



Figure 7.17: Number of manually annotated samples per attribute vs performance. We increase the number of samples from 1 to 5. The results are calculated by using the tops with K=n(GT), K=10 and the average of both Ks.

**Exploration of the performance per attribute space.** Analyzing the performance per attribute space helps us to know what attribute space is harder to learn and, for future work, to include an attention mechanism focused on it. Figure 7.18 presents the performance for the initialization settings with supervision at some degree Init2-Init6 at the level of each attribute space considered. The easiest one to learn is the attribute space describing *"Habitats"*, and the most difficult one being *"angles"*. We can notice that in the strategy Init2, the performance is mostly dependent on *Habitats*. Instead, with the best strategy Init6, each one of the performances coming from the main attribute spaces contributes in a significant way to the total performance. *Animal parts* has the most stable performance among all settings, followed by *Actions*, *Scales*, *Angles* and finally, *Habitats*.

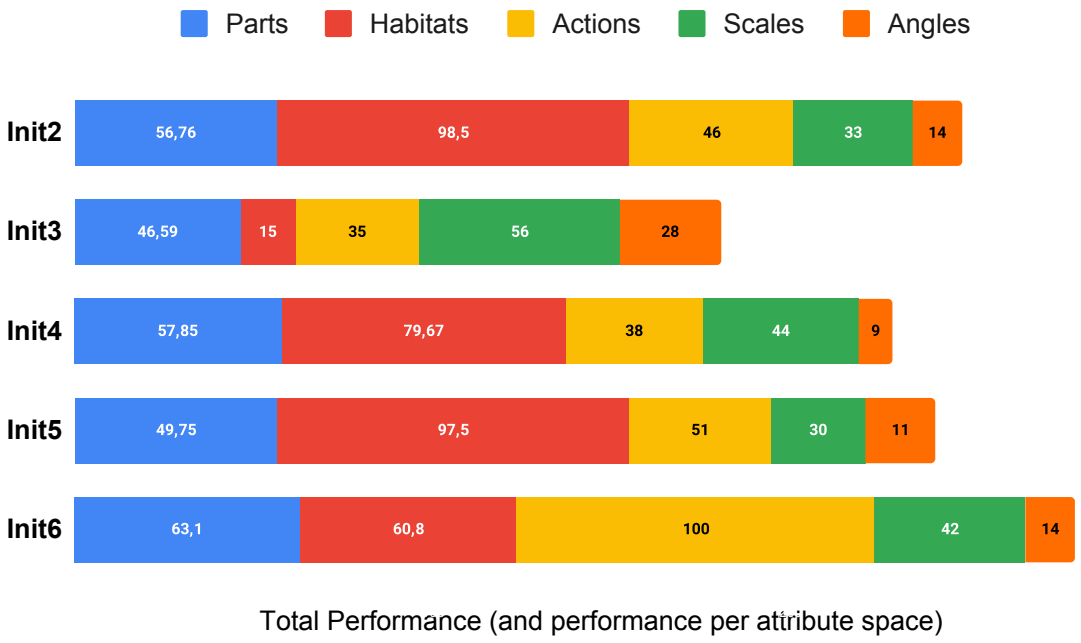



Figure 7.18: Performance per attribute space in each setting.

**Comparisons of the performance of the re-annotation strategy vs a number of iterations.** Our last analysis explores the performance when using the two different re-annotation strategies described in Section 7.2.3. Figure 7.19 presents the results for multiple iterations. We can notice that when we do not filter all the positive predictions and re-train the model with all the new annotations, the performance tends to converge at the same value (annotation strategy A1, Figure 7.7, Page 153). In the second case (annotation strategy A2, Figure 7.8, Page 154), we decide not to believe in all the positive predictions by filtering those smaller than the threshold $thr_a$. We observe that the performance decreases at each iteration, indicating that we are removing key annotations.

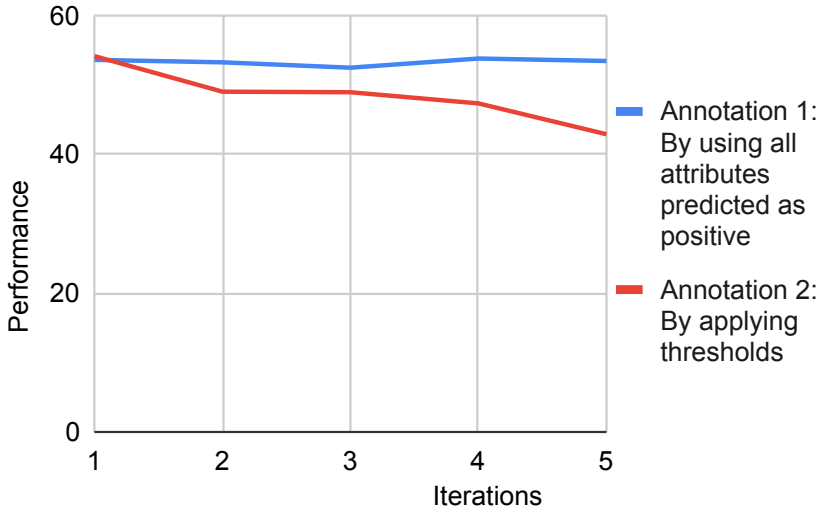



Figure 7.19: Comparison of the performance vs the number of iterations for the re-annotation strategies described in Section 7.2.3.

**Qualitative results.** Figure 7.20 presents an example of a synset with 5 random images from the total neighborhood of 10. The first column lists the manually annotated or ground truth attributes, while the second column shows the random annotations to train with. The last column provides the predicted annotations. Regarding the attribute spaces *parts* and *habitats*, it is highly subjective to determine how many attributes best define the visual concept. We can also notice, as we mentioned before, that the most difficult attribute space to learn is the *angles*. As the subset of annotations defined previously can be subjective and can unfairly damage the performance, we plan to explore this aspect by annotating our validation set $X_{Val}$ with more annotators. Figure 7.21, Page 173 presents additional examples of predictions for different samples for each attribute space.

| B |  | | |
|---|---|---|---|
| | **Manually annotated** | **Random** | **Predicted** |
| **Parts** | Ears, Eyes, Head, Tongue, Mouth, Nose | Eyes, Furry, Patches, Tail | Black,Tongue, Head, Gray, Brown, Mouth |
| **Habitats** | Domestic | Domestic, Farm | Domestic, Farm |
| **Actions** | Resting | Standing, Eating | Moving, Resting |
| **Scales** | Close-up Shot | Close-up shot | Close-up shot |
| **Angles** | Frontal \| Eye Level | Rear | High Angle (Or far) |

Figure 7.20: Attribute comparison for 5 random images belonging to a $synset_j$ with respect to the manual annotations taken as ground truth, the initial random annotations and the predicted for the trained model.

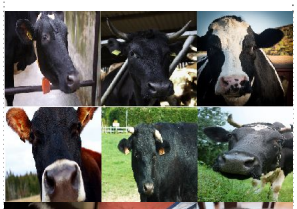
| | Predictions | | | | |
|---|---|---|---|---|---|
| | **Parts** | **Habitats** | **Actions** | **Scales** | **Angles** |
|  | nose<br>patches<br>black<br>white<br>brown<br>nostrils | farm<br>domestic | resting<br>eating | wide shot | high angle (or far) |
|  | tail<br>furry<br>nose<br>belly<br>tongue<br>head | farm | standing<br>resting | close-up shot | three-quarter rear |
|  | black<br>tongue<br>eyes<br>head<br>paws<br>gray | forest<br>tree | resting<br>moving | wide shot | three-quarter rear |
|  | horns<br>tail<br>nose<br>furry | savannah<br>farm | eating<br>standing | extreme wide shot | rear |
|  | furry<br>legs<br>ears<br>claws<br>neck<br>nose | river<br>forest | moving<br>resting | wide shot | profile |
|  | ears<br>tail<br>furry<br>belly<br>eyes<br>head | jungle<br>tree | moving<br>eating | wide shot | high angle (or far) |

Figure 7.21: Examples of predictions for each attribute space. Expected predicted annotations for the model on average include the group of images in lines 1, 2, 5, and 6, while wrong average annotations include the 3 and 5. Some attributes predicted for each vocabulary may be subjective even for humans.

## 7.4 Discussion

In this chapter, we addressed the problem of insufficiently labeled data. We have presented a semi-supervised learning approach that exploits class-level relationships to assign attribute

annotations to synsets. In the zero-shot learning strategy which is the closer learning paradigm to ours, attributes are annotated by the class which reduces the labeling cost significantly. However, the attribute-level descriptions used for ZSL are usually obtained from abstract vocabularies that make sense from the semantic perspective of classes and not for individual images. For example, sometimes these attributes were not visual properties of objects or they were not visible in all the samples of the class as we have shown with the AwA2 database. We also addressed another problem occurring under the ZSL strategy: the fact that testing instances are assumed to come from the unseen classes only. Indeed, in our approach, test instances can come from all classes because the emphasis is on propagating attributes to samples regardless of their class. And, unlike the state of the art, we assume that even at training, we do not have access to annotations for all the heterogeneous attribute spaces.

By extensive explorations over multiple initialization strategies, we found that the performance improved in a significant way even when there was no or few labeled data (up to 63% of precision for the validation set using less than 1% of the annotated training images). We found the strategy of pairing sample annotations via attribute-class data to be extremely helpful. Additionally, we proposed small but sufficient data sets created for the collection AwA2 to carry on a complete experimental evaluation.

# CHAPTER 8

# Conclusions

## 8.1 Overview

In this thesis, we presented the problem of multimodal learning. We limited our explorations to two of the most frequent and main modalities studied in the state of the art, the visual and the textual modality. We approached it by proposing frameworks that find a common semantic space for both modalities using deep learning as the main tool. As we presented, deep learning has shown incredible success in solving complex tasks (see Chapter 3). We evaluated our strategies in several well-studied as well as recent and relevant state-of-the-art applications on various widely used databases. In our evaluations, we found many data-related issues, such as noise or incorrectly annotated data, that drastically damage model performance. We also found that the process of fine-tuning deep learning models in conjunction with parameter setting is still very expensive. This creates a higher demand for computational resources that is not always possible. Another factor is the problem related to the comparison with the state of the art that requires reproducibility of the methods and that is not yet addressed. In this chapter, we present

a summary of the contributions of this dissertation as well as future lines of research.

## 8.2 Summary of Contributions

In this thesis, we explored the problem of multimodal learning for images and text from several perspectives. Our proposed frameworks and strategies resulted in 4 published papers and a final paper currently under preparation for submission. Next, we present a summary of the specific contributions per chapter concerning our proposals.

**Contributions of Chapter 4.** We proposed the re-interpretation of the VQA architecture for cross-modal retrieval, followed by an extensive experimental evaluation of its capabilities in this problem. Our work is the first to evaluate all cross-modal and uni-modal tasks with a single model trained once for all tasks (without task-specific fine-tuning), reaching highly competitive results as well as state-of-the-art results, even though it is compared to methods that are specific and/or optimized for specific tasks. This shows that a unifying model for multimodal retrieval is possible. The simplicity of our approach serves as a baseline for future research and can inspire extensions to push performance even further. We expect future researchers to take advantage of this well-established architecture to reach higher performance in their models. This contribution resulted in a journal publication entitled *"Deep Multimodal learning for Cross-Modal Retrieval: one model for all tasks"* [66] along with the code required to reproduce our experiments (available at https://github.com/lvbeltranb/DME).

**Contributions of Chapter 5.** We proposed an architecture with different components involved in an ST-VQA system: a visual, a textual, and a multimodal network in charge of pre-processing the different modalities and learning the features required for the target task. In the first part of our explorations, we made an emphasis on the impact of the representation of the critical data: questions and answers. We compared context-based and context-free textual embedding models for the questions. For the answers, we made use of an n-gram configuration, which gives the model a level of flexibility. To the best of our knowledge, we were the first to report results in this task using these embedding models. In the second part of our work, we proposed significant improvements to the framework with the addition of helper components that improved the

performance of the model. We evaluated the relevance of the dimension of the answer space for the case of a fixed set of words and the case when the copy module is used as the main strategy for the OOV problem. We exposed drawbacks related to the copy module which is the state-of-the-art solution, as well as the proposal of using a second metric to compute the scores for the dynamic spaces so that the copy module can take advantage of texts not 100% recognized by the OCR system. We evaluated the performance of including additional data to train the system in the form of a complementary network representing embeddings from textual and visual data. We presented the results of several ablative studies to validate the relevance of the components proposed in our framework. Our results serve as baselines for future research. This work resulted in two published papers entitled *"Semantic Text Recognition via Visual Question Answering"* [67] and *"An extended evaluation of the impact of different modules in ST-VQA systems"* [68].

**Contributions of Chapter 6.** In this chapter, we analyzed the principal problem of imbalance in attribute databases. We studied the efficacy of deep learning models when the data is available but in such conditions. To evaluate our strategies, we considered the application of attribute learning. We validated the effectiveness of classical imbalance learning approaches applied to straightforward multi-attribute approaches. These systems are seen as black-box models, that could be used by non-computer science experts. We presented a study that demonstrates that most bad performance problems are due to the data itself. We also exposed different problems derived from imbalanced databases in the context of very fine-grained multi-attribute learning, some of which are ignored in the state of the art, together with proposals to address them. After performing large experimentation of imbalance learning strategies, including mainly "sampling" and "cost-sensitive learning strategies", results indicated that these strategies are not adapted to multi-task or multi-label problems. These are adequate for the general problem of classification, under some ideal conditions present in the database. Therefore, we proposed strategies that aim to solve these problems. This work resulted in a published paper entitled *"Multi-Attribute Learning with Highly Imbalanced Data"* [70].

**Contributions of Chapter 7.** In this chapter, we addressed the problem of insufficiently labeled data. We formulated the problem of assigning attributes to images from class-attribute

relationships as a semi-supervised learning problem using no or few manual annotations. We explored several strategies to improve the initial set of annotations with none or less than 1% of manually annotated samples. Our formulation allowed us to incorporate attributes from multiple vocabularies, which can have mutually exclusive labels or multi-label tags. These attribute spaces allow for richer annotations and are modeled using multi-task learning. We created small but sufficient data sets to carry on a complete experimental evaluation, with image-level annotations obtained only for the test set from the database AwA2. We also defined five different attribute spaces (vocabularies) for 40 training classes. Our methodology can facilitate the propagation of attributes to any image set organized in classes. Our results showed that by using the information at the level of class-attribute relationships, we can learn models that recognize the correct set of attributes for test images. We also found that the initialization strategy for semi-supervised learning is the most important step for improving performance and assigning attributes. This final contribution is currently under preparation for submission.

## 8.3 Future lines of research

Taking into account the lessons learned from this work and the improvements due to recent models that are actively being developed in the research community, we list in the following paragraphs what we identified as key topics for future research in the field of multimodal machine learning.

**Cross-modal retrieval.** Graphical neural networks, probabilistic approaches, and transformers are gaining importance in many multimodal tasks. For cross-modal retrieval, new developments are based on graphs that do not project the original feature into an aligned representation space but adopt a cross-modal graph to link different modalities [315]. On the other hand, probabilistic frameworks where samples from the different modalities are represented as probabilistic distributions in the common embedding space are also innovating in this field [316]. Another future line of research concerns domain adaptation, which is retrieval in databases that contains data in multiple types of domains. For example, databases that contain Wikipedia articles composed of natural images and more general information, but also medical articles with more

specific information [317]. Or even much more specific databases such as those in the art domain that are only accessible through tags or labels but that have richer descriptions only for some images (see Figure 8.1 with examples extracted from the ROMANE database). Models can be trained in a more generic database and fine-tuned for more specific databases.


Figure 8.1: Example results from the ROMANE database. It is a French database that contains hundreds of images of documents related to architecture and monumental decorations from different periods. However, the search is only possible through textual tags. The query tags in the example images are "France" and "église", but more complete descriptions can be found in some images. Besides, visual search to retrieve relevant and similar architectural patterns with textual data containing specific descriptions can be interesting. Samples extracted from http://base-romane.fr/accueil2.aspx (accessed March, 2021).

**ST-VQA.** The problem of out-of-vocabulary in the answer space has not yet been successfully addressed. Although some works have explored copy mechanisms and represent a partial solution, these methods are based on a dynamic allocation of spaces that represent different words, and that can be interpreted as a random mechanism. Therefore, more robust strategies are required [318, 319].

**Attribute learning.** In existing works on zero-shot learning that are becoming more popular in this domain, the training data usually consists of predetermined labeled instances in the same feature space as the testing instances and of the same semantic type of the testing instances (for example, both are images of animals). It is desirable to explore other ways of selecting training data for robustness and generalization of models. These methods should include a study of the characteristics of input data to select the most appropriate feature extractions methods.

Also, they should include an optimal selection of training data that involves from which set it is extracted or if dynamically annotated data should also be used. Finally, the combination of different learning paradigms that take advantage of different capabilities is also required. For example, shot-learning and reinforcement learning, among others [276].

**Transformer methods.** Models based on transformers are revolutionizing language and vision and are starting to be used in multimodal learning tasks including retrieval [320] and ST-VQA [216]. These models took the concept of "attention" to another level because they help to discriminate between relevant and non-relevant information, and are showing performance increases in various applications. Recent works have shown how multimodal attention mechanisms (transformers) can outperform deeper models with a modality-specific attention mechanism [321].

**Improvements regarding the data.** Tackling noisy and restricted annotations is required to obtain good performances. A large amount of multi-modal data is created by people on various websites such as YouTube, Facebook, and Flickr, etc. This data from the web is not properly organized and annotated, and proper and exact labeling is required. Besides, extensive experimentation is required on these types of open databases to better show the efficiency of the proposed techniques. Another issue is the need to measure the impact of data sparsity on the classification performance for databases with class imbalance. This is necessary because when data is sparse, there is a large number of attribute values that are equal to zero. Reducing the dimensionality of these databases before constructing classifiers by machine learning algorithms becomes essential especially for databases that suffer from imbalanced data [322].

**Diversity in data composition and large-scale databases.** Most existing benchmark databases are old and consist only of images and textual data. Future works should include a more diverse set of modalities to test and validate the proposed algorithms such as video and audio or should extend the research to other languages than English in applications including textual data. This problem also refers to the lack of databases in different domains, especially in the medical one [106].

**Semi-supervised learning.** Making full use of the supervised information can improve the retrieval performance, but since the supervised information in the real world is often missing or incorrect, the question of how to ensure a certain performance level will be an active research direction. Developments of semi-supervised techniques are thus required in all multimodal tasks, especially for databases in domains such as the medical one in which there exists almost no annotated data because of the required expert knowledge [323].

**General perspectives.** More efficient ways to perform a strong parameter setting and model selection are required. Utilizing and combining appropriate deep neural network models is required to achieve better performance on multimodal tasks. Also, various representations can be acquired by various networks and some of them may be more adequate for the target problem. For this, we need to take into consideration that deep learning requires plenty of manual fine-tuning which represents a computational process very expensive. Another important issue is the need for reproducible and explainable methods in the state of the art. Many complications arise when reproducing methods such as incorrect parameter setting descriptions or outdated libraries. For all users, especially non-expert users who simply need to use these tools as black boxes, these difficulties in using these tools can represent great dissatisfaction. Finally, there is a need for implementing machine learning methods in big data, cloud, and IoT environments, as well as share them for further improvements.

# Publications

This thesis has led to the following publications:

**Conferences**

- Viviana Beltrán, Nicholas Journet, Mickael Coustaty, and Antoine Doucet. "Semantic text recognition via visual question answering." In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), vol. 5, pp. 97-102. IEEE, 2019.
- Viviana Beltrán, Mickaël Coustaty, Nicholas Journet, Juan C. Caicedo, and Antoine Doucet. "An extended evaluation of the impact of different modules in ST-VQA systems." In International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI), pp. 562-574. Springer, Cham, 2020.
- Viviana Beltrán, Mickaël Coustaty, Nicholas Journet, Juan C Caicedo, and AntoineDoucet. “Multi-Attribute Learning with Highly Imbalanced Data”. In 25th International Conference on Pattern Recognition (ICPR), 2020.

**Journals**

- L. Viviana Beltrán, Juan C. Caicedo, Nicholas Journet, Mickaël Coustaty, François Leceiller, and Antoine Doucet. "Deep Multimodal learning for Cross-Modal Retrieval: one

model for all tasks." Pattern Recognition Letters (2021).

# Appendices

# APPENDIX A

# Appendix

## A.1 Basic concepts

To carry on the learning process, deep neural networks (DNNs) require establishing loss, and activation functions, along with optimization and back-propagation training algorithms. Next, we present the definition for the most principal and more widely used.

### A.1.1 Loss functions

To minimize the objective function through an optimization process, we need to establish a function that is normally known as cost, loss, or error function. This function is in charge of calculating the model error what allows us to measure how far the predicted scores given by the model are away from their true values. Next, we describe the principal loss functions used in multimodal learning problems [324].

**Logistic loss: Binary Cross Entropy and Cross-Entropy Loss.** These functions are widely used for classification tasks. In binary classification, where the number of classes is two, the binary cross-entropy (BCE) can be calculated as follows:

$$BCE_{Loss} = -(y \log(p) + (1 - y) \log(1 - p)) \tag{A.1}$$

Where y is a binary indicator (0 or 1) with the true value, and p is the predicted probability.

For multi-class classification, i.e, when the number of classes n > 2, we calculate a separate loss for each class label per observation and sum the result.

$$CE_{Loss} = -\sum_{c=1}^{n} y_{o,c} \log(p_{o,c}) \tag{A.2}$$

Where $y_{o,c}$ is the true value and $p_{o,c}$ is the probability for the $i^th$ class.

**Mean squared error.** This function computes the average of squared differences between predictions $\hat{y}_i$ and the true actual values $y_i$ as follows:

$$MSE_{Loss} = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{2} \tag{A.3}$$

.

### A.1.2 Activation functions

In this section, we present the mathematical definition of the most popular nonlinear activation functions. Activation functions are mathematical equations that are applied to the output of a neural network and determine its value. They can put a neuron output on and off, or depending on a threshold or condition, or they can also normalize the output between one range such as 0 and 1, or between -1 and 1 [325].

Because the final neural network outputs are raw scores (or better known as logits $z(x)$), activation functions can also transform these logits into understandable and interpretable values. The most common solution is to assign the sample to the class with the highest raw output score. For

this, the better method is to convert these scores into probabilities. Thus, mathematical functions are applied to the output vector to convert raw scores to their probabilistic scores. The most well-known are the sigmoid and the softmax functions.

**Sigmoid function.** This function limits the output between 0 and 1. It is used in binary classification.

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{A.4}$$

**Softmax function.** The softmax function is the extended version of the sigmoid function. It is used in the multi-class classification problem, $C > 2$, and is computed as follows:

$$\mathbf{prob}(y = k|\mathbf{x}) = \frac{e^{\mathbf{z(x)}_k}}{\sum_{j=1}^{C} e^{\mathbf{z(x)}_j}},\ for\ k\ in\ \{0, 1, 2, ..., C\} \tag{A.5}$$

Where $y$ is the actual class of the sample $x$, $k$ is the class, and $C$ is the number of classes.

The resulting output vector adds up to one and can be seen as a probability vector. This helps to determine the class most likely to be associated with the input sample.

**TanH / Hyperbolic Tangent.** This function results in a zero centered output value. This makes it easier to model inputs that have strongly negative, neutral, and strongly positive values.

$$tanh\, x = \frac{sinh\, x}{cosh\, x} \tag{A.6}$$

**ReLU (Rectified Linear Unit).** This function always gives 0 for negative values.

$$f(x) = max(0, x) \tag{A.7}$$

Other activation functions include Linear Activation Function, Leaky ReLU, Parametric ReLU, Swish, etc., [325].

### A.1.3 Optimization and Back-propagation

A machine learning model requires a process of optimization, to be able to give accurate and acceptable predictions over the phenomenon of study. This process requires adjusting the parameters of the model to minimize the loss function.

Back-propagation is used in the training process of a neural network [326]. After each forward pass through the network, a backward pass follows to adjust the parameters of the models as are the weights $W$ and biases $b$ (see Figure 2.4). This is done to minimize the difference between the true expected values and the output of the model. This process is also called fine-tuning of the network and is necessary to ensure the reliability of the model and increase its generalization. Ideally, with each backward pass, the weights move towards an optimum, and therefore minimizing the loss function and obtaining the most accurate prediction. The basic steps are as follows:

1. Initialization of the model parameters. In this step, the weights W and biases b in the model are initialized. The initialization method can have an impact on the training convergence of the model. The most popular strategies include zero initialization, random initialization, and Xavier initialization [327].

2. Propagation of the inputs forward: in this step, the inputs are taken and fed in a forward direction through the whole network. Each hidden layer takes the input data, processes it, applies the activation function, and finally, passes it to the next layer.

3. Back-propagation of the error: in this step, the process of adjusting the parameters of the models is carried out. For these, optimization algorithms are required. Stochastic gradient descent is one of the most popular algorithms for the optimization of neural networks and, in general, machine learning models (see the description below).

4. Stopping the process once a predefined criterion is reached: these criteria can be, for example, to reach an acceptable error rate or performance metric, or several predefined iterations (epochs), etc.

**Stochastic gradient descent.** This optimization algorithm aims to find the parameter values (W) that achieve the minimum of the loss function. Its strategy is to approximate the solution using an iterative method. The main steps are as follows:

1. Adjusting the model parameters iteratively to reduce the value of the cost function. At every iteration, the parameters are adjusted according to the opposite direction of the gradient of the cost. Mathematically, to find the new model parameters $W$ of a model $\hat{y} = f(x, W) = Wx$, with $\hat{y}$ as the output of the model $f(x, W)$, and $x$ representing the input data, the following steps are performed:

   - Compute the step size as the learning rate * gradient of the loss function:

   $$SS = \alpha \frac{\partial \mathcal{J}(y, \hat{y})}{\partial W} \tag{A.8}$$

   - Calculate the new parameters by removing the step size from the old parameters:

   $$W_{new} = W_{old} - SS \tag{A.9}$$

   The learning rate $\alpha$ is a flexible parameter that indicates how much to adjust the value of $W$ per iteration. It heavily influences the convergence of the algorithm.

2. These steps are repeated until the gradient is approximated to 0.

Other algorithms widely used for deep learning optimization include Stochastic gradient descent with momentum, RMSProp, and Adam Optimizer [328].

### A.1.4 Standard metrics

**Accuracy.** The accuracy is computed as the ratio of correctly predicted observations over the total observations in the target set, as follows:

$$accuracy(y, \hat{y}) = \frac{1}{N_{samples}} \sum_{k=0}^{N_{samples}-1} 1(\hat{y}_i = y_i) \,, \tag{A.10}$$

where $1(k)$ is the indicator function.

**Mean average precision (MAP).** The most well-known measure to evaluate the performance of the ranking of retrieved results is called mean average precision (MAP). This is defined as follows:

$$AP@K = \frac{1}{m} \sum_{i=1}^{K} P(i) \cdot rel(i) \,, \tag{A.11}$$

where $rel(i)$ is just an indicator that says whether that $ith$ item was relevant ($rel(i) = 1$) or not ($rel(i) = 0$). Finally, $MAP$ is the average precision ($AP$) over all samples.

**Euclidean distance between vectors $x$ and $y$.**

$$D_{Eux,y} = \sqrt{\sum_{j=1}^{J} (x_j - y_j)^2} \tag{A.12}$$

**Cosine distance between vectors $x$ and $y$.**

$$D_{Cos_{x,y}} = 1 - \frac{x \cdot y}{\|x\| \, \|y\|} \tag{A.13}$$

**Normalized correlation or Pearson correlation coefficient between vectors $x$ and $y$.**

$$D_{rpb_{x,y}} = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2))} \,, \tag{A.14}$$

where $m_x$ is the mean of the vector $x$ and $m_y$ is the mean of the vector $y$.

**T-test.** A t-test is a type of statistical test that is used to compare the means of two groups. It is one of the most widely used statistical hypothesis tests. The outcome of a hypothesis test does not tell us whether the alternative hypothesis is true. Instead, it tells us the probability that the null hypothesis could produce a "fake improvement" at least as extreme as the data we are testing. If there are runs from two different information retrieval systems: one baseline and a

proposed model. The goal then is to know whether the new model outperforms the baseline. To test this, both systems are tested in a number of different queries using the same collection and compare the difference in average precision values per query (see the computation of MAP presented above). The computation is as follows:

$$\begin{aligned} x1 &:= Baseline\ values \\ x2 &:= New\ values \\ \overline{d} &:= \overline{x1 - x2} \\ s_d &= stddev(x1 - x2) \\ n &:= number\ of\ samples \end{aligned} \tag{A.15}$$

Then

$$t := \frac{\overline{d}}{s_d/\sqrt{n}} \tag{A.16}$$

where t is on the Student's t-distribution with n-1 degrees of freedom

$$p := Pr(T > t) \tag{A.17}$$

Therefore, depending on this probability value, a t-test will tell us whether the difference is big enough[1].

[1] http://www.ccs.neu.edu/home/vip/teach/IRcourse/5_eval_userstudy/slides/Evaluation_2.pdf (accessed March 2021)

# APPENDIX B

# Appendix

## B.1 Learning strategies

In this section, we present the concepts around the most popular learning paradigms in multimodal learning. First, concerning the task (multi-class, multi-label, and multi-task) and second, the learning strategy regarding the data itself (supervised, semi-supervised, and unsupervised).

### B.1.1 Learning tasks

Some of the most popular tasks in multimodal learning for images and text are classification and annotation with some target set of textual representations such as labels. We describe them next.

**Multi-class.** Multi-class refers to the problem of classifying samples into one of several possible target classes. Each sample belongs to only one of these classes. For example, in the multi-class problem depicted in Figure B.1, the task is to classify images of birds according to the primary color of the bird, which in the example would be *Yellow*, and therefore, the output

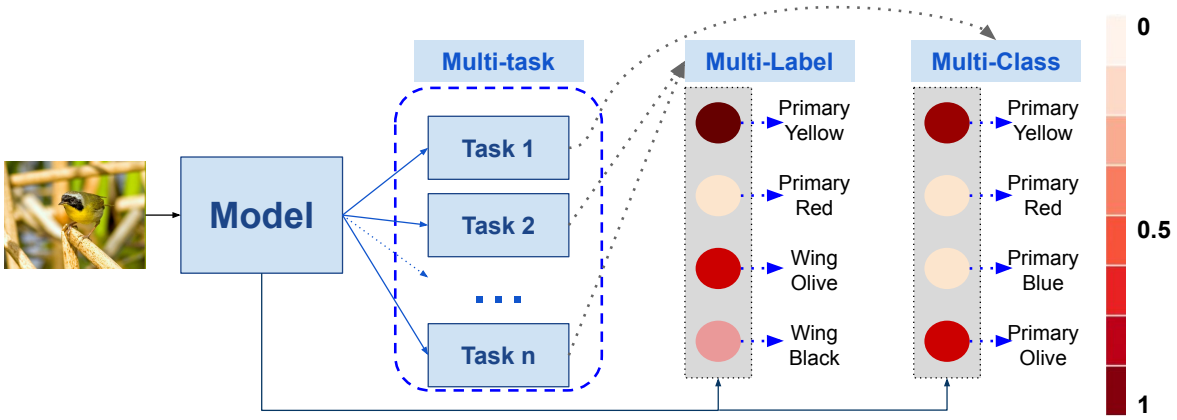



Figure B.1: Three main learning strategies: Multi-class (classify the sample only in one class), Multi-label (assign multiple labels to the sample) and Multi-task (optimize several tasks that can be a combination of multi-class and multi-label tasks).

with the biggest value should be the first one that corresponds to this class.

**Multi-label.** In this strategy, each sample is associated with one or several labels in the output. For example, in the multi-label problem depicted in Figure B.1, the task is to assign the most accurate set of labels that describe the image from a set of two attributes (*primary color of the bird* and *the main color of the wing*). For the sample image, the expected output would be *Primary Yellow* and *Wing Olive*.

**Multi-task.** This is a recent learning strategy in which multiple tasks are simultaneously learned by a shared model. This strategy comes with several advantages such as improving the data efficiency, reducing the over-fitting because of the shared representations, and it can accelerate learning by leveraging auxiliary information [329]. Each task is independent and can optimize different objective functions, in turn, expressed as a combination of multi-label, multi-class, and others (see Figure B.1).

### B.1.2 Training strategies

The three most popular strategies to learn used in machine learning models are supervised, unsupervised, and semi-supervised [330]. Essentially, this refers to the amount of labeled data (or supervision) available when studying a phenomenon of interest (see Figure B.2).

**Supervised learning.** In supervised learning, we have access to a full set of labeled data while training a model. This means that each example in the training set is labeled with the answer that the model should learn. Depending on the target task we want the model to learn to do automatically, in a dataset of birds, for example, each sample will have annotations such as the specific type or the colors for different body parts. Supervised learning is useful for two main areas: classification and regression tasks. Classification problems ask the model to predict a discrete value, identifying the input data as the member of a specific class. While regression aims to predict the most expected value for a set of variables defining a phenomenon.

**Unsupervised learning.** In unsupervised learning, the data even for training is a set of examples without specific annotations or correct answers. Hence, the model needs to attempt to automatically find the hidden structure in the data by extracting useful features and analyzing its structure without any supervision. Unsupervised learning is useful in tasks such as clustering, and anomaly detection where the data may be organized in different ways.

**Semi-supervised learning.** Finally, in semi-supervised learning, we have access to a dataset with both labeled and unlabeled data. This method is very useful when obtaining relevant features from the data is difficult, and labeling examples is a time-expensive task that many times requires experts. This strategy is closer to the realistic scenario and therefore, useful in all the tasks. Nowadays, for most applications, the goal is to design models that implement this learning strategy and therefore exploit all the data available (with and without any supervision).

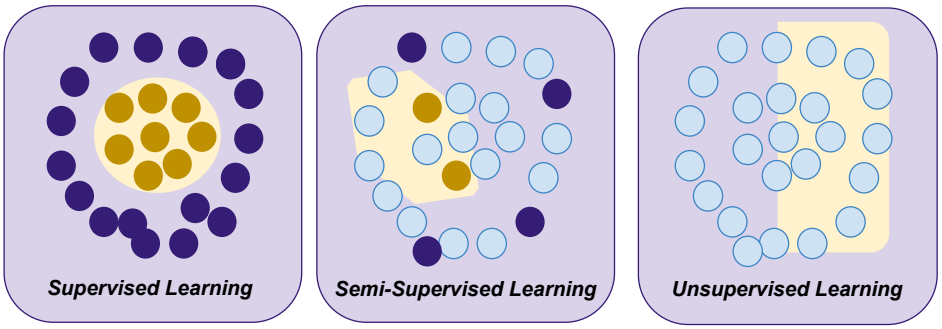



Figure B.2: Training strategies: 1) supervised learning in which each sample in the dataset contains information of classes, annotations, or another type of supervision; 2) semi-supervised learning in which only a few samples contain some type of supervision, and finally, 3) unsupervised learning in which no sample is annotated. In the example, the target task is to discriminate samples belonging to two classes and find the correct separation. While a supervised learning strategy can find a perfect separation (represented as the yellow region), a fully unsupervised one may struggle to discriminate among the samples and therefore not being able to find the correct separation.

# Bibliography


[1] Ben P Yuhas, Moise H Goldstein, and Terrence J Sejnowski. "Integration of acoustic and visual speech signals using neural networks". In: *IEEE Communications Magazine* 27.11 (1989), pp. 65–71.

[2] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal machine learning: A survey and taxonomy". In: *IEEE transactions on pattern analysis and machine intelligence* 41.2 (2018), pp. 423–443.

[3] Jean Piaget. *The language and thought of the child*. Vol. 5. Psychology Press, 2002.

[4] Sue Taylor Parker and Kathleen Rita Gibson. "A developmental model for the evolution of language and intelligence in early hominids". In: *Behavioral and Brain Sciences* 2.3 (1979), pp. 367–381.

[5] Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. "A survey on machine reading comprehension systems". In: *arXiv preprint arXiv:2001.01582* (2020).

[6] Stefania Montani and Manuel Striani. "Artificial intelligence in clinical decision support: a focused literature survey". In: *Yearbook of medical informatics* 28.1 (2019), p. 120.

[7] Juri Yanase and Evangelos Triantaphyllou. “A systematic survey of computer-aided diagnosis in medicine: Past and present developments”. In: *Expert Systems with Applications* 138 (2019), p. 112821.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[9] Micah Hodosh, Peter Young, and Julia Hockenmaier. “Framing image description as a ranking task: Data, models and evaluation metrics”. In: *Journal of Artificial Intelligence Research* 47 (2013), pp. 853–899.

[10] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. “Vqa: Visual question answering”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433.

[11] M Alex Meredith and Barry E Stein. “Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration”. In: *Journal of neurophysiology* 56.3 (1986), pp. 640–662.

[12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. “Sharpness-Aware Minimization for Efficiently Improving Generalization”. In: *arXiv preprint arXiv:2010.01412* (2020).

[13] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. “Deep learning based text classification: A comprehensive review”. In: *arXiv preprint arXiv:2004.03705* (2020).

[14] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. “Revisiting unreasonable effectiveness of data in deep learning era”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[16] Xulei Yang, Zeng Zeng, Sin G Teo, Li Wang, Vijay Chandrasekhar, and Steven Hoi. “Deep learning for practical image recognition: Case study on kaggle competitions”.

In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 923–931.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[18] Quan Le, Luis Miralles-Pechuán, Shridhar Kulkarni, Jing Su, and Oisín Boydell. "An overview of deep learning in industry". In: *Data Analytics and AI* (2020), pp. 65–98.

[19] Carole-Jean Wu, David Brooks, Kevin Chen, Douglas Chen, Sy Choudhury, Marat Dukhan, Kim Hazelwood, Eldad Isaac, Yangqing Jia, Bill Jia, et al. "Machine learning at facebook: Understanding inference at the edge". In: *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2019, pp. 331–344.

[20] Alfred Spector, Peter Norvig, and Slav Petrov. "Google's hybrid approach to research". In: *Communications of the ACM* 55.7 (2012), pp. 34–37.

[21] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. "Dive into deep learning". In: (2019).

[22] Li Deng and Dong Yu. "Deep learning: methods and applications". In: *Foundations and trends in signal processing* 7.3–4 (2014), pp. 197–387.

[23] Jack Stilgoe. "Machine learning, social learning and the governance of self-driving cars". In: *Social studies of science* 48.1 (2018), pp. 25–56.

[24] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. "The limits and potentials of deep learning for robotics". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 405–420.

[25] Keng Siau and Weiyu Wang. "Building trust in artificial intelligence, machine learning, and robotics". In: *Cutter Business Technology Journal* 31.2 (2018), pp. 47–53.

[26] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges". In: *Information Fusion* 58 (2020), pp. 52–68.

[27] Issam El Naqa, Dan Ruan, Gilmer Valdes, Andre Dekker, Todd McNutt, Yaorong Ge, Q Jackie Wu, Jung Hun Oh, Maria Thor, Wade Smith, et al. "Machine learning and mod-

eling: Data, validation, communication challenges". In: *Medical physics* 45.10 (2018), e834–e840.

[28] Ebtesam H Almansor and Farookh Khadeer Hussain. "Survey on Intelligent Chatbots: State-of-the-Art and Future Research Directions". In: *Conference on Complex, Intelligent, and Software Intensive Systems*. Springer. 2019, pp. 534–543.

[29] Yu Li, Chao Huang, Lizhong Ding, Zhongxiao Li, Yijie Pan, and Xin Gao. "Deep learning in bioinformatics: Introduction, application, and perspective in the big data era". In: *Methods* 166 (2019), pp. 4–21.

[30] Chunchun Chen, Pu Zhang, Yuan Liu, and Jun Liu. "Financial quantitative investment using convolutional neural network and deep learning technology". In: *Neurocomputing* 390 (2020), pp. 384–390.

[31] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and SS Iyengar. "A survey on deep learning: Algorithms, techniques, and applications". In: *ACM Computing Surveys (CSUR)* 51.5 (2018), pp. 1–36.

[32] Mohammad Jamshidi, Ali Lalbakhsh, Jakub Talla, Zdeněk Peroutka, Farimah Hadjilooei, Pedram Lalbakhsh, Morteza Jamshidi, Luigi La Spada, Mirhamed Mirmozafari, Mojgan Dehghani, et al. "Artificial intelligence and COVID-19: deep learning approaches for diagnosis and treatment". In: *IEEE Access* 8 (2020), pp. 109581–109595.

[33] Sudhen B. Desai, Anuj Pareek, and Matthew P. Lungren. "Deep learning and its role in COVID-19 medical imaging". In: *Intelligence-Based Medicine* 3-4 (2020), p. 100013.

[34] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. "Explainable deep learning: A field guide for the uninitiated". In: *arXiv preprint arXiv:2004.14545* (2020).

[35] Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'Connell, Terrance Gray, F Maxwell Harper, and Haiyi Zhu. "Explaining decision-making algorithms through UI: Strategies to help non-expert stakeholders". In: *Proceedings of the 2019 chi conference on human factors in computing systems*. 2019, pp. 1–12.

[36] Dong ping Tian et al. "A review on image feature extraction and representation techniques". In: *International Journal of Multimedia and Ubiquitous Engineering* 8.4 (2013), pp. 385–396.

[37] Yali Li, Shengjin Wang, Qi Tian, and Xiaoqing Ding. “A survey of recent advances in visual feature detection”. In: *Neurocomputing* 149 (2015), pp. 736–751.

[38] Berna Altınel and Murat Can Ganiz. “Semantic text classification: A survey of past and recent advances”. In: *Information Processing & Management* 54.6 (2018), pp. 1129–1153.

[39] Jose Camacho-Collados and Mohammad Taher Pilehvar. “From word to sense embeddings: A survey on vector representations of meaning”. In: *Journal of Artificial Intelligence Research* 63 (2018), pp. 743–788.

[40] Wenzhong Guo, Jianwen Wang, and Shiping Wang. “Deep multimodal representation learning: A survey”. In: *IEEE Access* 7 (2019), pp. 63373–63394.

[41] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. “Multimodal intelligence: Representation learning, information fusion, and applications”. In: *IEEE Journal of Selected Topics in Signal Processing* (2020).

[42] Dana Lahat, Tülay Adali, and Christian Jutten. “Multimodal data fusion: an overview of methods, challenges, and prospects”. In: *Proceedings of the IEEE* 103.9 (2015), pp. 1449–1477.

[43] Jing Gao, Peng Li, Zhikui Chen, and Jianing Zhang. “A survey on deep learning for multimodal data fusion”. In: *Neural Computation* 32.5 (2020), pp. 829–864.

[44] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. “A comprehensive survey of deep learning for image captioning”. In: *ACM Computing Surveys (CSUR)* 51.6 (2019), pp. 1–36.

[45] Tanveer Hussain, Khan Muhammad, Weiping Ding, Jaime Lloret, Sung Wook Baik, and Victor Hugo C de Albuquerque. “A comprehensive survey of multi-view video summarization”. In: *Pattern Recognition* 109 (2020), p. 107567.

[46] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. “Deep supervised cross-modal retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10394–10403.

[47] Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh, and Ajay Divakaran. “Align2ground: Weakly supervised phrase grounding guided by image-caption

alignment". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 2601–2610.

[48] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76.

[49] Vasili Ramanishka, Abir Das, Dong Huk Park, Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, and Kate Saenko. "Multimodal video description". In: *Proceedings of the 24th ACM international conference on Multimedia*. 2016, pp. 1092–1096.

[50] Sidra Shabir and Syed Yasser Arafat. "An image conveys a message: A brief survey on image description generation". In: *2018 1st International Conference on Power, Energy and Smart Grid (ICPESG)*. IEEE. 2018, pp. 1–6.

[51] Xiaoxiao Liu, Qingyang Xu, and Ning Wang. "A survey on deep neural network-based image captioning". In: *The Visual Computer* 35.3 (2019), pp. 445–470.

[52] Dengsheng Zhang, Md Monirul Islam, and Guojun Lu. "A review on automatic image annotation techniques". In: *Pattern Recognition* 45.1 (2012), pp. 346–362.

[53] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. "Meshed-Memory Transformer for Image Captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10578–10587.

[54] Himanshu Sharma, Manmohan Agrahari, Sujeet Kumar Singh, Mohd Firoj, and Ravi Kumar Mishra. "Image Captioning: A Comprehensive Survey". In: *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*. IEEE. 2020, pp. 325–328.

[55] Yuxin Peng, Jinwei Qi, and Yunkan Zhuo. "MAVA: Multi-Level Adaptive Visual-Textual Alignment by Cross-Media Bi-Attention Mechanism". In: *IEEE Transactions on Image Processing* 29 (2019), pp. 2728–2741.

[56] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. "Vizwiz grand challenge: Answering visual questions from blind people". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3608–3617.

[57] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. “Scene text visual question answering”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4291–4301.

[58] Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. “On the general value of evidence, and bilingual scene-text visual question answering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10126–10135.

[59] Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. “A comprehensive survey on cross-modal retrieval”. In: *arXiv preprint arXiv:1607.06215* (2016).

[60] Yifan Zhang, Wengang Zhou, Min Wang, Qi Tian, and Houqiang Li. “Deep Relation Embedding for Cross-Modal Retrieval”. In: *IEEE Transactions on Image Processing* 30 (2020), pp. 617–627.

[61] Fei Wu, Xiao-Yuan Jing, Zhiyong Wu, Yimu Ji, Xiwei Dong, Xiaokai Luo, Qinghua Huang, and Ruchuan Wang. “Modality-specific and shared generative adversarial network for cross-modal retrieval”. In: *Pattern Recognition* 104 (2020), p. 107335.

[62] Di Wang, Quan Wang, Lihuo He, Xinbo Gao, and Yumin Tian. “Joint and individual matrix factorization hashing for large-scale cross-modal retrieval”. In: *Pattern Recognition* 107 (2020), p. 107479.

[63] Vivek Sharma, Makarand Tapaswi, and Rainer Stiefelhagen. “Deep Multimodal Feature Encoding for Video Ordering”. In: *arXiv preprint arXiv:2004.02205* (2020).

[64] Zhongkai Sun, Prathusha Sarma, William Sethares, and Yingyu Liang. “Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8992–8999.

[65] Jesper E Van Engelen and Holger H Hoos. “A survey on semi-supervised learning”. In: *Machine Learning* 109.2 (2020), pp. 373–440.

[66] L Viviana Beltrán Beltrán, Juan C Caicedo, Nicholas Journet, Mickaël Coustaty, François Leceiller, and Antoine Doucet. "Deep Multimodal learning for Cross-Modal Retrieval: one model for all tasks". In: *Pattern Recognition Letters* (2021).

[67] Viviana Beltrán, Nicholas Journet, Mickael Coustaty, Antoine Doucet, et al. "Semantic Text Recognition via Visual Question Answering". In: *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. Vol. 5. IEEE. 2019, pp. 97–102.

[68] Viviana Beltrán, Mickaël Coustaty, Nicholas Journet, Juan C Caicedo, and Antoine Doucet. "An extended evaluation of the impact of different modules in ST-VQA systems". In: *International Conference on Pattern Recognition and Artificial Intelligence*. Springer, Cham. 2020, pp. 562–574.

[69] Vittorio Ferrari and Andrew Zisserman. "Learning visual attributes". In: *Advances in neural information processing systems*. 2008, pp. 433–440.

[70] Viviana Beltrán, Mickaël Coustaty, Nicholas Journet, Juan C Caicedo, and Antoine Doucet. "Multi-Attribute Learning with Highly Imbalanced Data". In: *25th International Conference on Pattern Recognition*. 2020.

[71] Zhiyuan Liu, Yankai Lin, and Maosong Sun. *Representation Learning for Natural Language Processing*. Springer Nature, 2020.

[72] Zhiyuan Liu, Yankai Lin, and Maosong Sun. "Word Representation". In: *Representation Learning for Natural Language Processing*. Singapore: Springer Singapore, 2020, pp. 13–41.

[73] Zellig S Harris. "Distributional structure". In: *Word* 10.2-3 (1954), pp. 146–162.

[74] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013), pp. 3111–3119.

[75] Yoav Goldberg and Omer Levy. "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method". In: *arXiv preprint arXiv:1402.3722* (2014).

[76] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[77] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

[78] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. "Deep contextualized word representations". In: *arXiv preprint arXiv:1802.05365* (2018).

[79] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[80] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[81] Yusuke Uchida. "Local feature detectors, descriptors, and image representations: A survey". In: *arXiv preprint arXiv:1607.08368* (2016).

[82] Tinne Tuytelaars and Krystian Mikolajczyk. *Local invariant feature detectors: a survey*. Now Publishers Inc, 2008.

[83] Mohamed Aly, Peter Welinder, Mario Munich, and Pietro Perona. "Automatic discovery of image families: Global vs. local features". In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2009, pp. 777–780.

[84] Gabriela Csurka and Florent Perronnin. "Fisher vectors: Beyond bag-of-visual-words image representations". In: *International Conference on Computer Vision, Imaging and Computer Graphics*. Springer. 2010, pp. 28–42.

[85] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[86] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[87] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[88] Dhanesh Ramachandram and Graham W Taylor. "Deep multimodal learning: A survey on recent advances and trends". In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 96–108.

[89] Zoubin Ghahramani and Michael I Jordan. "Factorial hidden Markov models". In: *Machine learning* 29.2 (1997), pp. 245–273.

[90] James H. Martin Daniel Jurafsky. *Speech and Language Processing*. 3rd ed. 10. https://web.stanford.edu/ jurafsky/slp3/, 2020.

[91] Mangi Kang, Jaelim Ahn, and Kichun Lee. "Opinion mining using ensemble text hidden Markov models for text classification". In: *Expert Systems with Applications* 94 (2018), pp. 218–227.

[92] Kevin Patrick Murphy and Stuart Russell. "Dynamic bayesian networks: representation, inference and learning". In: (2002).

[93] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[94] Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Morency Collins, and Trevor Darrell. "Hidden conditional random fields". In: *IEEE transactions on pattern analysis and machine intelligence* 29.10 (2007), pp. 1848–1852.

[95] Bernhard Schölkopf. "The kernel trick for distances". In: *Advances in neural information processing systems*. 2001, pp. 301–307.

[96] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[97] Pei Ling Lai and Colin Fyfe. "Kernel and nonlinear canonical correlation analysis". In: *International Journal of Neural Systems* 10.05 (2000), pp. 365–377.

[98] Jorge A Vanegas, Viviana Beltrán, Hugo Jair Escalante, and Fabio A González. "Transductive non-linear semantic embedding for multi-class classification". In: *Pattern Recognition Letters* 128 (2019), pp. 370–377.

[99] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. "Multiple kernel learning, conic duality, and the SMO algorithm". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 6.

[100] Mehmet Gönen and Ethem Alpaydın. “Multiple kernel learning algorithms”. In: *The Journal of Machine Learning Research* 12 (2011), pp. 2211–2268.

[101] Imad A Basheer and Maha Hajmeer. “Artificial neural networks: fundamentals, computing, design, and application”. In: *Journal of microbiological methods* 43.1 (2000), pp. 3–31.

[102] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[103] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. “Deep convolutional neural networks for LVCSR”. In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 8614–8618.

[104] Y.-S. Park and S. Lek. “Chapter 7 - Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling”. In: *Ecological Model Types*. Ed. by Sven Erik Jørgensen. Vol. 28. Developments in Environmental Modelling. Elsevier, 2016, pp. 123 –140.

[105] Anke Meyer-Baese and Volker Schmid. “Chapter 7 - Foundations of Neural Networks”. In: *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*. Ed. by Anke Meyer-Baese and Volker Schmid. Second Edition. Oxford: Academic Press, 2014, pp. 197 –243.

[106] Parminder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. “Comparative analysis on cross-modal information retrieval: A review”. In: *Computer Science Review* 39 (2021), p. 100336.

[107] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. “A new approach to cross-modal multimedia retrieval”. In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM. 2010, pp. 251–260.

[108] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. “Every picture tells a story: Generating sentences from images”. In: *European conference on computer vision*. Springer. 2010, pp. 15–29.

[109] Kevin Crowston. “Amazon mechanical turk: A research tool for organizations and information systems scholars”. In: *Shaping the Future of ICT Research. Methods and Approaches*. Springer, 2012, pp. 210–221.

[110] Mark J Huiskes and Michael S Lew. "The MIR flickr retrieval evaluation". In: *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. ACM. 2008, pp. 39–43.

[111] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. "Nus-wide: a real-world web image database from national university of singapore". In: *Proceedings of the ACM international conference on image and video retrieval*. 2009, pp. 1–9.

[112] Yuxin Peng, Xin Huang, and Yunzhen Zhao. "An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges". In: *IEEE Transactions on circuits and systems for video technology* 28.9 (2017), pp. 2372–2385.

[113] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[114] Yiling Wu, Shuhui Wang, and Qingming Huang. "Learning semantic structure-preserved embeddings for cross-modal retrieval". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 825–833.

[115] Harold Hotelling. "Relations between two sets of variates". In: *Breakthroughs in statistics*. Springer, 1992, pp. 162–190.

[116] Yunchao Wei, Y A O Zhao, Zhenfeng Zhu, Shikui Wei, Yanhui Xiao, Jiashi Feng, and Shuicheng Yan. "A Modality-dependent Cross-media Retrieval". In: V.212 (2015), pp. 1–13. arXiv: arXiv:1506.06628v2.

[117] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022.

[118] Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Nikhil Rasiwasia, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. "On the role of correlation and abstraction in cross-modal multimedia retrieval". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.3 (2014), pp. 521–535.

[119] Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. "Generalized multiview analysis: A discriminative latent space". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 2160–2167.

[120] Yiling Wu, Shuhui Wang, and Qingming Huang. "Multi-modal semantic autoencoder for cross-modal retrieval". In: *Neurocomputing* 331 (2019), pp. 165–175.

[121] Jie Shao, Zhicheng Zhao, and Fei Su. "Two-stage deep learning for supervised cross-modal retrieval". In: *Multimedia Tools and Applications* 78.12 (2019), pp. 16615–16631.

[122] Kaiye Wang, Ran He, Liang Wang, Wei Wang, and Tieniu Tan. "Joint Feature Selection and Subspace Learning for Cross-Modal Retrieval". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.10 (2016), pp. 2010–2023.

[123] Liang Xie, Lei Zhu, Peng Pan, and Yansheng Lu. "Cross-Modal Self-Taught Hashing for large-scale image retrieval". In: *Signal Processing* 124 (2016), pp. 81–92.

[124] Liang Zhang, Bingpeng Ma, Guorong Li, Qingming Huang, and Qi Tian. "Cross-modal retrieval using multiordered discriminative structured subspace learning". In: *IEEE Transactions on Multimedia* 19.6 (2016), pp. 1220–1233.

[125] Jinxing Li, Mu Li, Guangming Lu, Bob Zhang, Hongpeng Yin, and David Zhang. "Similarity and diversity induced paired projection for cross-modal retrieval". In: *Information Sciences* 539 (2020), pp. 215–228.

[126] Jianlong Wu, Xingxu Xie, Liqiang Nie, Zhouchen Lin, and Hongbin Zha. "Reconstruction Regularized Low-Rank Subspace Learning for Cross-Modal Retrieval". In: *Pattern Recognition* (2021), p. 107813.

[127] Guanqun Cao, Alexandros Iosifidis, Ke Chen, and Moncef Gabbouj. "Generalized multi-view embedding for visual recognition and cross-modal retrieval". In: *IEEE transactions on cybernetics* 48.9 (2017), pp. 2542–2555.

[128] Liang Zhang, Bingpeng Ma, Guorong Li, Qingming Huang, and Qi Tian. "Generalized semi-supervised and structured subspace learning for cross-modal retrieval". In: *IEEE Transactions on Multimedia* 20.1 (2017), pp. 128–141.

[129] Peng Hu, Dezhong Peng, Xu Wang, and Yong Xiang. "Multimodal adversarial network for cross-modal retrieval". In: *Knowledge-Based Systems* 180 (2019), pp. 38–50.

[130] Po-Yao Huang, Xiaojun Chang, Alexander G Hauptmann, et al. "Improving What Cross-Modal Retrieval Models Learn through Object-Oriented Inter-and Intra-Modal Attention Networks". In: *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. ACM. 2019, pp. 244–252.

[131] Peng Hu, Liangli Zhen, Dezhong Peng, and Pei Liu. "Scalable deep multimodal learning for cross-modal retrieval". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 635–644.

[132] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. "Finding beans in burgers: Deep semantic-visual embedding with localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3984–3993.

[133] Yonghao He, Shiming Xiang, Cuicui Kang, Jian Wang, and Chunhong Pan. "Cross-modal retrieval via deep and bidirectional representation learning". In: *IEEE Transactions on Multimedia* 18.7 (2016), pp. 1363–1377.

[134] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. "Heterogeneous Network Embedding via Deep Architectures". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15* (2015), pp. 119–128.

[135] Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. "Centralnet: a multilayer approach for multimodal fusion". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 0–0.

[136] Jian Wang, Yonghao He, Cuicui Kang, Shiming Xiang, and Chunhong Pan. "Image-Text Cross-Modal Retrieval via Modality-Specific Feature Learning". In: *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR '15* (2015), pp. 347–354.

[137] Christopher Thomas and Adriana Kovashka. "Preserving Semantic Neighborhoods for Robust Cross-modal Retrieval". In: *European Conference on Computer Vision*. Springer. 2020, pp. 317–335.

[138] Huaping Liu, Feng Wang, Xinyu Zhang, and Fuchun Sun. "Weakly-paired deep dictionary learning for cross-modal retrieval". In: *Pattern Recognition Letters* 130 (2020), pp. 199–206.

[139] Yash Patel, Lluis Gomez, Marçal Rusiñol, Dimosthenis Karatzas, and CV Jawahar. "Self-Supervised Visual Representations for Cross-Modal Retrieval". In: *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. ACM. 2019, pp. 182–186.

[140] Kaiyi Lin, Xing Xu, Lianli Gao, Zheng Wang, and Heng Tao Shen. “Learning cross-aligned latent embeddings for zero-shot cross-modal retrieval”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 11515–11522.

[141] Xing Xu, Huimin Lu, Jingkuan Song, Yang Yang, Heng Tao Shen, and Xuelong Li. “Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval”. In: *IEEE transactions on cybernetics* 50.6 (2019), pp. 2400–2413.

[142] Yale Song and Mohammad Soleymani. “Polysemous visual-semantic embedding for cross-modal retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1979–1988.

[143] Xi Zhang, Hanjiang Lai, and Jiashi Feng. “Attention-aware deep adversarial hashing for cross-modal retrieval”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 591–606.

[144] Ignazio Gallo, Alessandro Calefati, and Shah Nawaz. “Multimodal classification fusion in real-world scenarios”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 5. IEEE. 2017, pp. 36–41.

[145] Yiling Wu, Shuhui Wang, and Qingming Huang. “Online asymmetric similarity learning for cross-modal retrieval”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4269–4278.

[146] Yiling Wu, Shuhui Wang, Guoli Song, and Qingming Huang. “Online asymmetric metric learning with multi-layer similarity aggregation for cross-modal retrieval”. In: *IEEE Transactions on Image Processing* 28.9 (2019), pp. 4299–4312.

[147] Yunchao Wei, Yao Zhao, Canyi Lu, Shikui Wei, Luoqi Liu, Zhenfeng Zhu, and Shuicheng Yan. “Cross-modal retrieval with CNN visual features: A new baseline”. In: *IEEE transactions on cybernetics* 47.2 (2016), pp. 449–460.

[148] Daixin Wang, Peng Cui, Mingdong Ou, and Wenwu Zhu. “Deep multimodal hashing with orthogonal regularization”. In: *IJCAI International Joint Conference on Artificial Intelligence* 2015-Janua.Ijcai (2015), pp. 2291–2297.

[149] Di Hu, Xiaoqiang Lu, and Xuelong Li. “Multimodal Learning via Exploring Deep Semantic Similarity”. In: *Proceedings of the 2016 ACM on Multimedia Conference - MM ’16* (2016), pp. 342–346.

[150] Wenming Cao, Qiubin Lin, Zhihai He, and Zhiquan He. “Hybrid representation learning for cross-modal retrieval”. In: *Neurocomputing* 345 (2019), pp. 45–57.

[151] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial networks”. In: *arXiv preprint arXiv:1406.2661* (2014).

[152] Xing Xu, Li He, Huimin Lu, Lianli Gao, and Yanli Ji. “Deep adversarial metric learning for cross-modal retrieval”. In: *World Wide Web* 22.2 (2019), pp. 657–672.

[153] Bokun Wang, Yang Yang, Xing Xu, Alan Hanjalic, and Heng Tao Shen. “Adversarial cross-modal retrieval”. In: *Proceedings of the 25th ACM international conference on Multimedia*. 2017, pp. 154–162.

[154] Peng Hu, Xi Peng, Hongyuan Zhu, Jie Lin, Liangli Zhen, Wei Wang, and Dezhong Peng. “Cross-modal discriminant adversarial network”. In: *Pattern Recognition* (2020), p. 107734.

[155] Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. “Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7181–7189.

[156] Xing Xu, Jingkuan Song, Huimin Lu, Yang Yang, Fumin Shen, and Zi Huang. “Modal-adversarial semantic learning network for extendable cross-modal retrieval”. In: *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. 2018, pp. 46–54.

[157] Wen Gu, Xiaoyan Gu, Jingzi Gu, Bo Li, Zhi Xiong, and Weiping Wang. “Adversary Guided Asymmetric Hashing for Cross-Modal Retrieval”. In: *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. ACM. 2019, pp. 159–167.

[158] Fei Shang, Huaxiang Zhang, Lei Zhu, and Jiande Sun. “Adversarial cross-modal retrieval based on dictionary learning”. In: *Neurocomputing* 355 (2019), pp. 93–104.

[159] Yiling Wu, Shuhui Wang, Guoli Song, and Qingming Huang. “Augmented adversarial training for cross-modal retrieval”. In: *IEEE Transactions on Multimedia* (2020).

[160] Xu Wang, Peng Hu, Liangli Zhen, and Dezhong Peng. "DRSL: Deep Relational Similarity Learning for Cross-modal Retrieval". In: *Information Sciences* 546 (2021), pp. 298–311.

[161] Li He, Xing Xu, Huimin Lu, Yang Yang, Fumin Shen, and Heng Tao Shen. "Unsupervised cross-modal retrieval through adversarial learning". In: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2017, pp. 1153–1158.

[162] Xin Huang, Yuxin Peng, and Mingkuan Yuan. "Mhtn: Modal-adversarial hybrid transfer network for cross-modal retrieval". In: *IEEE transactions on cybernetics* 50.3 (2018), pp. 1047–1059.

[163] Xiushan Nie, Bowei Wang, Jiajia Li, Fanchang Hao, Muwei Jian, and Yilong Yin. "Deep multiscale fusion hashing for cross-modal retrieval". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2020).

[164] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S Yu. "Deep visual-semantic hashing for cross-modal retrieval". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 1445–1454.

[165] Devraj Mandal, Kunal N Chaudhury, and Soma Biswas. "Generalized semantic preserving hashing for n-label cross-modal retrieval". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4076–4084.

[166] Xing Xu, Fumin Shen, Yang Yang, Heng Tao Shen, and Xuelong Li. "Learning discriminative binary codes for large-scale cross-modal retrieval". In: *IEEE Transactions on Image Processing* 26.5 (2017), pp. 2494–2507.

[167] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. "Pairwise relationship guided deep hashing for cross-modal retrieval". In: *proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[168] Yufeng Shi, Xinge You, Feng Zheng, Shuo Wang, and Qinmu Peng. "Equally-Guided Discriminative Hashing for Cross-modal Retrieval." In: *IJCAI*. 2019, pp. 4767–4773.

[169] Dejie Yang, Dayan Wu, Wanqian Zhang, Haisu Zhang, Bo Li, and Weiping Wang. "Deep Semantic-Alignment Hashing for Unsupervised Cross-Modal Retrieval". In: *Proceedings of the 2020 International Conference on Multimedia Retrieval*. 2020, pp. 44–52.

[170] Xinzhi Wang, Xitao Zou, Erwin M Bakker, and Song Wu. "Self-constraining and attention-based hashing network for bit-scalable cross-modal retrieval". In: *Neurocomputing* 400 (2020), pp. 255–271.

[171] Ge Song, Dong Wang, and Xiaoyang Tan. "Deep memory network for cross-modal retrieval". In: *IEEE Transactions on Multimedia* 21.5 (2018), pp. 1261–1275.

[172] Cheng Deng, Zhaojia Chen, Xianglong Liu, Xinbo Gao, and Dacheng Tao. "Triplet-based deep hashing network for cross-modal retrieval". In: *IEEE Transactions on Image Processing* 27.8 (2018), pp. 3893–3903.

[173] Lei Ma, Hongliang Li, Fanman Meng, Qingbo Wu, and King Ngi Ngan. "Global and local semantics-preserving based deep hashing for cross-modal retrieval". In: *Neurocomputing* 312 (2018), pp. 49–62.

[174] Lin Wu, Yang Wang, and Ling Shao. "Cycle-consistent deep generative hashing for cross-modal retrieval". In: *IEEE Transactions on Image Processing* 28.4 (2018), pp. 1602–1612.

[175] Zhikai Hu, Xin Liu, Xingzhi Wang, Yiu-ming Cheung, Nannan Wang, and Yewang Chen. "Triplet Fusion Network Hashing for Unpaired Cross-Modal Retrieval". In: *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. ACM. 2019, pp. 141–149.

[176] Zhenyan Ji, Weina Yao, Wei Wei, Houbing Song, and Huaiyu Pi. "Deep Multi-Level Semantic Hashing for Cross-Modal Retrieval". In: *IEEE Access* 7 (2019), pp. 23667–23674.

[177] Shupeng Su, Zhisheng Zhong, and Chao Zhang. "Deep Joint-Semantics Reconstructing Hashing for Large-Scale Unsupervised Cross-Modal Retrieval". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3027–3035.

[178] De Xie, Cheng Deng, Chao Li, Xianglong Liu, and Dacheng Tao. "Multi-task consistency-preserving adversarial hashing for cross-modal retrieval". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 3626–3637.

[179] Tao Yao, Xiangwei Kong, Lianshan Yan, Wenjing Tang, and Qi Tian. "Efficient Discrete Supervised Hashing for Large-scale Cross-modal Retrieval". In: *arXiv preprint arXiv:1905.01304* (2019).

[180] Yixian Fang, Huaxiang Zhang, and Yuwei Ren. “Unsupervised cross-modal retrieval via Multi-modal graph regularized Smooth Matrix Factorization Hashing”. In: *Knowledge-Based Systems* 171 (2019), pp. 69–80.

[181] Gengshen Wu, Zijia Lin, Jungong Han, Li Liu, Guiguang Ding, Baochang Zhang, and Jialie Shen. “Unsupervised Deep Hashing via Binary Latent Factor Models for Large-scale Cross-modal Retrieval.” In: *IJCAI*. 2018, pp. 2854–2860.

[182] Chengkai Huang, Xuan Luo, Jiajia Zhang, Qing Liao, Xuan Wang, Zoe L Jiang, and Shuhan Qi. “Explore instance similarity: An instance correlation based hashing method for multi-label cross-model retrieval”. In: *Information Processing & Management* 57.2 (2020), p. 102165.

[183] Yu-Wei Zhan, Xin Luo, Yongxin Wang, and Xin-Shun Xu. “Supervised Hierarchical Deep Hashing for Cross-Modal Retrieval”. In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 3386–3394.

[184] Qiubin Lin, Wenming Cao, Zhihai He, and Zhiquan He. “Semantic deep cross-modal hashing”. In: *Neurocomputing* 396 (2020), pp. 113–122.

[185] Ruiqing Xu, Chao Li, Junchi Yan, Cheng Deng, and Xianglong Liu. “Graph Convolutional Network Hashing for Cross-Modal Retrieval.” In: *Ijcai*. 2019, pp. 982–988.

[186] Kai Li, Guo-Jun Qi, Jun Ye, and Kien A Hua. “Linear subspace ranking hashing for cross-modal retrieval”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.9 (2016), pp. 1825–1838.

[187] Chao Li, Cheng Deng, Lei Wang, De Xie, and Xianglong Liu. “Coupled cyclegan: Unsupervised hashing network for cross-modal retrieval”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 176–183.

[188] Song Liu, Shengsheng Qian, Yang Guan, Jiawei Zhan, and Long Ying. “Joint-modal Distribution-based Similarity Hashing for Large-scale Unsupervised Deep Cross-modal Retrieval”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, pp. 1379–1388.

[189] Tao Yao, Yaru Han, Ruxin Wang, Xiangwei Kong, Lianshan Yan, Haiyan Fu, and Qi Tian. “Efficient discrete supervised hashing for large-scale cross-modal retrieval”. In: *Neurocomputing* 385 (2020), pp. 358–367.

[190] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. "Self-supervised adversarial hashing networks for cross-modal retrieval". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4242–4251.

[191] Jian Zhang and Yuxin Peng. "Multi-pathway generative adversarial hashing for unsupervised cross-modal retrieval". In: *IEEE Transactions on Multimedia* 22.1 (2019), pp. 174–187.

[192] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. "Multi-level correlation adversarial hashing for cross-modal retrieval". In: *IEEE Transactions on Multimedia* 22.12 (2020), pp. 3101–3114.

[193] Haopeng Qiang, Yuan Wan, Lun Xiang, and Xiaojing Meng. "Deep semantic similarity adversarial hashing for cross-modal retrieval". In: *Neurocomputing* 400 (2020), pp. 24–33.

[194] Haopeng Qiang, Yuan Wan, Ziyi Liu, Lun Xiang, and Xiaojing Meng. "Discriminative deep asymmetric supervised hashing for cross-modal retrieval". In: *Knowledge-Based Systems* 204 (2020), p. 106188.

[195] Xin Liu, Yiu-ming Cheung, Zhikai Hu, Yi He, and Bineng Zhong. "Adversarial Tri-Fusion Hashing Network for Imbalanced Cross-Modal Retrieval". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* (2020).

[196] Frank Zalkow and Meinard Müller. "Using weakly aligned score-audio pairs to train deep chroma models for cross-modal music retrieval". In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2020, pp. 184–191.

[197] Qing-Yuan Jiang and Wu-Jun Li. "Deep cross-modal hashing". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3232–3240.

[198] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. "The limitations of adversarial training and the blind-spot attack". In: *arXiv preprint arXiv:1901.04684* (2019).

[199] Wenming Cao, Wenshuo Feng, Qiubin Lin, Guitao Cao, and Zhihai He. "A review of hashing methods for multimodal retrieval". In: *IEEE Access* 8 (2020), pp. 15377–15391.

[200] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. “Hierarchical question-image co-attention for visual question answering”. In: *Advances In Neural Information Processing Systems*. 2016, pp. 289–297.

[201] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. “Bottom-up and top-down attention for image captioning and visual question answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6077–6086.

[202] Vahid Kazemi and Ali Elqursh. “Show, ask, attend, and answer: A strong baseline for visual question answering”. In: *arXiv preprint arXiv:1704.03162* (2017).

[203] Yang Liu, Zhaowen Wang, Hailin Jin, and Ian Wassell. “Synthetically supervised feature learning for scene text recognition”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 435–451.

[204] Baoguang Shi, Xiang Bai, and Cong Yao. “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.11 (2016), pp. 2298–2304.

[205] Christian Bartz, Haojin Yang, and Christoph Meinel. “SEE: towards semi-supervised end-to-end scene text recognition”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[206] Fan Bai, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Shuigeng Zhou. “Edit probability for scene text recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1508–1516.

[207] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. “Towards vqa models that can read”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8317–8326.

[208] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. “Ocr-vqa: Visual question answering by reading text in images”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2019, pp. 947–952.

[209] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. "TAP: Text-Aware Pre-training for Text-VQA and Text-Caption". In: *arXiv preprint arXiv:2012.04638* (2020).

[210] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching Word Vectors with Subword Information". In: *arXiv preprint arXiv:1607.04606* (2016).

[211] Ali Furkan Biten, Rubèn Tito, Andres Mafla, Lluis Gomez, Marçal Rusiñol, Minesh Mathew, CV Jawahar, Ernest Valveny, and Dimosthenis Karatzas. "ICDAR 2019 Competition on Scene Text Visual Question Answering". In: *arXiv preprint arXiv:1907.00490* (2019).

[212] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: *ArXiv* abs/1910.03771 (2019).

[213] Anthony Kay. "Tesseract: an open-source optical character recognition engine". In: *Linux Journal* 2007.159 (2007), p. 2.

[214] Difei Gao, Ke Li, Ruiping Wang, Shiguang Shan, and Xilin Chen. "Multi-modal graph neural network for joint reasoning on vision and scene text". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12746–12756.

[215] Chenyu Gao, Qi Zhu, Peng Wang, Hui Li, Yuliang Liu, Anton van den Hengel, and Qi Wu. "Structured multimodal attentions for textvqa". In: *arXiv preprint arXiv:2006.00753* (2020).

[216] Yash Kant, Dhruv Batra, Peter Anderson, Alex Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. "Spatially aware multimodal transformers for textvqa". In: *arXiv preprint arXiv:2007.12146* (2020).

[217] Fen Liu, Guanghui Xu, Qi Wu, Qing Du, Wei Jia, and Mingkui Tan. "Cascade Reasoning Network for Text-based Visual Question Answering". In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 4060–4069.

[218] Lluís Gómez, Ali Furkan Biten, Rubèn Tito, Andrés Mafla, and Dimosthenis Karatzas. "Multimodal grid features and cell pointers for Scene Text Visual Question Answering". In: *arXiv preprint arXiv:2006.00923* (2020).

[219] Wei Han, Hantao Huang, and Tao Han. "Finding the Evidence: Localization-aware Answer Prediction for Text Visual Question Answering". In: *arXiv preprint arXiv:2010.02582* (2020).

[220] Zan-Xia Jin, Heran Wu, Chun Yang, Fang Zhou, Jingyan Qin, Lei Xiao, and Xu-Cheng Yin. "RUArt: A Novel Text-Centered Solution for Text-Based Visual Question Answering". In: *arXiv preprint arXiv:2010.12917* (2020).

[221] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[222] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. "Iterative answer prediction with pointer-augmented multimodal transformers for textvqa". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9992–10002.

[223] Qi Zhu, Chenyu Gao, Peng Wang, and Qi Wu. "Simple is not Easy: A Simple Strong Baseline for TextVQA and TextCaps". In: *arXiv preprint arXiv:2012.05153* (2020).

[224] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need. jun 2017". In: *URL http://arxiv. org/abs/1706.03762* (2019).

[225] Olga Russakovsky and Li Fei-Fei. "Attribute learning in large-scale datasets". In: *European Conference on Computer Vision*. Springer. 2010, pp. 1–14.

[226] Xiangyun Zhao, Yi Yang, Feng Zhou, Xiao Tan, Yuchen Yuan, Yingze Bao, and Ying Wu. "Recognizing Part Attributes with Insufficient Data". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 350–360.

[227] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. "Learning deep representation for imbalanced classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5375–5384.

[228] Ning Zhang, Manohar Paluri, Marc'Aurelio Ranzato, Trevor Darrell, and Lubomir Bourdev. "Panda: Pose aligned networks for deep attribute modeling". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1637–1644.

[229] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. "Deep imbalanced attribute classification using visual attention aggregation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 680–697.

[230] Saeed Anwar, Nick Barnes, and Lars Petersson. "A Systematic Evaluation: Fine-Grained CNN vs. Traditional CNN Classifiers". In: *arXiv preprint arXiv:2003.11154* (2020).

[231] Dat Huynh and Ehsan Elhamifar. "Fine-grained generalized zero-shot learning via dense attribute-based attention". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4483–4493.

[232] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.

[233] Genevieve Patterson and James Hays. "Sun attribute database: Discovering, annotating, and recognizing scene attributes". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 2751–2758.

[234] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. "Feature generating networks for zero-shot learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5542–5551.

[235] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015.

[236] Seyed Mohsen Shojaee and Mahdieh Soleymani Baghshah. "Semi-supervised zero-shot learning by a clustering-based approach". In: *arXiv preprint arXiv:1605.09016* (2016).

[237] Yang Liu, Jishun Guo, Deng Cai, and Xiaofei He. "Attribute attention for semantic disambiguation in zero-shot learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6698–6707.

[238] Jie Yang, Jiarou Fan, Yiru Wang, Yige Wang, Weihao Gan, Lin Liu, and Wei Wu. "Hierarchical feature embedding for attribute recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13055–13064.

[239] Kun Wei, Muli Yang, Hao Wang, Cheng Deng, and Xianglong Liu. "Adversarial fine-grained composition learning for unseen attribute-object recognition". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3741–3749.

[240] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. "Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1232–1241.

[241] Peixi Peng, Yonghong Tian, Tao Xiang, Yaowei Wang, Massimiliano Pontil, and Tiejun Huang. "Joint semantic and latent attribute modelling for cross-class transfer learning". In: *IEEE transactions on pattern analysis and machine intelligence* 40.7 (2017), pp. 1625–1638.

[242] Wenhe Liu, Xiaojun Chang, Ling Chen, and Yi Yang. "Semi-supervised bayesian attribute learning for person re-identification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[243] Xiao Wang, Shaofei Zheng, Rui Yang, Bin Luo, and Jin Tang. "Pedestrian attribute recognition: A survey". In: *arXiv preprint arXiv:1901.07474* (2019).

[244] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. "Improving person re-identification by attribute and identity learning". In: *Pattern Recognition* 95 (2019), pp. 151–161.

[245] Jiajiong Cao, Yingming Li, and Zhongfei Zhang. "Partially shared multi-task convolutional neural network with local constraint for face attribute learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4290–4299.

[246] Nathan Thom and Emily M Hand. "Facial Attribute Recognition: A Survey". In: *Computer Vision: A Reference Guide* (2020), pp. 1–13.

[247] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. “Deep imbalanced learning for face recognition and attribute prediction”. In: *IEEE transactions on pattern analysis and machine intelligence* 42.11 (2019), pp. 2781–2794.

[248] Xin Zheng, Yanqing Guo, Huaibo Huang, Yi Li, and Ran He. “A survey of deep facial attribute analysis”. In: *International Journal of Computer Vision* (2020), pp. 1–33.

[249] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep learning face attributes in the wild”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738.

[250] Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. “Describing people: A poselet-based approach to attribute classification”. In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 1543–1550.

[251] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. “The sun attribute database: Beyond categories for deeper scene understanding”. In: *International Journal of Computer Vision* 108.1-2 (2014), pp. 59–81.

[252] Tomoharu Iwata and Atsutoshi Kumagai. “Meta-learning from Tasks with Heterogeneous Attribute Spaces”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[253] Zihang Meng, Nagesh Adluru, Hyunwoo J Kim, Glenn Fung, and Vikas Singh. “Efficient relative attribute learning using graph neural networks”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 552–567.

[254] Nanyi Fei, Jiechao Guan, Zhiwu Lu, and Yizhao Gao. “Few-Shot Zero-Shot Learning: Knowledge Transfer with Less Supervision”. In: *Proceedings of the Asian Conference on Computer Vision*. 2020.

[255] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. “Attribute prototype network for zero-shot learning”. In: *arXiv preprint arXiv:2008.08290* (2020).

[256] Justin M Johnson and Taghi M Khoshgoftaar. “Survey on deep learning with class imbalance”. In: *Journal of Big Data* 6.1 (2019), pp. 1–54.

[257] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. “Learning from class-imbalanced data: Review of methods and applications”. In: *Expert Systems with Applications* 73 (2017), pp. 220–239.

[258] Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. "Training deep neural networks on imbalanced data sets". In: *2016 international joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 4368–4374.

[259] Robert O'Brien and Hemant Ishwaran. "A random forests quantile classifier for class imbalanced data". In: *Pattern recognition* 90 (2019), pp. 232–249.

[260] Sahar Sardari, Mahdi Eftekhari, and Fatemeh Afsari. "Hesitant fuzzy decision tree approach for highly imbalanced data classification". In: *Applied Soft Computing* 61 (2017), pp. 727–741.

[261] Qi Dong, Shaogang Gong, and Xiatian Zhu. "Class rectification hard mining for imbalanced deep learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1851–1860.

[262] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. "Class-balanced loss based on effective number of samples". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9268–9277.

[263] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. "Dynamic curriculum learning for imbalanced data classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5017–5026.

[264] Zhong Ji, Xuejie Yu, Yunlong Yu, Yanwei Pang, and Zhongfei Zhang. "A Semantics-Guided Class Imbalance Learning Model for Zero-Shot Classification". In: *arXiv preprint arXiv:1908.09745* (2019).

[265] Fei Wu, Xiao-Yuan Jing, Shiguang Shan, Wangmeng Zuo, and Jing-Yu Yang. "Multi-set feature learning for highly imbalanced data classification". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[266] Liangjun Feng and Chunhui Zhao. "Transfer Increment for Generalized Zero-Shot Learning". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[267] Rui Gao, Xingsong Hou, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, Zhao Zhang, and Ling Shao. "Zero-VAE-GAN: Generating unseen features for generalized and transductive zero-shot learning". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 3665–3680.

[268] Huajie Jiang, Ruiping Wang, Shiguang Shan, Yi Yang, and Xilin Chen. "Learning discriminative latent attributes for zero-shot classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4223–4232.

[269] Yuval Atzmon and Gal Chechik. "Probabilistic and-or attribute grouping for zero-shot learning". In: *arXiv preprint arXiv:1806.02664* (2018).

[270] Yuchen Guo, Guiguang Ding, Jungong Han, and Sheng Tang. "Zero-shot learning with attribute selection". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[271] Ping Gong, Xuesong Wang, Yuhu Cheng, Z Jane Wang, and Qiang Yu. "Zero-Shot Classification Based on Multitask Mixed Attribute Relations and Attribute-Specific Features". In: *IEEE Transactions on Cognitive and Developmental Systems* 12.1 (2019), pp. 73–83.

[272] Dat Huynh and Ehsan Elhamifar. "Compositional Zero-Shot Learning via Fine-Grained Dense Feature Composition". In: *Advances in Neural Information Processing Systems* 33 (2020).

[273] Chen Huang, Chen Change Loy, and Xiaoou Tang. "Unsupervised learning of discriminative attributes and visual representations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5175–5184.

[274] Liangchen Liu, Feiping Nie, Teng Zhang, Arnold Wiliem, and Brian C Lovell. "Unsupervised automatic attribute discovery method via multi-graph clustering". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE. 2016, pp. 1713–1718.

[275] Liangchen Liu, Feiping Nie, Arnold Wiliem, Zhihui Li, Teng Zhang, and Brian C Lovell. "Multi-modal joint clustering with application for unsupervised attribute discovery". In: *IEEE Transactions on Image Processing* 27.9 (2018), pp. 4345–4356.

[276] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. "A survey of zero-shot learning: Settings, methods, and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–37.

[277] Zhiyong Yang, Qianqian Xu, Xiaochun Cao, and Qingming Huang. "Task-Feature Collaborative Learning with Application to Personalized Attribute Prediction". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[278] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. "Curriculum Labeling: Self-paced Pseudo-Labeling for Semi-Supervised Learning". In: *arXiv preprint arXiv:2001.06001* (2020).

[279] Nikolaos Sarafianos, Theodoros Giannakopoulos, Christophoros Nikou, and Ioannis A Kakadiaris. "Curriculum learning of visual attribute clusters for multi-task classification". In: *Pattern Recognition* 80 (2018), pp. 94–108.

[280] Dongxiang Zhang, Rui Cao, and Sai Wu. "Information fusion in visual question answering: A Survey". In: *Information Fusion* 52 (2019), pp. 268–280.

[281] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. "LSTM: A search space odyssey". In: *IEEE transactions on neural networks and learning systems* 28.10 (2017), pp. 2222–2232.

[282] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. "Deep modular co-attention networks for visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6281–6290.

[283] Junyu Luo, Ying Shen, Xiang Ao, Zhou Zhao, and Min Yang. "Cross-modal Image-Text Retrieval with Multitask Learning". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM. 2019, pp. 2309–2312.

[284] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[285] Han Xiao. *bert-as-service*. https://github.com/hanxiao/bert-as-service. 2018.

[286] Cheng Wang, Haojin Yang, and Christoph Meinel. "A deep semantic framework for multimodal representation learning". In: *Multimedia Tools and Applications* 75.15 (2016), pp. 9255–9276.

[287] Aditya Krishna Menon, Didi Surian, and Sanjay Chawla. "Cross-Modal Retrieval: A Pairwise Classification Approach". In: *Proceedings of the 2015 SIAM International Conference on Data Mining* (2015), pp. 199–207.

[288] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[289] Yash Srivastava, Vaishnav Murali, Shiv Ram Dubey, and Snehasis Mukherjee. “Visual question answering using deep learning: A survey and performance analysis”. In: *arXiv preprint arXiv:1909.01860* (2019).

[290] Xiyan Liu, Gaofeng Meng, and Chunhong Pan. “Scene text detection and recognition with advances in deep learning: a survey”. In: *International Journal on Document Analysis and Recognition (IJDAR)* 22.2 (2019), pp. 143–162.

[291] Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. “Murel: Multimodal relational reasoning for visual question answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1989–1998.

[292] Ali Biten, Lluís Bigordà, C Jawahar, Dimosthenis Karatzas, Andrés Mafia, Minesh Mathew, Rubèn Tito, Marçal Rusiñol, and Ernest. Valveny. *ICDAR 2019 Robust Reading Challenge on Scene Text Visual Question Answering*. https://rrc.cvc.uab.es/?ch=11. 2019.

[293] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 19–27.

[294] Sebastian Sudholt and Gernot A. Fink. “PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents”. In: (2016). arXiv: 1604.00187.

[295] Francisco J Valverde-Albacete and Carmen Peláez-Moreno. “100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox”. In: *PloS one* 9.1 (2014), e84217.

[296] Yuzhe Yang and Zhi Xu. “Rethinking the value of labels for improving class-imbalanced learning”. In: *arXiv preprint arXiv:2006.07529* (2020).

[297] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. “Label-embedding for attribute-based classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 819–826.

[298] Xiao Liu, Jiang Wang, Shilei Wen, Errui Ding, and Yuanqing Lin. “Localizing by describing: Attribute-guided attention localization for fine-grained recognition”. In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[299] Sadaf Gulshad and Arnold Smeulders. “Explaining with Counter Visual Attributes and Examples”. In: *Proceedings of the 2020 International Conference on Multimedia Retrieval*. 2020, pp. 35–43.

[300] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. “Deep learning for computer vision: A brief review”. In: *Computational intelligence and neuroscience* 2018 (2018).

[301] Gary M Weiss, Kate McCarthy, and Bibi Zabar. “Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?” In: *Dmin* 7.35-41 (2007), p. 24.

[302] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[303] John Platt et al. “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods”. In: *Advances in large margin classifiers* 10.3 (1999), pp. 61–74.

[304] Min-Ling Zhang and Zhi-Hua Zhou. “ML-KNN: A lazy learning approach to multi-label learning”. In: *Pattern recognition* 40.7 (2007), pp. 2038–2048.

[305] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[306] Marie Dumont, Raphaël Marée, Louis Wehenkel, and Pierre Geurts. “Fast multi-class image annotation with random subwindows and multiple output randomized trees”. In: *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*. Vol. 2. 2009, pp. 196–203.

[307] Francisco Fernández-Navarro, César Hervás-Martínez, Javier Sanchez-Monedero, and Pedro Antonio Gutiérrez. “MELM-GRBF: a modified version of the extreme learning machine for generalized radial basis function neural networks”. In: *Neurocomputing* 74.16 (2011), pp. 2502–2510.

[308] Tony F Chan, Gene Howard Golub, and Randall J LeVeque. “Updating formulae and a pairwise algorithm for computing sample variances”. In: *COMPSTAT 1982 5th Symposium held at Toulouse 1982*. Springer. 1982, pp. 30–41.

[309] Donghyun Kim, Kuniaki Saito, Kate Saenko, Stan Sclaroff, and Bryan A Plummer. "Self-supervised Visual Attribute Learning for Fashion Compatibility". In: *arXiv preprint arXiv:2008.00348* (2020).

[310] Chufeng Tang, Lu Sheng, Zhaoxiang Zhang, and Xiaolin Hu. "Improving Pedestrian Attribute Recognition With Weakly-Supervised Multi-Scale Attribute-Specific Localization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 4997–5006.

[311] Liuyu Xiang, Xiaoming Jin, Guiguang Ding, Jungong Han, and Leida Li. "Incremental few-shot learning for pedestrian attribute recognition". In: *arXiv preprint arXiv:1906.00330* (2019).

[312] Yaohui Zhu, Weiqing Min, and Shuqiang Jiang. "Attribute-Guided Feature Learning for Few-Shot Image Recognition". In: *IEEE Transactions on Multimedia* (2020).

[313] Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. "Constrained semi-supervised learning using attributes and comparative attributes". In: *European Conference on Computer Vision*. Springer. 2012, pp. 369–383.

[314] Jie Song, Chengchao Shen, Jie Lei, An-Xiang Zeng, Kairi Ou, Dacheng Tao, and Mingli Song. "Selective zero-shot classification with augmented attributes". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 468–483.

[315] Qingrong Cheng and Xiaodong Gu. "Bridging multimedia heterogeneity gap via Graph Representation Learning for cross-modal retrieval". In: *Neural Networks* 134 (2021), pp. 143–162.

[316] Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio de Rezende, Yannis Kalantidis, and Diane Larlus. "Probabilistic Embeddings for Cross-Modal Retrieval". In: *arXiv preprint arXiv:2101.05068* (2021).

[317] Yu-Xi Huang, Wen-Xiao Wang, Sai Zhang, Yu-Ping Tang, and Shi-Jun Yue. "The database-based strategy may overstate the potential effects of traditional Chinese medicine against COVID-19". In: *Pharmacological research* (2020).

[318] Monica Sunkara, Chaitanya Shivade, Sravan Bodapati, and Katrin Kirchhoff. "Neural Inverse Text Normalization". In: *arXiv preprint arXiv:2102.06380* (2021).

[319] Arya Roy. “Recent Trends in Named Entity Recognition (NER)”. In: *arXiv preprint arXiv:2101.11420* (2021).

[320] Shujie Luo, Hang Dai, Ling Shao, and Yong Ding. “C4AV: Learning Cross-Modal Representations from Transformers”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 33–38.

[321] Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac, and Aida Nematzadeh. “Decoupling the Role of Data, Attention, and Losses in Multimodal Transformers”. In: *arXiv preprint arXiv:2102.00529* (2021).

[322] Fadi Thabtah, Suhel Hammoud, Firuz Kamalov, and Amanda Gonsalves. “Data imbalance in classification: Experimental evaluation”. In: *Information Sciences* 513 (2020), pp. 429–441.

[323] Ting Pang, Jeannie Hsiu Ding Wong, Wei Lin Ng, and Chee Seng Chan. “Semi-supervised GAN-based Radiomics Model for Data Augmentation in Breast Ultrasound Mass Classification”. In: *Computer Methods and Programs in Biomedicine* (2021), p. 106018.

[324] Katarzyna Janocha and Wojciech Marian Czarnecki. “On loss functions for deep neural networks in classification”. In: *arXiv preprint arXiv:1702.05659* (2017).

[325] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. “Activation functions: Comparison of trends in practice and research for deep learning”. In: *arXiv preprint arXiv:1811.03378* (2018).

[326] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[327] Kian Katanforoosh and Daniel Kunin. *Initializing neural networks*. deeplearning.ai. Accessed on 16.01.2019 at https://www.deeplearning.ai/ai-notes/initialization. 2019.

[328] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. “A survey of optimization methods from a machine learning perspective”. In: *IEEE transactions on cybernetics* (2019).

[329] Michael Crawshaw. “Multi-Task Learning with Deep Neural Networks: A Survey”. In: *arXiv preprint arXiv:2009.09796* (2020).

[330] Lars Schmarje, Monty Santarossa, Simon-Martin Schröder, and Reinhard Koch. “A survey on semi-, self-and unsupervised learning for image classification”. In: *arXiv preprint arXiv:2002.08721* 2 (2020).