Active and deep learning for multimedia

Mateusz Budnik

Communauté
UNIVERSITÉ Grenoble Alpes

Thèse

Pour obtenir le grade de

Docteur de l'Université de Grenoble

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Mateusz Budnik

Thèse dirigée par **Laurent Besacier et Georges Quénot**

préparée au sein **Laboratiore d'informatique de Grenoble**
et de **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)**

Active and deep learning for multimedia.

Apprentissage actif et profond pour le multimédia.

Thèse soutenue publiquement le **24 Février 2017**,
devant le jury composé de :

Mme. Catherine Berrut (présidente du jury)
Professeur, Université Grenoble-Alpes, LIG, Examinatrice
M. Philippe Joly
Professeur, Université Toulouse-III-Paul-Sabatier, IRIT, Rapporteur
M. Guillaume Gravier
Directeur de recherche CNRS, IRISA, Rapporteur
M. Hervé Bredin
Chargé de Recherche CNRS, LIMSI, Examinateur
M. Laurent Besacier
Professeur, Université Grenoble-Alpes, LIG, Directeur de thèse
M. Georges Quénot
Directeur de recherche CNRS, LIG, Directeur de thèse

Abstract

The main topics addressed in this thesis are the use of active learning and deep learning methods in the context of retrieval of multimodal document processing. The contributions proposed in this thesis address both these topics. An active learning framework was introduced for allowing for a more efficient annotation of broadcast TV videos thanks to the propagation of labels, to the use of multimodal data and to effective selection strategies. Several scenarios and experiments were considered in the context of person identification in videos, taking into account the use of different modalities (such as faces, speech segments and overlaid text) and different selection strategies. The whole system was additionally validated in a dry run test involving real human annotators.

A second major contribution was the investigation and use of deep learning (in particular the convolutional neural network) for video information retrieval. A comprehensive study was made using different neural network architectures and different training techniques such as fine-tuning or more classical classifiers like SVM. A comparison was made between learned features (the output of neural networks) and engineered features. Despite the lower performance of the latter, a fusion of these two types of features increases overall performance.

Finally, the use of convolutional neural network for speaker identification using spectrograms is explored. The results have been compared to those obtained with other state-of-the-art speaker identification systems. Different fusion approaches were also tested. The proposed approach obtained results comparable to those of some of the other tested approaches and offered an increase in performance when fused with the output of the best system.

Résumé

Les thèmes principaux abordés dans cette thèse sont l'utilisation de méthodes d'apprentissage actif et d'apprentissage profond dans le contexte du traitement de documents multimodaux. Les contributions proposées dans cette thèse abordent ces deux thèmes. Un système d'apprentissage actif a été introduit pour permettre une annotation plus efficace des émissions de télévision grâce à la propagation des étiquettes, à l'utilisation de données multimodales et à des stratégies de sélection efficaces. Plusieurs scénarios et expériences ont été envisagés dans le cadre de l'identification des personnes dans les vidéos, en prenant en compte l'utilisation de différentes modalités (telles que les visages, les segments de la parole et le texte superposé) et différentes stratégies de sélection. Le système complet a été validé au cours d'un "test à blanc" impliquant des annotateurs humains réels.

Une deuxième contribution majeure a été l'étude et l'utilisation de l'apprentissage profond (en particulier les réseaux de neurones convolutifs) pour la recherche d'information dans les vidéos. Une étude exhaustive a été réalisée en utilisant différentes architectures de réseaux neuronaux et différentes techniques d'apprentissage telles que le réglage fin (fine-tuning) ou des classificateurs plus classiques comme les SVMs. Une comparaison a été faite entre les caractéristiques apprises (la sortie des réseaux neuronaux) et les caractéristiques plus classiques ("engineered features"). Malgré la performance inférieure des seconds, une fusion de ces deux types de caractéristiques augmente la performance globale.

Enfin, l'utilisation d'un réseau neuronal convolutif pour l'identification des locuteurs à l'aide de spectrogrammes a été explorée. Les résultats ont été comparés à ceux obtenus avec d'autres systèmes d'identification de locuteurs récents. Différentes approches de fusion ont également été testées. L'approche proposée a permis d'obtenir des résultats comparables à ceux certains des autres systèmes testés et a offert une augmentation de la performance lorsqu'elle est fusionnée avec la sortie du meilleur système.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last several years, more and more multimedia documents are available thanks to the growing presence and use of cameras, smartphones and other recording devices. The use of the Internet and numerous social media enabled easy access to an unprecedented amount of diverse data. Also, due to on-line sharing and distribution the volume of this type of data is rapidly increasing. With this swift growth comes the need to make the data more useful and accessible to potential users. Annotation and indexing allows this multimedia content to be searchable. However, manual annotation of such a quantity of data is prohibitively expensive. In order to address this problem, many potential solutions were created.

A possible solution would be by the means of automatic multimedia indexing. This is done with the use of various machine learning methods. Such a system would be able to assign labels corresponding to the semantic content of a given multimedia document (be it an object that can be seen, a person or a speaker identity of an audio track) without human intervention, thus making it identifiable and traceable to a prospective user. To be able to construct such a system, which would also have a satisfying level of accuracy, and successfully train it, a large set of already annotated data is required. In most cases this data needs to be annotated by hand by human annotators. However, this process is always constrained by the costs in both time and resources. Because not all of the data can be labeled, it is necessary to prioritize and select some of the data instances over others. This can help in avoiding redundancies and lead to a potentially more representative set of labeled instances, which in turn can increase the overall performance of the machine learning model.

Active learning represents a set of algorithms designed to select appropriate instances from an unlabeled pool of data given a certain criterion. There is a wide range of potential criteria, which depend on the task at hand, but the main aim is to predict the usefulness of a new instance to a given model. Other methods can be used to increase the overall number of annotated samples thanks to label propagation. Also, active learning may help avoid annotating redundant instance, i.e. those that do not carry any new useful information.

As the amount of available annotated data increases, bigger and more complex models can be trained. This leads to the possibility of using state-of-the-art

classifiers such as deep learning. The most advanced models from this family of algorithms require a vast amount of annotated data to achieve top performance.

In this thesis, some active learning strategies are proposed that help with label propagation, which can increase the amount of useful labeled data and by consequence the overall performance of the trained models. Also, several different experiments with deep learning were undertaken to explore the usefulness of fine-tuning of networks, fusion and other aspects when applied to the treatment of multimedia. Finally, an additional application of this class of algorithms to speaker recognition is explored.

When it comes to the terminology used throughout this thesis, both *annotation* and *label* denote a textual keyword used to describe a concept (be it a name of an object or an identity of a person) that is present in a given data instance and they are used interchangeably. The same goes to the following: *learner*, *classifier* and *model*, all of which are considered synonymous. And lastly, the same applies to *feature* and *descriptor*, both of which in the most general sense are the means to describe and store information of a piece of data (be it an image, an audio file or other).

1.1 Active learning for multimedia

Most of the applications involving multimedia documents benefit or even require a certain amount of manual annotation. Given the complexity of some concepts (be it a particular person or a specific object), human intervention is necessary. With proper annotation, a lot of potential applications are made possible such as training models for recognition or retrieval. However, the annotation process (provided no prior source of labels is given) can be prohibitively expensive, which becomes even more evident when dealing with multimedia documents such as videos. In the case of the latter, often when giving a label some additional actions are also required, e.g. providing the location of the object within a frame of a video (by a simple box or a more complex shape) or adding a timestamp to denote when a given concept is visible. Those additional steps only compound the potential cost of each annotation.

Therefore, the use of active learning related methods can potentially be very beneficial in this context. When having limited time and resources, the ability to select the most informative segments of the data for annotation may be invaluable. Also, any likely source of weak labels such as overlaid names that appear on the screen or subtitles can greatly reduce the cost of annotation. By automatically extracting probable labels and assigning them to the most likely instance, the annotation process turns from identification into a verification task, which is usually faster and easier to perform.

1.2 Deep learning

Deep learning emerged recently as a very promising and effective set of algorithms that are able to tackle complex recognition problems given enough training data.

In particular, the convolutional neural network is very well suited when dealing with data composed of images or videos and it is capable to find high level concepts that appear within the data.

Contrary to the traditional approach where features describing an image are "hand-made", i.e. explicitly designed to extract certain characteristics from an image such as color or texture, convolutional neural networks are able to learn the most suitable features for a given dataset. This is by far the most important element that contributes to the superior performance over more traditional methods.

Due to significantly better overall performance in many vision-based tasks, the convolutional neural network methods became the most widely used methods when tackling problems such as image indexing and retrieval, image classification and many more. Due to this potential, a significant part of this thesis is dedicated to a further evaluation of its performance compared to the more traditional paradigm used for image classification. Also, further exploration of potential applications of the convolutional neural network to other domains is investigated.

1.3 Problem description

In this section, a more detailed problem statement for each aspect of this thesis is defined. This includes the central challenge and some of the main problems. The latter will be more precisely defined in the subsequent chapters.

1.3.1 Active learning for multimedia

The main challenge is to create an active learning framework that would be able to deal with a set of problems and constraints. In short, they can be defined as:

- Incorporating the data coming from different sources, i.e. the use of multimedia data.
- Making use of weak labels, including a way to verify its accuracy.
- Highly imbalanced datasets, where the use of trained model may be limited at least for some of the less frequent concepts.
- Potential use in practice, which puts constraints on the execution time of any proposed solution.

1.3.2 Deep learning applications

The application of deep learning in the context of video indexing poses several challenges. The main potential problems are:

- The highly imbalanced and noisy data, which makes it difficult to train the model from scratch.

- The presence of multiple concepts per frame, which would require a different approach than the one used in most image classification methods.
- Handling the volume of video data.

1.4 Contributions

The following contributions were made throughout the work on this thesis. They are related either to the application of deep learning in multimedia or to active learning and label propagation for person identification.

- A method for efficient multimedia annotation using active learning and label propagation. Several different sampling strategies are proposed. Additionally, different sources of information (faces, audio, written names) are used. Experiments have shown the usability of this approach for speaker model training. Details are presented in Chapter 3.
- A comparison between engineered and deep learning based features was made in Chapter 4. The use of deep learning models as feature extractors was tested as well as fine-tuning in the context of image indexing and retrieval. Fusion between engineered and deep features was also explored.
- A method for speaker identification based on a convolutional neural network trained on spectrograms is presented in Chapter 5. Several different fusion techniques were also proposed involving the output of the CNN and other state-of-the-art approaches for speaker identification.

1.5 Structure of this work

The remaining part of the manuscript is organized in the following way.

Chapter 2: Literature overview – This chapter provides the background and context of this work. It gives the introduction to the "classical pipeline" used for image classification and retrieval as well as to deep learning. Different active learning scenarios and approaches are discussed in the second part of this chapter.

Chapter 3: Active learning for multimedia – The chapter starts with a discussion of different limitations of active learning when used in practice. Afterwards, the problem of person annotation and label propagation in videos broadcasts is introduced. An active learning based solution is proposed and tested in a range of scenarios, including different modalities such as speech, faces and overwritten names. The chapter ends with a short discussion regarding a dry run where the proposed method was tested in a real-life scenario.

Chapter 4: Deep learning for multimedia – Here, the results of several deep learning experiments in the context of video indexing are presented. Deep learning models are used as feature extractors and as the basis for finetuning. Fusion with classical features is also explored.

Chapter 5: Deep learning for speaker identification – This chapter presents a deep convolutional neural network approach for speaker identification. Experiments also include fusion with state-of-the-art speaker recognition methods.

Chapter 6: Conclusions and perspectives – In this chapter, a general summary of the presented work and contributions is given. It also highlights some possible perspectives and future directions of research based on this thesis.

Chapter 2

Literature overview

2.1 Introduction

In this chapter the literature related to this work is reviewed and principal concepts are introduced. The overview starts with a presentation of classical approaches to image classification and retrieval, which focuses on popular descriptors and classifier algorithms. What follows in the second major part of this chapter is an introduction to deep learning, which methods recently produced state-of-the-art performance when dealing with tasks related to this thesis. They also made a lot of classical approaches obsolete in some tasks. The use of deep learning in multimedia (including audio, speech, video and still images) will be explored in greater detail. Next, the recent research on active learning is presented. This is followed by its application to multimedia documents such as videos and its use in practice. Finally, the most important datasets used in this thesis are described in greater detail. Some general conclusions are made at the end of this chapter.

2.2 Engineered descriptors

This section gives a brief introduction to the classical engineered descriptors used in computer vision. Even though most of them were recently surpassed in terms of performance by the CNN-based approaches, they can still be very useful. This type of descriptors, for example, are still considered as state-of-the-art for image retrieval [LLH15]. Additionally as it is going to be shown in Chapter 4, the classical descriptors can be used to enhance the results when certain fusion approaches are applied.

The name *engineered* descriptors emerged very recently as a way to distinguish between descriptors based on classical, "hand-crafted" feature extraction methods (such as SIFT) and automatically generated features based on deep learning. Figure 2.1 highlights this distinction. The engineered features are fixed and usually designed with some particular property in mind (such as to distinguish images having different colors or objects having different shapes). The advantage of this approach is that such a feature extractor is understandable to human (both the

motivation and the mechanism). In contrast, deep learning feature extractors can be trained and they change depending on the dataset. The higher performance obtained with this approach comes with a disadvantage of not knowing exactly why they work.

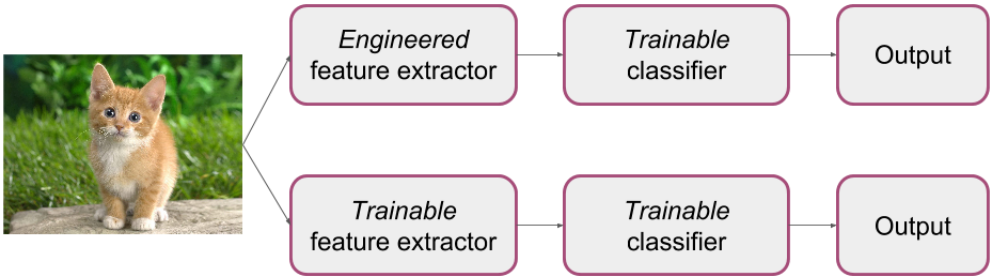


Figure 2.1: Two pattern recognition pipelines. *Upper row:* the classical pipeline made of fixed and crafted features followed by a trainable classifier. *Bottom row:* an approach present in deep learning where both the feature extractor and the classifier are trainable.

What follows in the rest of this section is a non-exhaustive overview of different categories of engineered descriptors: global, local and feature aggregation.

2.2.1 Global descriptors

Global descriptors provide a single comprehensive description of an image. These types of approaches are relatively fast to compute, which makes them useful when dealing with vast datasets or with limited computational resources. On the other hand, they are not very robust as a local change in an image (for example shift) influences the final value of the whole descriptor. Global descriptors focus on 3 main sources of information: color, texture and shape.

Color

The use of color as a descriptor was a popular choice for many years when dealing with object and scene recognition [VDSGS10b]. One of the key issues concerning color descriptors is the choice of the color space. The RGB space is probably the most popular. However, many other spaces exist, including HSV, HSL and CIELAB. In most cases color information from an image is globally aggregated in a form of a histogram. Color histograms indicate the global color distribution in a given image. This can be a robust solution, which can successfully cope with changes in perspective or rotation of an object. However, varying luminosity conditions can pose a challenge.

Texture

Another important source of information about an image are texture features. The assumption is that an image is composed of a set of texture regions, which can be used to retrieve or classify an image. Texture information can help when dealing with characteristic object surfaces such as glass, stone, wood, skin, etc. Gabor wavelet features are often used for this purpose [MM96]. This approach focuses on capturing the frequencies and principal orientation of the image.

Shape

A final example of a global descriptor is the general shape present in an image. In [OT01] an approach was proposed that enables the description of an image in terms of the scene it represents and its spatial structure. Rather than considering a scene as a configuration of separate objects, this method looks at a scene as a single object with a unitary shape. This approach is very efficient, however it is not robust to image transformations such as rotation.

2.2.2 Local descriptors

Due to some of the limitations of the global descriptors (such as the sensitivity to image transformation), local descriptors emerged as an alternative. These features can be invariant to geometric transformations, such as translation, and changes in illumination. By focusing on particular regions of an image and describing an image as a set of such regions, a more robust representation can be achieved. Also, due to the importance of local characteristics of an image, this type of features can potentially deal with partial occlusion.

Before a local descriptor can be extracted, an additional step to detect regions of interest in an image is required. Points (or regions) of interest can be considered as points in an image where the signal changes in two dimensions. This includes different types of corners, regions of high contrast (black dots on a white background) and any kind of texture. There are many approaches that were developed (an evaluation of different detectors can be found in [SMB00]), this includes contour based methods (extraction of junctions, line segments, ridge detection, etc.), intensity based methods (e.g., difference of a grayvalue between a given window and a shifted window, methods based on the auto-correlation matrix, etc.) and parametric model based methods (by fitting a parametric intensity model to a given signal). Once the appropriate regions are selected, a local descriptor can then be extracted.

In what follows, some of the most popular types of local descriptors are presented.

SIFT

Scale Invariant Feature Transform (SIFT) was presented for the first time in [Low04]. It is the most popular descriptor still in use for image and video indexing.

It is invariant to image translation, scaling and rotation. Also to some degree, it can handle changes in illumination.

Once the points of interest (or keypoints) are detected, this descriptor calculates the gradient magnitude and orientations for a region around each such point (as can be seen in Figure 2.2). Afterwards, a Gaussian window is used to apply a weight to each sample in the region. Finally, the sample are accumulated to create orientation histograms. Figure 2.2 shows an example where a 2×2 descriptor is generated based on a 8×8 set of samples.

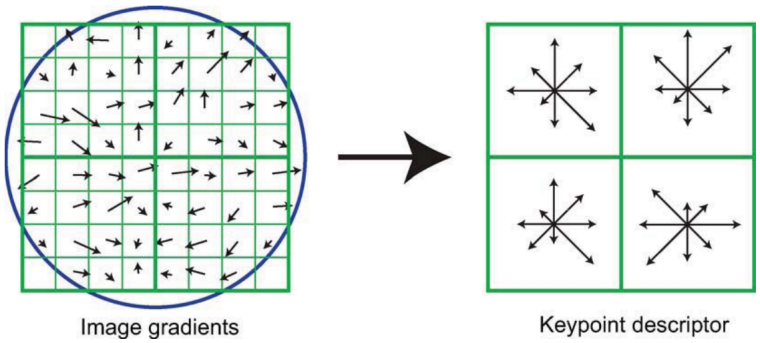


Figure 2.2: The stages of calculation of the SIFT descriptor. Image taken from [Low04].

The procedure described above is repeated for several different scales. One of the shortcomings of this descriptor in its basic form is its lack of color information. However, this problem was investigated in, for example, [VdSGS08] where different color extensions for the SIFT descriptor are proposed and compared. This includes the OpponentSIFT, which as an addition to the basic SIFT uses color histograms obtained in the opponent color space, the rgSIFT (using the normalized RGB color model) and Transformed color SIFT (where the transformed color histogram is used).

SURF

Speed Up Robust Features (SURF) [BTVG06] is a local descriptor that focuses on enhancing the computation time, but at the same time trying to maintain a competitive performance. It is also rotation and scale invariant. The detection part of this method is based on the Hessian matrix applied to integral images in order to reduce computation time. The feature itself (composed of only 64 dimensions) is a distribution of Haar-wavelet responses within the area around the point of interest. Additionally, the color information in used neither at the detection nor the descriptor stage.

HOG

The Histogram of Oriented Gradients (HOG) [DT05] were initially applied to human silhouette recognition. These features resemble the SIFT descriptor, however they are computed on a dense grid of small spatial regions (or cells), which

have the same distance from one another. Also, the overlapping local contrast normalization is used as an additional way to enhance performance by making it more invariant to changes in luminosity. Each cell is represented by a 1-D histogram of gradient directions or orientations of the edges computed based on the pixels within the cell. The histogram entries are then combined to create the final descriptor. Figure 2.3 provides an example of the visualization of the HOG features given an input image.

Figure 2.3: A given input image (*left*) and its corresponding HOG descriptor (*right*). Image taken from [DT05].

LBP

Local Binary Pattern (LBP) was first introduced in [OPH96]. This approach provides the description of texture patterns that occur in a given image. The texture information is gathered in the 3×3 neighborhood of each pixel (Figure 2.4a) by applying a binary pattern. This pattern is created by taking the value of the center pixel and using it as a threshold, i.e. other pixels in the neighborhood with values below the threshold are set to 0 and those above are set to 1 (the outcome can be seen in Figure 2.4b). The resulting pattern is then multiplied by weights corresponding to the pixel position (Figure 2.4c) and the result (Figure 2.4d) is summed up together to give the final single value (169). These values calculated for each pixel and then accumulated in the form of a histogram to give the final descriptor.

The Orthogonal Combination of Local Binary Patterns (OC-LBP) was presented for the first time in [ZBC13] and can be considered as one of the more recent extensions of the LBP descriptor. The main goal here was to reduce the dimensionality of the original descriptor, which would greatly speed up the computation process. This is achieved by splitting the original neighborhood of 8 pixels to two orthogonal sets of 4 neighbors each. One set consists of horizontal and vertical neighbors while the other contains the diagonal pixels. This enables the size reduction of the original 256-value vector to a vector of just 32 dimensions.

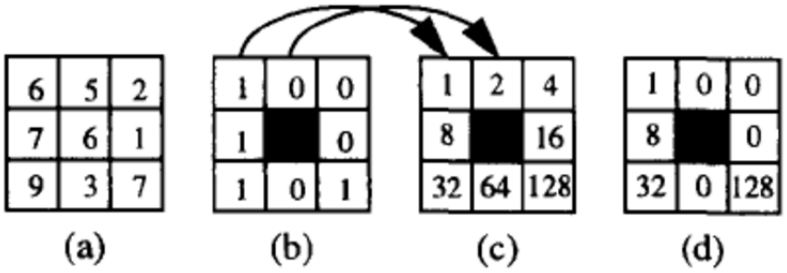


Figure 2.4: The steps used to calculate the LBP descriptor. Image taken from [OPH96].

STIP

Spatio-Temporal Interest Points [Lap05] (or STIP) is a local image descriptor that not only represents spatial interest points (as is the case with every local descriptor presented here thus far), but also extends them in time, which is crucial when dealing with video data. The points of interest that are extracted with the use of this method often correspond to certain events in the video (e.g., hands clapping, water splash, etc.). In order to make it possible the Harris interest point detector, which is designed to detect corners in images, is extended to the 3D space by incorporating the time dimension. After the interest points are detected, their subsequent representation is made using either the HOG descriptor or the histogram of optical flow.

2.2.3 Feature aggregation

The number of local features that can be extracted from a given image may vary greatly depending on the content of that image, such as the number of corners, type of textures, etc. For very detailed images this may results in a enormous amount of features that take a great amount of time to compute and apply. This issue becomes even more pronounced when dealing with videos, to the point of not being practically usable. To address this problem, aggregation of features can be performed, which reduces the size of the descriptor and makes it more global. Two of the most popular approaches to feature aggregation are Bag of Visual Words and Fisher Kernel. Also, the VLAD algorithm is presented as well as its extension, the VLAT algorithm, as an example of more recent developments in the field.

Bag of Visual Words

The Bag of Visual Words (BoVW) method was introduced in [SZ03a] and was inspired by the bag of words technique [Har54] used to create descriptors of text documents based on frequency of every word appearing in that document. The BoVW is probably the most widely used aggregation approach. In essence, this method tries to represent an image by taking into account the frequency of

appearance of each element (e.g. a local descriptor) present in that image.

After the extraction of local descriptors is complete, the generation of a visual dictionary takes place. The most common approach involves the use of the k-means clustering algorithm, which is applied on all the local descriptors generated from a given set of images or video. Each resulting centroid is then used as a visual word, i.e. is added to the visual dictionary. To generate the BoVW representation for an image each local descriptor of that image is assigned to the closest visual word (based on the shortest distance between the local descriptor and the centorids). The final representation is created in a form of a histogram of the frequency of visual words present in the image. This more global representation of an image is more robust to changes in perspective and deformations. It can also handle partial occlusions.

Fisher Kernel

A potential alternative to the BoVW approach is the Fisher Kernel method. Its use as a way to aggregate features was first proposed in [PD07]. Here, the Gaussian Mixture Model is used to approximate the distribution of the local features (or visual words) extracted from a given image to create the visual vocabularies. Afterwards, the Fisher Kernels are applied to these vocabularies, which in turn provides a natural similarity measure which can be used with a discriminative classifiers. This approach to feature aggregation seems to extend the BoVW method by not only including the frequency of occurrence of each local descriptor, but also takes into account their distribution.

VLAD

Vectors of Locally Aggregated Descriptors (VLAD) [JDSP10] is an aggregation method reminiscent of the previously discussed Fisher Kernel approach and can be considered as its simplification. Similarly, the local descriptors can be grouped together using the Gaussian Mixture Model or K-means clustering to generate the visual vocabulary. In the case of VLAD, however, the hard membership is used, contrary to the soft one used in the Fisher Kernel method. Afterwards, every local descriptor is assigned to the nearest visual word. The main novelty is that the description of each visual word is made out of the accumulated difference between each local feature assigned to that visual word and the visual word itself. The result consists of the distribution of local features with respect to the center, which in turn is used as the final descriptor.

VLAT

Vectors of Locally Aggregated Tensors (VLAT) [PG13] is a proposed extension of the VLAD descriptor, which involves the use of the aggregation of the tensor product of local descriptors. After the set of visual words is calculated by using the K-means clustering algorithm, each visual word is described by a signature over local descriptors that are closest to that visual word. The signature is composed of:

- the difference between each local descriptor and the visual word (the same as in VLAD),
- the sum of self tensor product of descriptors belonging to a given visual word.

The signature is created by the concatenation of these two elements. For the final image descriptor a vector is created, which includes the flattened signatures for each visual word in the image.

Compact codes

One of the major obstacles when dealing with visual descriptors is speed. As a lot of datasets contain tens of millions of images and each image needs to be represented by a complex descriptor, often enough tasks like image retrieval may be prohibitively expensive. One of the potential solutions is to have a more compact representation of an image, usually as a final step after the feature aggregation (using one of the methods mentioned above) is done.

In [JPD+12] a coding method is proposed, which goal is to encode the image descriptor so that its representation consists of a specified number of bits and a nearest neighbor of a given non-encoded query can be efficiently found in the database of coded image descriptors. To achieve this, two steps have to be applied and optimized jointly. First is a proper dimensionality reduction (such as principal component analysis), which is followed by a quantization step that is used for indexing the vectors. To address the latter step, a variant of approximate nearest neighbor search method was used, namely the asymmetric distance computation.

2.3 Classical classification methods

Because of their use throughout this work, some of the classical classification algorithms are shortly introduced in this section. A very general description is made here, while more specific information (such as particular parameter values and so on) are provided in the relevant chapters.

k-NN

The k Nearest Neighbors algorithm is one of the most well known and frequently used classification algorithm. This is due to its simplicity, intuitiveness and a small amount of parameters that need to be tuned [CH67]. An additional advantage is that it does not require a model to be learned in order to make predictions, but rather the class of each new unidentified instance is decided based on the labels of its immediate k neighbors. The decision is usually made by voting, often weighted based on the distance from the instance.

There are several potential disadvantages of this approach (at least in the case of the standard implementation), which often depend on the dataset at hand. When dealing with bigger datasets (in terms of the number of features

and instances), the prediction stage can be both time and memory consuming, due to the need to store every training sample in memory and the calculation of the distance for each of those samples. Also, the performance of the algorithm can be reduced when presented with noisy features. Often, feature selection or reduction is required to obtain satisfying results [WA95]. Finally, the overall accuracy greatly depends on the value of k and the distance measure that is used, which can be problematic.

SVM

The Support Vector Machine (SVM) algorithm was first presented in [CV95] and it is one of the most popular classifiers, frequently used in many classification tasks. The main idea is that, given a two class classification problem, the SVM algorithm finds the optimal hyperplane that separates the data points belonging to those two classes in such a way that the gap (or margin) between them is as wide as possible. Figure 2.5 gives an example of a hyperplane H dividing a 2D space; the margin (dotted lines) is also visible.

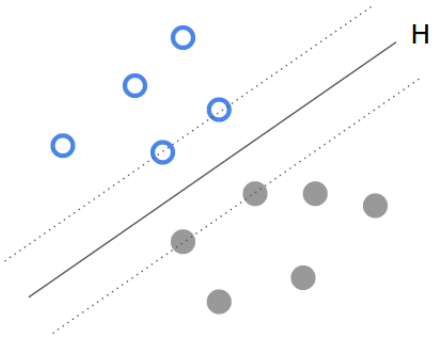


Figure 2.5: A linear separation of two classes in a 2D space produced by the SVM algorithm. Points lying on the margin (dotted lines) are the support vectors.

However, real life data is usually not linearly separable. In order to deal with this problem, the SVM algorithm can perform a mapping to a higher dimensional space where the data may be separable. This can be achieved with the use of the *kernel trick* described in [SS02], which bypasses the requirement of mapping the data into the new feature space and instead computes only the inner product of two data points in that feature space.

The basic version of the SVM algorithm can be defined in the following way. When dealing with a two class problem, the training dataset of n points has the following form : $(x_1, y_1), ..., (x_n, y_n)$, where y_i denotes the label of a given data point with values either -1 or 1 and x_i represents a vector of real values and d dimensions. The classification hyperplane produced by the SVM can then be defined as:

$$\langle w, \Phi(x) \rangle + b = 0 \tag{2.1}$$

where $\Phi(\cdot)$ can be considered as a mapping between R^d and a higher dimensional Hilbert space, while $\langle \cdot, \cdot \rangle$ is the dot product in that space. Also, w is the normal vector to the hyperplane and b is the offset. Given the above, the decision function $f(x)$ can now be defined as follows:

$$f(x) = sign(\langle w, \Phi(x) \rangle + b) \tag{2.2}$$

In order to find the optimal hyperplane with the maximum margin between 2 classes, the following quadratic optimization problem needs to be solved:

$$\text{minimize } \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \|w\|^2 \tag{2.3}$$

$$\text{subject to } y_i(\langle w, \Phi(x) \rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \text{ for all } i \tag{2.4}$$

where λ is a parameter that defines the trade-off between the size of the margin and that the vectors x_i are lying on the proper side of the margin.

MSVM

Multi-learner SVM (MSVM) approach is an expansion of the SVM algorithm, which makes use of under-sampling and ensemble learning approaches. It was designed to deal with binary classification in highly imbalanced datasets and was first presented in [SQ10]. To create an ensemble, the two class training set is sub-sampled to create sets, which contain all of the minority class instances and a random subset of the majority class. Afterwards, an SVM is trained on each of the subsets. The actual proportion between minority and majority class as well as the number of the sets is determined by setting appropriate parameters. It is worth noting that in the case of a balanced dataset only a single SVM model is used. The final decision is made through a fusion function over the prediction scores from every SVM model in the ensemble. Many fusion methods are possible, including the arithmetic mean, maximum, etc.

2.4 Fusion

The use of fusion has many potential advantages. Nowadays, a lot of data is multimodal (for example videos) and the ability to combine information from different sources can be very beneficial to the overall accuracy of the classification system. Also in the case of a monomodal data, the fusion of different models trained on the same data often leads to increased performance. In order to enhance the classification results different fusion methods are possible. Fusion can be applied at the descriptor level, before the classification model is trained, namely early fusion. Late fusion consists of combining the outputs of classifiers. Finally, kernel fusion is done during the classification step and it is performed on the computed kernel. The rest of this section provides some more detail about each of the fusion methods.

2.4.1 Early fusion

Early fusion (or feature-level fusion) combines the descriptors before the learning process takes place. This has the advantage of providing a single representation of a multimodal dataset, which in turn requires only a single model to be trained. Also, the correlation between different modalities (or descriptor types) can be used at early stage, which in turn may be beneficial to the overall performance of the system [AHESK10]. One of the potential difficulties is the need for a proper format for every descriptor before fusion. Also, the concatenation of several features may lead to very large descriptors. This may cause longer training time and, potentially, an increased complexity of a model.

2.4.2 Late fusion

In late fusion, the final output is based on scores taken from individual models. Each model may be trained on a different descriptor and the scores are later fused together. In the case of multimedia documents (where different sources of information are available such as images, audio, motion or text), each model can be trained on a specific modality, which would be the most appropriate for it [SWS05]. Another type of late fusion is used extensively in the classifier ensemble algorithms where all the individual models are trained using the same modality. For this approach to work, individual models need to be of a different type (like in Stacking introduced in [Wol92]) or a same model type needs to have diverse perspectives of the same data (e.g., different subset of the data like in Bagging [Bre96] or different feature space like in Random Forest [Bre01]).

One of the main drawbacks of late fusion is the increased computational requirements when compared to a single model. In most cases this can be resolved through parallel processing (training all individual models at the same time), but this is only an apt solution when proper hardware is available. Additionally, this solution only applies to methods in which the training of an individual model is not dependent on the output of the previous one (as is the case with the Boosting algorithm [FS95]).

2.4.3 Kernel fusion

Finally, kernel fusion (being closely related to multiple kernel learning [BLJ04]) can be considered as intermediate approach in relation to the previous two described earlier. In this case, the fusion takes place on computed kernels within the classification step for the kernel-based methods such as SVMs. Among other advantages, this approach enables the choice of an optimal kernel for each information source and its descriptor.

This is by no means exhaustive; other fusion types also exist. They can be considered as a combination or an extension of some of the above approaches (including, for example, meta learning [VD02]).

2.5 Deep learning

In recent years, neural networks got more and more attention from the research community. It is mostly due to the deep neural networks that were able to achieve outstanding performance on numerous high profile tasks. The best example is the algorithm introduced in [KSH12], which won the ImageNet 2012 competition by a large margin. This was followed by additional successes in many different fields and applications ranging from face identification [TYRW14] and human action recognition in videos [JXYY13] to language identification [LMGDP+14]. These accomplishments happening in such a short time frame propelled them to become a general baseline and a point of reference for a majority of tasks dealing with signal processing.

The rise of interest in these methods can be attributed to a number of factors. Even though most of the methods are not recent[1], some recent advances have been made which enabled the wider application of these methods. One of the main factors was the use of graphics processing units (GPU) for faster model training, which gave speedups ranging between 5 and 70-fold over a standard CPU implementation depending on the model and application [RMN09]. Each modern GPU contains hundreds (or even thousands) processing cores, which allow for a very efficient parallelism of computation. This in turn goes well with most neural network architectures, which can be easily parallelized (processing batches of instances at a time, the convolution computation, etc.). The use of GPUs not only helped to train models faster, but more importantly it made feasible the construction of more complex and deeper architectures.

The second factor is the growing availability of bigger datasets. Especially in the field of image recognition, the introduction of more complex and sizable sets of annotated images opened up the possibility for use of more complex algorithms. A prime example is the ImageNet challenge dataset [DDS+09], which provides over 1 million labeled images of 1000 different concepts (for the whole database the numbers are even bigger with over 14 million images and almost 22 thousand categories). This dataset serves as the main evaluation set for most of the recently developed convolutional neural network models (including [KSH12] and [SLJ+14] among others).

The third element comes from the improvement in the neural network algorithm itself. The introduction of the rectified linear unit (ReLU) in [GBB11] helped made the training of deep neural networks faster. Also, before for very deep networks it was necessary to pre-train one layer at a time [HOT06]. With the use of ReLU this is no longer needed.

In this section a general overview of deep neural network is presented. This includes their history. Their recent application to multimedia is also presented in detail.

[1] E.g., one of the earlier works involving CNNs can be found in [LBBH98] and the first paper on CNN trained with backpropagation here [LCDH+90].

2.5.1 Multilayer perceptron

The artificial neural network is one of the most popular and recognized machine learning algorithms. Its first iteration was introduced in 1958 and presented the Perceptron [Ros58], a supervised binary classifier, which makes predictions based on the weighted sum of its inputs, as follows:

$$f(x) = \begin{cases} 1 & \text{if } \sum_{i=0}^{m} w_i x_i + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where w is the weight vector having the length equal to m, x is the feature or input vector and b represents the bias.

The Perceptron is a linear classifier, which means that it is unable to classify all of the instances correctly if they are not linearly separable. Its use is, therefore, limited in practice as most of the real life classification problems share this characteristic, especially for more sophisticated applications such as distinguishing between two breeds of dogs. In order to deal with this problem, a more complex model was proposed that would make use of several layers, each composed of a number of individual perceptrons.

2.5.2 Backpropagation and gradient descent

The backpropagation algorithm was proposed in [RHW88] as a way to make efficient supervised training of multilayer networks possible. This approach is usually used alongside gradient descent as an optimization method. The backpropagation is composed out of two stages: the forward pass or propagation and the backward pass when the weights are updated. Figure 2.6 gives an example of a multilayer perceptron, which is comprised out of four layers: input, output and 2 hidden layers. The forward and backward passes are also denoted.

The forward pass propagates a given training vector from the input layer all the way to the output where it is compared to the target and the loss is calculated. Following the notation presented in Figure 2.6, the forward pass may be described as:

$$z_j = \sum_{i \epsilon Input} w_{ij} x_i \qquad z_k = \sum_{j \epsilon H1} w_{jk} y_j \qquad z_l = \sum_{k \epsilon H2} w_{kl} y_k$$

$$y_j = f(z_j) \qquad y_k = f(z_k) \qquad y_l = f(z_l)$$

where x_i is the initial input, z is the weighted sum of the inputs to a given neuron (before the activation function is applied), y denotes the final output of a neuron, $f(\cdot)$ is the activation function that is applied at the output of each neuron and w corresponds to a weight between two neurons.

The backward pass is used to update the weights. The equations shown below can be applied to calculate the backward pass using the notation introduced in Figure 2.6.

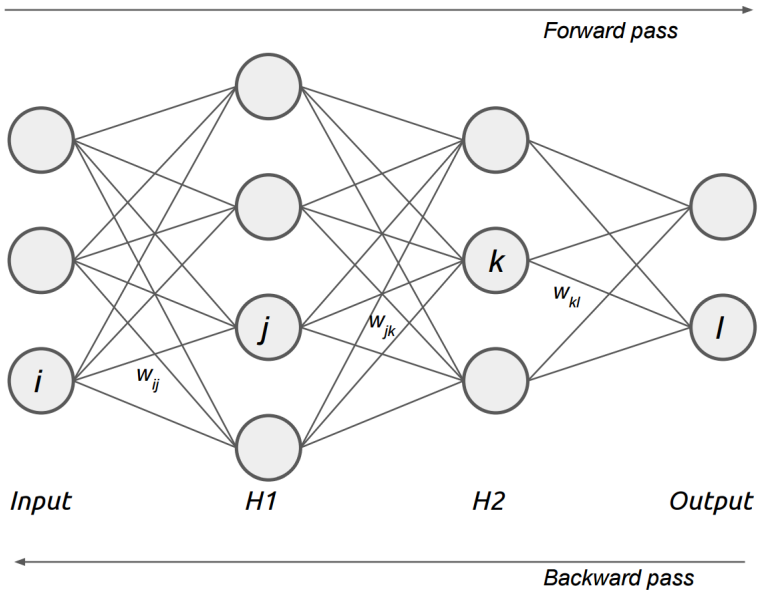


Figure 2.6: Forward and backward pass depicted in a neural network.

$$\frac{\delta E}{\delta y_l} = y_l - t_l \qquad \frac{\delta E}{\delta y_k} = \sum_{l \epsilon out} w_{kl} \frac{\delta E}{\delta z_l} \qquad \frac{\delta E}{\delta y_j} = \sum_{k \epsilon H2} w_{jk} \frac{\delta E}{\delta z_k}$$

$$\frac{\delta E}{\delta z_l} = \frac{\delta E}{\delta y_l} \frac{\delta y_l}{\delta z_l} \qquad \frac{\delta E}{\delta z_k} = \frac{\delta E}{\delta y_k} \frac{\delta y_k}{\delta z_k} \qquad \frac{\delta E}{\delta z_j} = \frac{\delta E}{\delta y_j} \frac{\delta y_j}{\delta z_j}$$

Whether used for training of a fully connected network or a convolutional neural network, the same backpropagation principal can be applied. The first convolutional neural network trained with the backpropagation algorithm using low resolution images of handwritten digits was presented in [LCDH+90].

2.5.3 Convolutional Neural Networks

One of the most popular neural network architectures is the convolutional neural network (CNN), especially when dealing with data in a known, grid-like form, such as images or videos. CNNs are neural network that use the convolution operation at least once. Figure 2.7 shows an early example of such a structure. These types of networks became more complex in recent years, both in terms of the number of parameters and layers involved and because of the number and variations of the basic components. However, most currently used CNNs rely on the convolution layer that usually has the following stages (following [LKF10]):

- Convolutions
- Activation functions
- Normalization
- Pooling or subsampling

In this section the above main building blocks of the convolution layer are discussed in more detail, which also includes their role, main variations and recent development in the field. At the end of this section some additional elements of the network are briefly discussed.

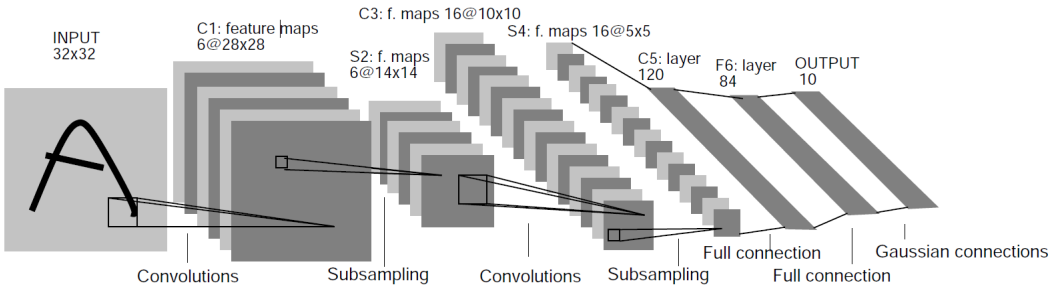


Figure 2.7: An early example of a successful convolutional neural network, which was used for handwritten digits recognition. Figure taken from [LBBH98].

Convolutions

The convolution operation is the key element of the CNN architecture. The basic discrete form of convolution on two functions can be defined in the following way:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.6}$$

where x and w are two functions which are defined on integer t.

The CNN is mostly used on images or other 2-D matrices (even though 1-D convolution for audio and 3-D convolution for video also are often used), which means that the convolution operation can be two dimensional and be defined as follows:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m, j-n) \tag{2.7}$$

where (in the context of the convolutional network definitions) I denotes the input (in this case a two-dimensional image or matrix), K the two-dimensional kernel and the resulting output S the feature map.

Figure 2.8 shows an illustrative example of the convolution process when applied to two-dimensional arrays. This example shows the use of a 2×2 kernel when applied to an 4×3 input array. Assuming that no image padding is used (i.e. convolution is done only when the kernel is positioned totally within the image) and that the kernel is moved by a single "pixel" after each operation, the output presents a complete output of the 2-D convolution.

Activation functions

The activation function is one of the key elements in the neural network structure. It is due to the nonlinear activation function that the network is able to adapt and

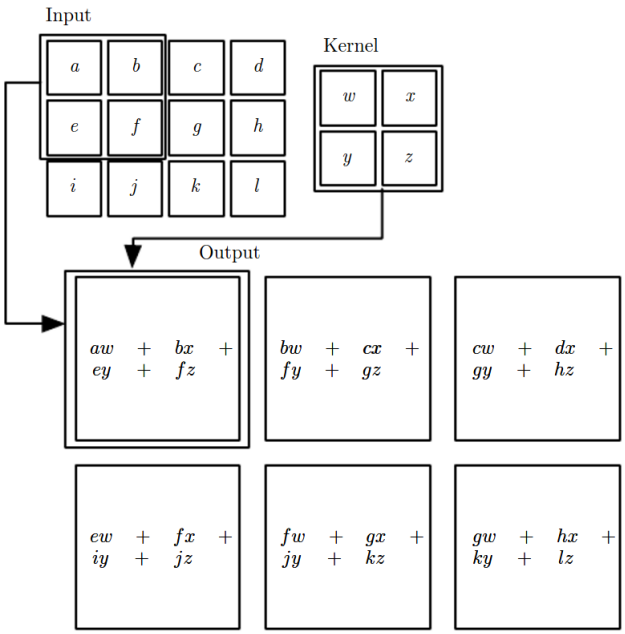


Figure 2.8: A simple example of 2-D convolution. Figure taken from [GBC16].

solve nontrivial classification problems. The most widely used activation function for neural networks is a sigmoid or a similar nonlinear function, the standard for the multilayer perceptron. It usually takes the form of $f(v) = (1 - e^{-v})^{-1}$ or $f(v) = \tanh(v)$, where $v = \sum_{i=0}^{m} w_i x_i + b$.

As an alternative, rectified linear unit (ReLU) was proposed in [NH10] and is defined simply as $f(x) = \max(0, x)$. There are two main advantages of using the ReLU function over the more traditional sigmoid. First, ReLU seems to be better at propagating the gradient and avoiding the vanishing gradient problem, which becomes more severe as the network gets deeper. Also, a faster supervised deep neural network training when ReLU is used was shown in [GBB11]. An additional point is that the unsupervised pre-training that was necessary when training deep nets with sigmoid activation functions (which usually was applied to one layer at a time like in [HOT06]) can now be avoided. This may be attributed to the lack of saturation and the linear response. Second, because all negative values are set to 0, the use of ReLU introduces a lot of sparsity, which can improve the learning process and was shown to have a stronger theoretical foundation [SLJ+14].

However, a potential drawback is the "dying ReLU" problem [MHN13], which is connected to the 0 output for all negative inputs. A large gradient may update the weights in such a way that a neuron with ReLU may never activate for any training data point. Because of the resulting zero gradient, this situation becomes irreversible. To address this shortcoming several alternatives were proposed. Leaky ReLU proposed in [MHN13] introduces a small negative slope to avoid these 0 values. The choice of the negative slope value may seem quite arbitrary. An alternative is to choose it randomly from a predefined range as it was proposed in [XWCL15], which introduces the Randomized Leaky ReLU (RReLU). The visualization of the ReLU unit and its two variations can be seen

in Figure 2.9.

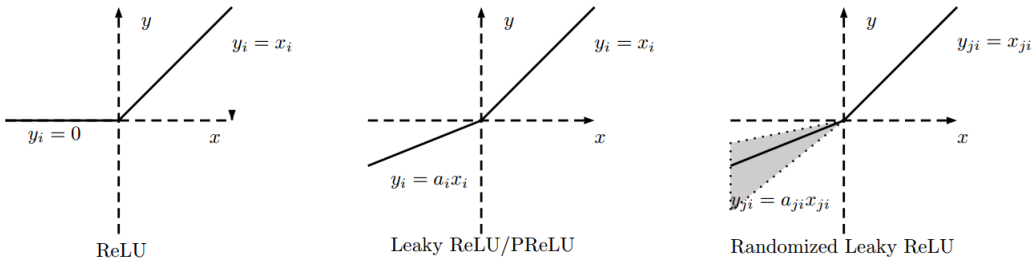


Figure 2.9: Different variations of ReLU. Figure taken from [XWCL15].

Recently, some further developments and alternatives for ReLU were introduced. The Parametric ReLU (PReLU) [HZRS15b] tries to eliminate the need of setting the negative slope values by defining them as parameters that can be learned at the training stage. Another alternative is the Exponential Linear Unit (ELU) [CUH15] where the negative values are present, but they get saturated contrary to PReLU and Leaky ReLU. The inclusion of negative values seems to speed up the learning by reducing the mean unit activations (similar to batch normalization) because of a reduced bias shift effect.

Finally, the Maxout method [GWFM+13] can be considered as an alternative not directly related to ReLU. It consists of two learnable linear functions that can approximate or imitate a range of possible activation functions (ReLU, absolute value, quadratic activation, etc.). Maxout is more flexible than other functions described here, however this comes with a higher computational cost due to the presence of additional parameters.

A systematic and objective comparison between different activation functions is often difficult, especially for bigger dataset due to computational costs. However, one such study addressing this issue was presented in [MSM16]. There, the tests were made on the ImageNet dataset using a modified CaffeNet model (similar to AlexNet, see Section 2.5.5 for details). The modifications were made mainly to reduce the training time and include: a smaller input image (128×128 pixels), the size of the fully connected layers is reduced by half, which results in 2048 neurons per layer. Given this setup, different activation functions were tested. The result for each method along with their respective formulas are given in Table 2.1. More details concerning the training setup and preprocessing (which is constant for each test) can be found in [MSM16].

Based on the results presented in Table 2.1, all the methods based and including ReLU outperform the more standard tanh function. Surprisingly, the total lack of a nonlinear activation function gives a satisfying results, not much worse than the tanh function. Maxout gives the highest accuracy, but it is also the most computationally complex method.

Normalization

Local Response Normalization is usually used as an additional step that can help with generalization when the ReLU activation function is present, as it was

Name	Formula	Score
Linear	$y = x$	38.9
Tanh	$y = \frac{e^{2x}-1}{e^{2x}+1}$	40.1
ReLU	$y = \max(x, 0)$	47.1
Very Leaky ReLU	$y = \max(x, \alpha x), \alpha \in (0.1, 0.5)$	46.9
RReLU	$y = \max(x, \alpha x), \alpha = \text{random}(0.1, 0.5)$	47.8
PReLU	$y = \max(x, \alpha x), \alpha$ is learned	48.5
ELU	$y = x$ if $x \geq 0$, else $\alpha(e^x - 1)$	48.8
Maxout	$y = \max(W_1 x + b_1, W_2 x + b_2)$	51.7

Table 2.1: Different popular activation functions and their score (top-1 accuracy) obtained on the ImageNet dataset with the model and configuration presented in [MSM16].

observed in [KSH12]. This normalization is applied after ReLU and consists of normalizing the convolutional layer outputs across adjacent feature maps (even though the order of the feature maps is arbitrary).

A more recent development is the introduction of the Batch Normalization [IS15], which helps with avoiding bad network initialization. It is usually applied after the fully connected (or convolutional) layer to each training mini-batch, but before the activation function. It allows the use of higher learning rates, which can speed up training considerably. By applying this normalization approach after each layer (and not just at the very beginning as a part of the preprocessing step) the gradual shift from the zero mean and unit variance (also known as the internal covariate shift) in deep nets can be avoided.

Pooling or subsampling

There are two main reasons to use pooling or other forms of subsampling, like striding. It helps to gain a level of invariance and also serves as a way to reduce the feature map size. The pooling layer is usually applied right after the convolution. The most popular pooling methods use either the average or the maximum value as the output. Recently, however some additional pooling methods were proposed. This includes the stochastic pooling [ZF13] and a pooling presented in [LGT16] that is a sum of maximum and average response (max + average). Table 2.2 presents the formulas and accuracy scores for different pooling methods. The experimental setup is the same as in Section 2.5.3 and more details (including the values of different hyperparameters) can be found in [MSM16].

In that experiment the best pooling results were obtained by taking the maximum value or the combination of the maximum and the average value. The alternative subsampling method, namely the stride, has a surprisingly good performance, comparable to max pooling. In this case, the stride is applied during convolution by skipping a certain number of pixels. Indeed, this approach was

Name	Formula	Score
max	$y = \max_{i,j=1}^{h,w} x_{i,j}$	47.1
average	$y = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} x_{i,j}$	43.5
max + average	$y = \max_{i,j=1}^{h,w} x_{i,j} + \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} x_{i,j}$	48.3
stochastic	$y = x_{i,j}$ with probability $\frac{x_{i,j}}{\sum_{i=1}^{h} \sum_{j=1}^{w} x_{i,j}}$	43.8
stride in convolution	–	47.2

Table 2.2: Different subsampling methods and their score (top-1 accuracy) obtained on the ImageNet dataset with the model and configuration presented in [MSM16].

used in [SDBR14] where a CNN structure exclusively composed of convolutional layers (no pooling after convolution and no fully connected layer) was presented and can be considered as a push towards a more simplified architecture (compared to AlexNet) while still maintaining the state-of-the-art performance.

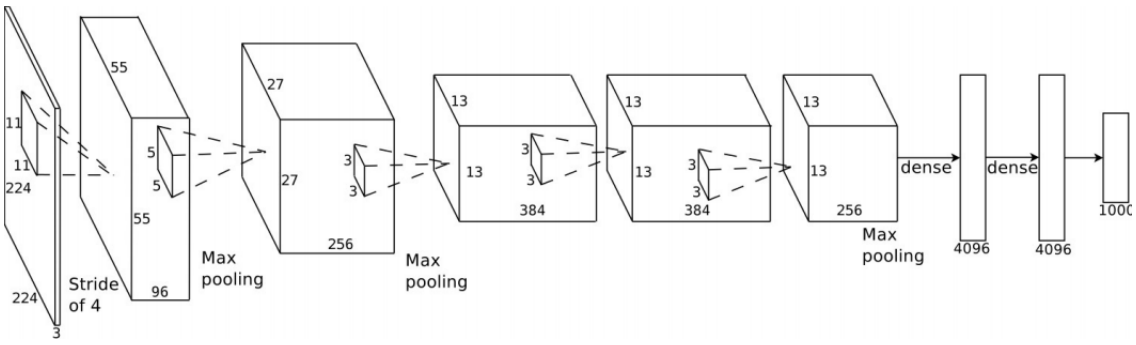


Figure 2.10: The deep convolutional neural network architecture proposed in [KSH12].

Other

This section presents some other elements and strategies that are often used to enhance the overall performance or speed up convergence of a CNN.

- *Dropout* [SHK+14] – is a regularization technique that helps to avoid overfitting. It turns off random neurons and their connections during training, which helps to prevent co-adaptation.
- *Weight initialization* – usually the weights and biases in a network are initialized with random values (often with Gaussian noise with zero mean and 0.01 standard deviation and bias equal to 1, as in [KSH12]). However, this

approach was shown to be problematic when training very deep networks [SZ14]. Some alternatives initialization schemes were proposed, which include:

- *Xavier* [GB10] – estimates the value of the standard deviation based on the number of input and output channels in a layer, assuming no non-linearity between layers.
- *MSRA* [HZRS15b] – extends the above method to include the ReLU non-linearity as a valid assumption.
- *LSUV* [MM15] – starts with setting weights in each layer with orthonormal matrices and then normalizes the variance of their outputs to be equal to 1.

2.5.4 Multi-label CNN and localization

Most CNN models are designed to give the most probable class for a given image. In such cases, during training it is assumed that the concepts are somewhat mutually exclusive. In order to be able to obtain multiple labels per image several approaches were proposed. One of them is presented in [WXH+14]. Here, a single label model is used as a shared network. The input image is divided into subsegments containing potential concepts. Each segment is then passed through the shared network. The outputs from all the segments are finally fused. The structure of this approach can be seen in Figure 2.11.

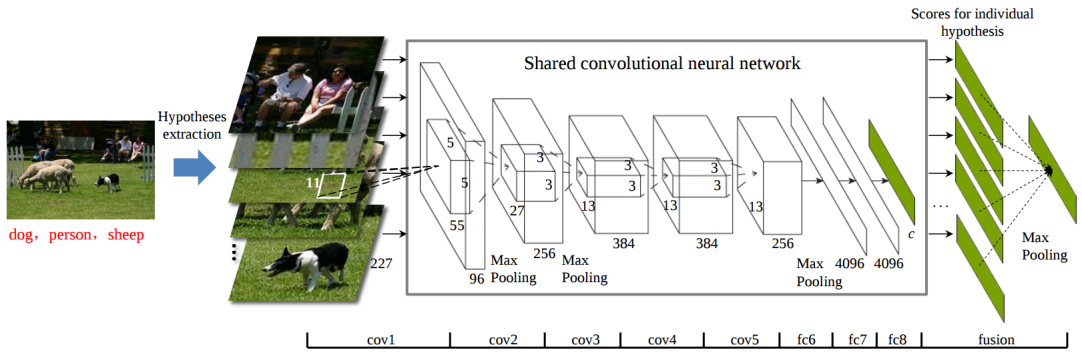


Figure 2.11: The network architecture proposed in [WXH+14] for dealing with multi-label images.

This method is very similar to a range of algorithms used for concept localization in images. There, the coordinates of a bounding box containing a given concept are also given as output. The most prominent examples of this type of approaches are the Region-based Convolutional Neural Network (RCNN) [GDDM14] and its more recent improvements: Fast-RCNN presented in [Gir15] and Faster-RCNN introduced in [RHGS15]. Figure 2.12 shows the overview of the RCNN architecture. In it, the input image is divided into around 2000 proposed sub-regions (or bounding boxes). Afterwards, each region is fed through

the CNN to extract the features, which are then used to determine the concepts present in the region with the help of linear SVMs.

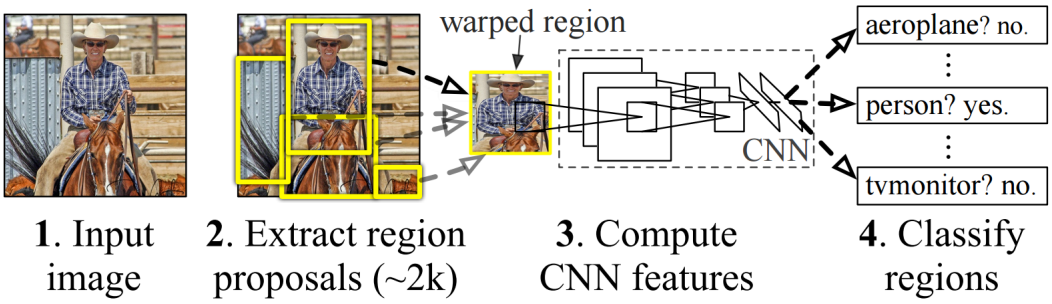


Figure 2.12: The RCNN architecture as it was presented in [GDDM14].

One of the major setbacks of the RCNN approach is the need to generate candidate sub-regions from every image and then passing each one through the network to generate the necessary features. When dealing with large image datasets containing millions of images this approach can become too computationally expensive. Fast-RCNN is one of the attempts to address this problem, namely it reduces the computation time and space that is required. Also it needs only one stage of training. It takes as input the image and the region propositions and for each region it outputs the concept and the refined position of the bounding box.

The Faster-RCNN is another improvement over the initial RCNN approach and an extension of the Fast-RCNN method. Its major contribution is the introduction of the Region Proposal Network, which is able to generate region proposals while processing the image. This removes the need to generate region proposals beforehand (which often is the most time consuming step of the approach) as was the case with both the RCNN and the Fast-RCNN methods.

2.5.5 State-of-the-art CNN architectures for vision

GoogLeNet [SLJ+14] is one of the currently leading architectures, when it comes to CNN systems used for image classification. It was partially inspired by a more theoretical work on sparse deep network presented in [ABGM13]. The main conclusion in that is that, under very strict conditions, it is possible to construct an optimal network topology in a layer by layer manner by taking the correlation statistics of the previous layer activations and clustering neurons, which outputs are correlated. This can only be achieved if the probability distribution of a given dataset can be accurately represented by a large and sparse neural network. This result seems to correspond to the Hebbian theory [Heb05] established in neuroscience, which claims that "the neurons that fire together, wire together". With this observation in mind, the proposed design sets out to find local clusters of units, which would correspond to a local region in the input image. These clusters can be more spread out, creating a need for varying filter sizes. The result of this investigation is a structure shown in Figure 2.13, which is used repeatedly as the building block of the final architecture.

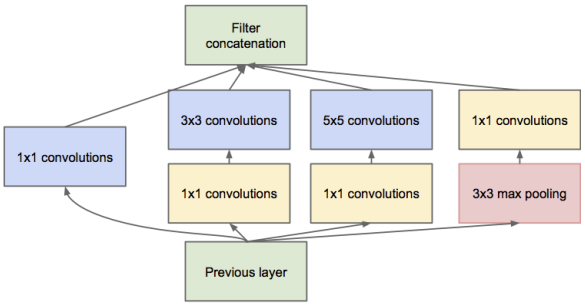


Figure 2.13: The Inception module – the main building block of the GoogLeNet architecture. Figure taken from [SLJ+14].

Another very successful CNN architecture was presented in [SZ14] and it's popularly denoted as VGG net. This approach tries to improve the original structure given in [KSH12] (which also can be seen in Figure 2.10) by exploring one of the most important design aspects – its depth. The size of the convolution filter (receptive field) was set to 3×3 and it was the same for every convolution layer. This significant reduction in size (compared to for example 7×7 in GoogLeNet and 11×11 in Alex net) made it possible to add additional layers while avoiding prohibitive training and execution times. Several different depths are tested ranging from 11 to 19 weight layers (not including pooling layers). Each architecture has 5 max pooling layers. The initial convolution layer has 64 outputs (channels) and this is doubled after every pooling layer up to the final 512.

The Residual Network (ResNet), introduced in [HZRS15a], is one of the most recent advances when it comes to convolutional neural networks. This approach tries to address one of the major issues of CNNs, namely the difficulty of training deeper architectures. It seems that deeper networks suffer not only from the vanishing or exploding gradients (a problem that was already addressed to a degree, see Section 2.5.3), but also from a degradation problem. As the network gets deeper, the accuracy is getting saturated and can even degrade. Additionally, this situation is not related to overfitting. Therefore, the problem may be connected to the way the networks are optimized, i.e. the approach that successfully trains shallow networks may not be the best at training deeper ones. The ResNet algorithm tries to address this problem by using "shortcut" connections (as can be seen in Figure 2.14). These connections usually skip several layers. In this case, their output is added to the output of the stacked layers. The intuition behind this approach is that it is easier to optimize just the residual mapping rather than the complete desired mapping. This approach is inspired by certain classical image descriptors that encode residual vectors in reference to a dictionary (for example VLAD, see Section 2.2.2).

The ResNet approach enables successful training of networks with unprecedented number of layers with a substantial increase in accuracy. Several versions of the ResNet architecture were presented, ranging from 50 to 152 parameter layers.

All of the above architectures were trained and tested on the ImageNet dataset,

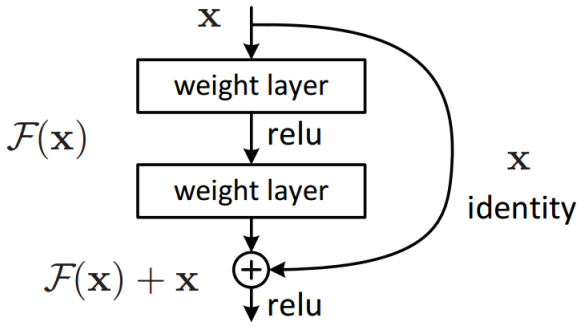


Figure 2.14: The residual learning module. Figure taken from [HZRS15a].

which established itself as one of the most representative and comprehensive datasets available in computer vision. Table 2.3 presents the results of the recent CNN architectures obtained on that dataset. The final and official score in the challenge was the top-5 error rate, which also corresponds to the final column in the table and was generated on the test set. All final CNN submissions were based on ensembles. The top-1 for single and ensemble models (as well as top-5 for single) was obtained on the validation set and these particular results are not available in every case.

method	single		ensemble	
	top1	top5	top1	top5[2]
AlexNet	39.0	16.6	36.7	15.3
VGG	25.5	8.0	24.7	7.3
GoogLeNet	-	7.9	-	6.7
ResNet	19.4	4.5	-	3.6

Table 2.3: Error rates (in %) obtained on ImageNet by state-of-the-art CNN methods.

The results presented in the table show a rapid development of the CNN architectures for vision in recent years. The error rate was reduced by a factor of two in just two years, i.e. AlexNet (2012) and GoogLeNet (2014). And another significant drop was achieved during the ILSVRC'15 challenge with ResNet.

2.5.6 CNN as a feature extractor

Aside from the principal use of CNNs as classifiers, which also requires training, they can also be considered as a source of robust features, in the same manner as the engineered feature algorithms described in Section 2.2. One study presented

[2]Results based on the test set, contrary to all other cases, which were obtained on the validation set.

in [RASC14a] uses a CNN (trained on the ImageNet dataset) called OverFeat [SEZ+13], which is similar in its structure to AlexNet, as a feature extractor for deep features. The features are extracted from the first fully connected layer in that network. After a L2 normalization, the features are used together with a linear SVM classifier to produce the final results. A second setup was also proposed. In it some additional augmentation is made by adding cropped and rotated images as well as applying the component-wise power transform.

The overview of the results can be seen in Figure 2.15. The non-CNN state-of-the-art methods (this includes some well known methods such as VLAD descriptors and fusion of other HOG, SIFT, etc.) are compared to the two approaches described above and to a specialized CNN-based solution for a given task (when applicable). These approaches were used on a varied range of different vision-based tasks, some significantly different from the initial training set. All of which are far smaller than the ImageNet dataset and have between around 1.5 to 6.5 thousand images. Despite this, the results indicate that even the simplest CNN approach is comparable and often outperforms the classical state-of-the-art. The augmented version is (with one exception) consistently better than the classical methods.

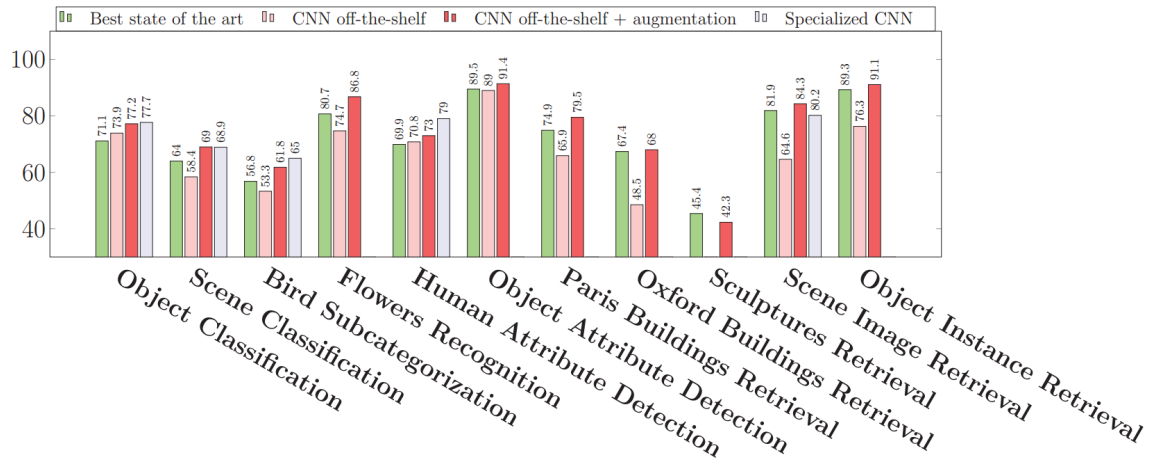


Figure 2.15: The accuracy results comparing the CNN-based features with the non-CNN state-of-the-art on a range of classification and retrieval problems. The CNN augmentation is based on such simple techniques like rotating and cropping. Specialized CNN refers to other work applying CNN-based methods to that particular dataset. Figure taken from [RASC14a].

These results show the strength and robustness of the CNN-based features when dealing with diverse vision tasks. Based on that, a question arises about how general and transferable such features can be. A work presented in [YCBL14] tries to give some insight into that issue. There, an approach is proposed to tell to what degree the features extracted from a given layer are general or task specific. Lower layers usually learn more general features that can be successfully applied to a wide range of tasks. Figure 2.16 gives an example of the features that

are learned in the first two layers of a convolutional network along with images that give the highest response. The first layer features tend to resemble Gabor filters (edge detectors) and color blobs. Based on that, the second layer learns to recognize corners and more complex shapes. Each subsequent convolutional layer in the network detects more and more advanced and specific features.

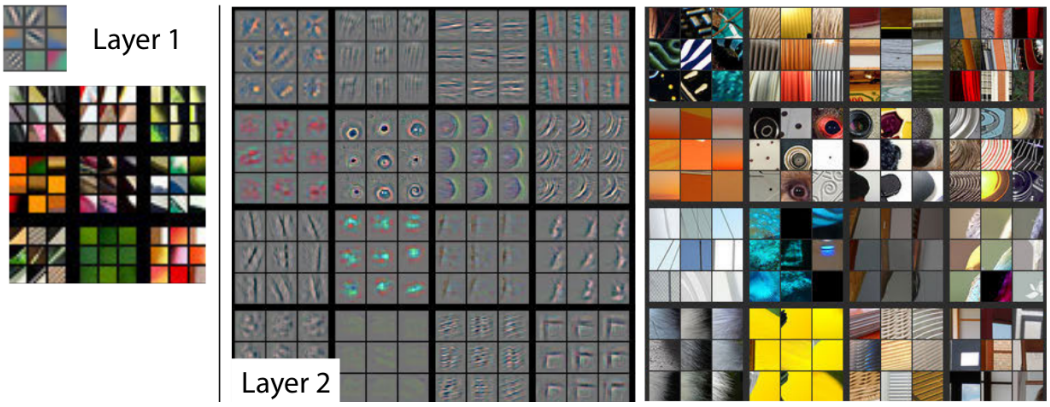


Figure 2.16: Feature visualization from the first two layers of a network. Images taken from [ZF14].

Figure 2.17 presents the main findings from that study. The point at layer 0 is the baseline CNN network, which was trained to classify 500 random classes from the ImageNet dataset. One of the major insights is the existence of co-adaptation, which indicates that there are certain interactions and adaptations between neighboring layers that are lost if the layers are trained separately (as can be seen in curve 2). This co-adaptation seems to be most visible between higher convolutional layers (layers 4 and 5) and between the last convolutional layer and the first fully connected one (layers 5 and 6). Another major observation is the performance drop attributed to the specificity of features from a given layer. It seems that transferring the first 2 layers (and then training the rest) does not affect performance. However, as more layers are transferred (and less trained) the accuracy drops considerably.

In this thesis, some additional experiments are done to give further insight into the transferability of pretrained layers and fine tuning (especially from ImageNet to TRECVid). Additionally, based on the performance of the off-the-shelf CNN models compared to the more classical state-of-the-art, different types of fusion between the two are explored.

2.5.7 CNN application to audio

The application of the CNN algorithm is not limited to images or video. Its growing use can also be observed in speech and audio. In that case there are two main approach that are available. Either using a 1D convolution on a raw audio signal or a 2D convolution on a 2 dimensional representation of sound, namely a spectrogram, which represents the audio in terms of time and frequency.

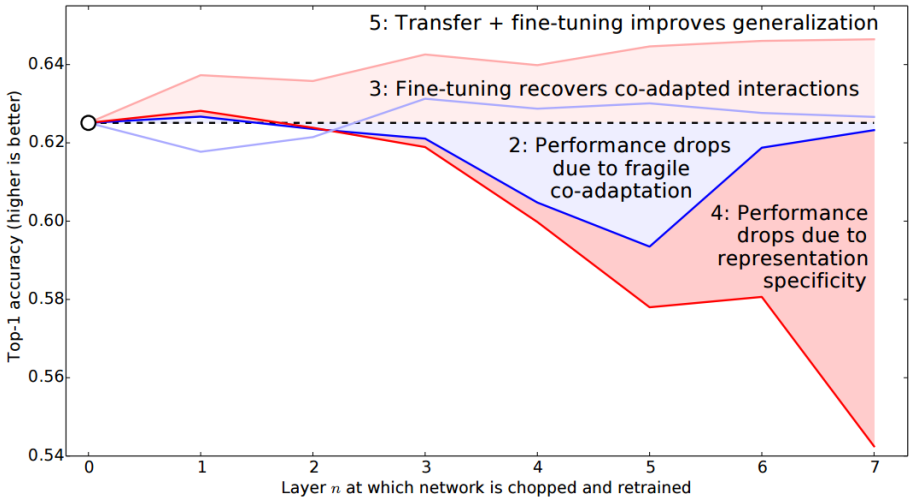


Figure 2.17: The main findings of the study [YCBL14].

In [PC+15] a speech recognition system is presented that uses raw speech as input, which is then processed by a 1D convolutional layer. A lot of neural networks (usually fully connected) used in speech recognition take as input a well established set of features such as Mel-frequency cepstral coefficients (MFCC). Here, however, the CNN is tasked to learn appropriate features from the raw signal alone.

The use of CNN with spectrograms was also explored previously: in [DAHY13] a system dealing with phonetic confusion is presented, however convolution is done only along the frequency axis. In order to address the problem of phonetic confusion in speech, a new pooling strategy is proposed where different pooling sizes are used depending on the trade-off between invariance to frequency shift and the phonetic confusion.

An interesting approach was presented in [LPLN09] where an unsupervised approach to feature learning is proposed. The algorithm is based on a convolutional deep belief network, which is applied to a set of unlabeled audio data, including speech and music. The extracted features seem to outperform the MFCC baseline on a set of speech recognition related tasks such as speaker identification and gender classification. This is also one of the first studies that applies deep learning in this context. Apart from that, Recurrent Neural Networks (RNN) can also be successfully used with speech spectrograms, as it is suggested in [HCC+14] for automatic speech recognition.

Some attempts were also made to use CNNs in noisy conditions. A recent study [MLSF14] uses 1D convolutions on filter banks. Surrounding frames are taken into account and serve as context to reduce noise impact as can be seen in Figure 2.18 where the convolution operation in this network is shown. In this study the CNN is used as an alternative to the more traditional universal background model. The output of the CNN is used as a basis to create i-vector models.

A study presented in [AV] investigate the problem of emotion recognition by combination of audio and visual features. Spectrograms were used as input

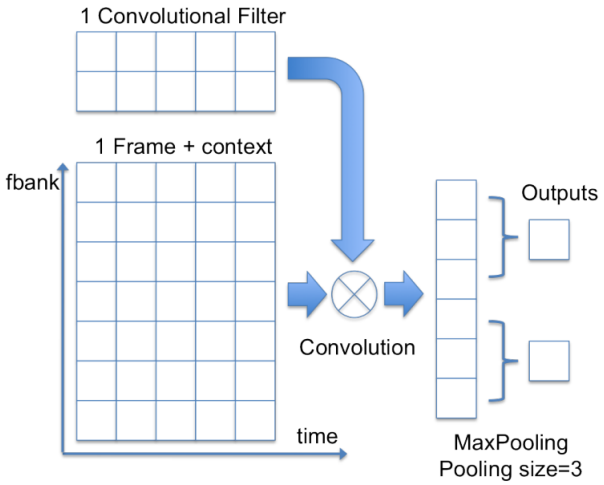


Figure 2.18: A 1D convolution taking into account the current and neighboring frames (as context) applied to a speech signal. Figure taken from [MLSF14].

and 1D (convolution only over the time axis) and 2D (both time and frequency) CNNs were evaluated. The final decision, about which emotion is present in a video fragment, is made using a LSTM model.

Speaker and language recognition using neural networks was also presented in [RRD15a]. Most deep neural network approaches used in this domain concentrate on the use of DNN as the source of bottleneck features. Alternatively, the DNN posteriors can replace the GMM model and serve as the base for i-vector extraction. Also, this paper is one of the few that explores how transferable are the DNN models trained on a given task and applied to another (for example: DNN trained on automatic speech recognition and used for language and speaker recognition).

In [GHT+14], CNNs were used for the language identification task. This work presents a good example of the use of the CNN combined with a fully connected deep neural net as a source of bottleneck features. This is an alternative to the work presented previously where only a fully connected network is used to extract these features. These features are then used to extract i-vector models, which are projected with a MLP and afterwards serve as a training set for language specific SVMs with polynomial kernels.

An approach which tries to identify disguised voices is shown in [UW15]. A CNN is trained using spectrograms in order to identify people imitating different voices. This study considered a relatively small number of speakers with the explicit goal of identifying fraudulent behavior. It also uses one of the well known CNN architectures (AlexNet [KSH12]) to achieve this goal.

Chapter 5 presents an approach to speaker recognition based on a CNN and spectrograms. Contrary to the studies presented in this section, this approach uses the CNN both as a source of features and a classifier. Additionally, a set of fusion methods with the non-DNN based state-of-the-art approaches for speaker identification is tested.

2.5.8 Recurrent neural networks

So far, the contents of this chapter concentrated on feedforward neural networks, i.e. architectures where information travels in a single direction: from the input nodes, through the hidden layers and up to the output layer. After the resurgence of the CNN architecture, the Recurrent Neural Networks (RNN) also became more popular and more widely used. Their structure enables the formation of direct cycles and allows for dynamic and temporal behavior. This is done by taking into account past inputs (or the resulting internal states). Although the RNN-based methods are usually used in such domains as machine translation and language modeling, more recent advances also show great potential in computer vision, which includes video classification [DAHG+15], image captioning [VTBE15] and more.

One of the most interesting RNN architectures is the Long Short-Term Memory (LSTM) [HS97]. Its detailed structure can be seen in Figure 2.19. Compared to the standard recurrent neural network, the LSTM is able to classify and predict time series with very long and unspecified periods between events. Due to the presence of gates, the LSTM is able to remember certain input values over the course of many time steps and forget them if needed.

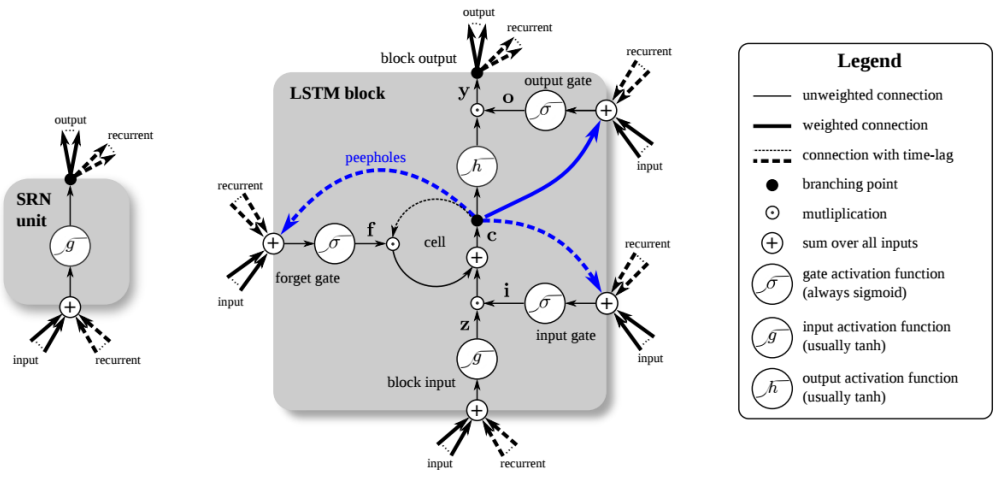


Figure 2.19: The structure of a simple recurrent network compared to the LSTM block. Image taken from [GSK+15].

The RNNs can be trained using gradient descent, in a similar manner as the standard feedforward networks. In order to be able to train on sequences of inputs, the backpropagation through time [Moz89] approach is used, which unfolds the network in time so that the input and hidden layers are replicated for every element of the input sequence. The standard RNN algorithm often suffers from the vanishing gradient problem if time periods between events are too big. The LSTM algorithm, on the other hand, can avoid this issue through the use of the memory block and appropriate gates.

2.6 Active learning

Active learning [Set09] is a subsection of machine learning and represents a family of algorithms that can help train a machine learning model more efficiently. The main assumption is that a lot of the data that could be potentially used for training is redundant. Therefore, it is possible to train a model with a comparative or even better performance, but with the use of a significantly smaller amount of data. This can be done by allowing the learning algorithm to choose the data, which then will be used for its training. Another advantage and application is when most of the data comes initially unlabeled and manual annotation is required. Nowadays for most applications, there is an easy access to abundant amounts of data. The Internet and the easy and cheap access to multimedia recording devices such as cameras can serve as primary examples. The main drawback is that most (if not all) of the data is not initially annotated, which in turn makes its use for model training somewhat limited. Most of the time, in order to be able to train accurate models, manual human annotation is required. However, this is usually very expensive and time consuming. An active learning algorithm can help to choose which data samples are the most useful.

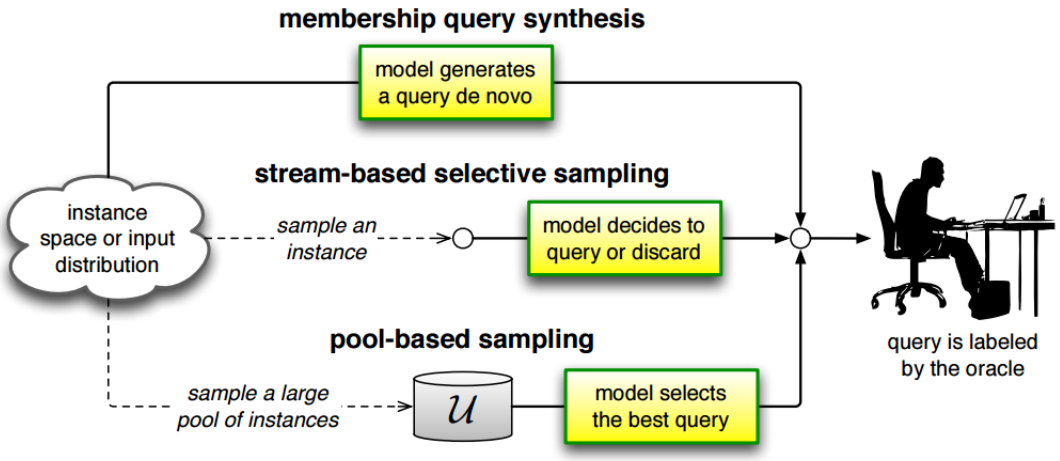


Figure 2.20: Different scenarios involving active learning. Figure taken from [Set09].

2.6.1 Active learning scenarios

Several scenarios involving active learning are usually considered. They usually take into account different ways the data is presented to the learner. The three main ones are depicted in Figure 2.20. They assume that all the queries consist of unlabeled data, which is going to be labeled by the oracle once selected.

- **Membership query synthesis** was first introduced in [Ang88]. In this scenario, the learner asks for the correct annotation of the unlabeled instance for the input space, which in this case is not taken from an underly-

ing natural distribution, but are generated by the learner itself. The main drawback of this approach is that the newly generated instances may lack any clear meaning and therefore be difficult to handle by the human annotators. This issues was first pointed out in [BL92], where this approach was applied to train a neural network to recognize handwritten characters.

- **Stream-based selective sampling** serves as an alternative to the artificially generated queries. This approach is based on selective sampling [ACL90], which expects that providing unlabeled instances for annotation is not expensive or even free. In that case, the instance is taken from the distribution and then the learner can decide if this example needs to be annotated or not. Another feature is that this approach goes through the data sequentially and the decision is made for each single query separately. This approach also assumes that the samples come from the underlining distribution, which mitigates the problem of generated samples present in the membership query synthesis scenario. Within this framework, different techniques may be used for the selection process. Any given sample can be evaluated based on how much new information it provides to the learner [DE95]. It may also belong to a region of the sample space that is ambiguous and problematic to the learner [CAL94] or a region that it still knows not enough about [Mit82].

- **Pool-based sampling** is similar to the previous scenario. The main difference here is that a larger pool of unlabeled data is available from the start and usually it is assumed that its size remains constant. Queries are selected from this pool based on some measure of potential informativeness, which is applied to all instances in the data. This scenario is by far the most common in the literature. It was introduced in [LG94].

2.6.2 Query strategies

Throughout the years there was a growing number of different active learning strategies to select the best samples for annotation. They depend on the criteria, application, the data that needs to be annotated and so on. In this section an overview of the main methods is presented as well as some more recent developments in the field.

Uncertainty sampling was first introduced in [LG94] and is one of the most popular approaches used to this day. The idea is to label the instances that are the least certain according to the active learner. One of the main advantages is the ease of implementation in the case of probabilistic learners. When dealing with a binary classification problem, choosing the instances with posterior probability closest to 0.5 is the most preferable under this query strategy.

When dealing with a classification problem with more than 2 classes there are three main variants of uncertainty sampling:

1. *Least confident* – takes into account only the class with the highest posterior probability and chooses the instance with the lowest value. It can be defined as:

$$x^*_{LC} = \underset{x}{\operatorname{argmax}}\, 1 - P_\theta(\hat{y}|x), \tag{2.8}$$

where $\hat{y} = \underset{y}{\operatorname{argmax}}\, P_\theta(y|x)$ is the class with the highest posterior probability according the the learner θ. The drawback of this approach is that it only takes into account the most likely label while ignoring all the rest.

2. *Margin sampling* – an alternative approach proposed in [SDW01], where the probability of the two most likely classes is used. It is defined in the following way:

$$x^*_M = \underset{x}{\operatorname{argmax}}\, P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x), \tag{2.9}$$

where $\hat{y}_1$ and $\hat{y}_2$ are the two classes with the highest posterior probability. The smaller the margin, the more uncertain a given instance is.

3. *Entropy* – this method uses the entropy as defined in [Sha01] to determine the most uncertain sample:

$$x^*_E = \underset{x}{\operatorname{argmax}} - \sum_{i=1}^{m} P_\theta(y_i|x) \log P_\theta(y_i|x), \tag{2.10}$$

where m is the total number of classes. Entropy (based on its information theory definition) is the expected value of the information contained in a given data point. In general, if one instance has the posterior probability lower than other, it becomes more informative.

Another classical and extensively used approach is **Query-by-committee** (QBC) [SOS92]. This algorithm assumes that there is a group of models (i.e. classifiers or learners), which are all trained on the available labeled set of data L. At the same time, they should output different hypotheses.A level of diversity should be present, in other words for every input not all models should produce the same output. Each model in such a committee would be voting on how to label a given input. The level of disagreement for such an instance can be used to determine a potential candidate for annotation. The most informative queries are the ones that produce the most disagreement.

Numerous ways for introducing diversity in the committee of classifiers were proposed throughout the years. A significant portion of them are based on the research done with classifier ensembles, and so boosting [FS+96] and bagging [Bre96] algorithms were used to create query-by-boosting and query-by-bagging (both introduced in [Mam98]), respectively. Also, a solution based on the partition of the feature space was introduced in [MMK00] and resembles other well known classifier ensemble methods such as Random Forest [Bre01]. There are many ways to measure the level of disagreement between classifiers in a given committee. An example of such a measure is vote entropy (introduced in [DE95]), which can be calculated as follows:

$$x_{VE} = \underset{x}{\operatorname{argmax}} - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i}{C}, \tag{2.11}$$

where y_i represents all possible potential classes/labels, $V(y_i)$ is the number of votes for a given label and C is the size of the committee.

Relevance sampling [TC01] is another popular approach to active learning. This approach was extensively used for image retrieval. The main goal of this approach is to effectively produce accurate classifiers. This is done by minimizing the amount of labeled queries by selecting those that should be most beneficial for the performance of the classifier. To that end, the relevance of a query is taken into account. This approach relies on the properties of the support vector machines (SVM) algorithm. Intuitively speaking, the SVM is a hyperplane that separates the training data (in the simplest case with only two classes) by a maximal margin. The instances from the training set that are closest to that hyperplane are called support vectors. In the simplest form of the relevance sampling, the queries (or unlabeled instances) that are closest to the hyperplane, i.e. potential support vectors, are chosen for annotation.

An extension of the approaches presented above was introduced in [SQ12a]. Its main application is with highly imbalanced datasets, which frequently occur when dealing with real-life data and applications such as multimedia indexing. An imbalanced dataset is characterized by the presence of a significantly over-represented majority class and a small minority class (in the case of the binary classification). The minority class usually contains instances that are more interesting to the potential user (for example a set of patients with a rare disease as opposed to the set of healthy people). Image retrieval is another application where taking dataset imbalance into account is important. Here, the user is usually interested in just a very small and specific subset of images chosen from among thousands of other categories.

This method involves the use of multiple classifiers (SVMs as in the case of relevance sampling). For every individual classifier, it randomly undersamples the majority class so it has more or less the same sample size as the minority class. This new subset is afterwards used for training. It is worth noting that each SVM differs with respect to the samples from the majority class, while the minority class is the same for each one. After the training, the output of each classifier is fused to produce the final prediction. This approach works well with both uncertainty and relevance active learning selection strategies.

2.6.3 Active learning and clustering

In order to increase the overall performance, many active learning algorithms try to take into account the prior data distribution. This usually involves the use of different clustering algorithms that are integrated into the active learning cycle. The clustering is either used at the very beginning as initialization or the cluster structure is constantly adjusted throughout the active learning process.

A good example of the above can be found in [NS04] where an active learning approach is presented that integrates clustering, which helps to select the most

representative samples and to avoid repeated labeling of the samples in the same cluster. The method starts with the initial K-medoid clustering algorithm. This is followed by the estimation of the class label model, which helps to determine the cluster representatives. Afterwards, the selection step takes place where the samples close to the classification boundary and cluster representative are chosen. After the labeling of those data samples takes place, a adjustment of the clustering is made as well as the retraining of the classification algorithm. The whole process repeats for a predefined number of steps or until a certain criterion (like a low enough error on the test set) is met.

Another example connecting active learning with clustering was presented in [VLBM10]. Here, a semi-supervised clustering K-means algorithm is proposed with active learning, which is used to select and label cluster seed to enhance clustering quality. For the seed selection, a min-max approach is proposed, which tries to choose samples that are furthest from the already labeled samples. This ensures a good coverage of labels throughout the dataset.

2.6.4 Unsupervised active learning

So far the active learning approaches were based around a classifier (or a classifier with clustering as it was described in the previous section), which would be updated after every iteration. However, there are also methods which do not rely on the presence of a trainable model.

In [DH08] hierarchical sampling for active learning is presented. The overview of the algorithm can be found in Figure 2.21. The main idea is to start with the initial clustering and, by labeling subsequent data points from these clusters, iteratively refine the cluster structure by querying the most impure clusters. The clusters were all labels belong to the same class are kept, while impure clusters are divided to minimize the impurity (as it is shown in Figure 2.21(d)).

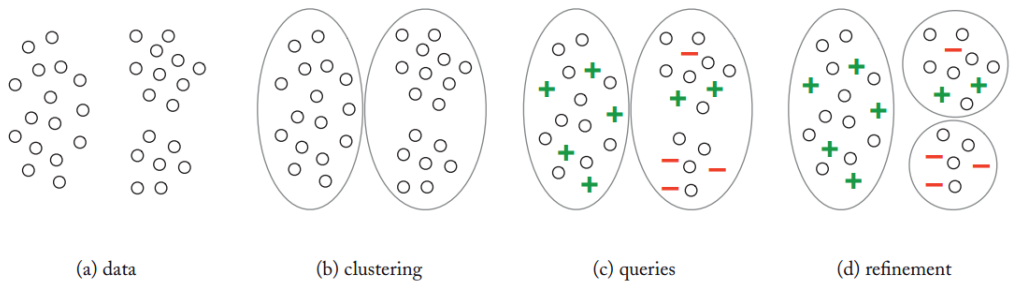


Figure 2.21: The hierarchical sampling algorithm. Figure taken from [Set12].

The use of unsupervised approaches can also help with the detection of anomalies or rare classes in the data as it was shown in [PM04]. The algorithm tries to identify useful anomalies (for example: data points belonging to a very rare class) and at the same time disregard any outliers or noisy data points. This is done using a Gaussian mixture model that is refined by consecutive annotations provided by the human expert.

2.6.5 Label propagation

In recent years, a growing number of work was published that involves the acquisition of supplementary annotation without the involvement of a human annotator. Most of these approaches are based on some sort of label propagation and are heavily dependent on the data, its source and any additional information that can be extracted (such as speech transcription or a script of a TV program). Contrary to the more classical active learning approaches (e.g. presented in the previous section), most of the time there is no learner and the labeling is based on proximity or overlap rather than probability from a learned model.

In [ESZ09] a method for automatic naming of characters in TV shows is presented. It uses different sources of information to automatically generate a time stamped character annotation. As the main source of data, faces appearing in the video are used under the form of face tracks, which can be defined as appearances of an individual character across frames of the video, usually confined to a given shot. In order to get the necessary label to annotate the faces, both the script of a given show is used and the available subtitles. While in a script the dialog is divided among the characters, in the subtitles a given line of text contains a time stamp. By aligning both sources the identity of a character can be matched to its exact appearance on screen.

Another example of label propagation can be found in [BBE+04]. Here, the experiments are done on a collection of news pictures and associated captions taken from the online news service Yahoo News. The names (potential labels) are extracted from the text that comes with the image and not the image itself. The images serve as the basis for face extraction. Figure 2.22 gives the example of the data source for this study. The aim of the paper is to produce good quality clustering when only having access to inaccurate or ambiguous labels. Kernel principal components analysis and linear discriminant analysis are applied to reduce the dimensionality and to project the data into a more discriminative space, respectively. Afterwards, k-means is used to assign ambiguous face to one of the labels. The faces that are far from the mean are removed and the new discriminant coordinates are calculated. Finally, the clusters are merged based on the similarity between faces.

President George W. Bush makes a statement in the Rose Garden while Secretary of **Defense Donald Rumsfeld** looks on, July 23, 2003. Rumsfeld said the United States would release graphic photographs of the dead sons of **Saddam Hussein** to prove they were killed by American troops. Photo by Larry Downing/Reuters

Figure 2.22: An example of a news image and associated caption used in [BBE+04].

The images are preprocessed by using the kernel principal components analysis

and linear discriminant analysis to reduce the dimensionality of the data and project it to a more appropriate space for this task.

Another approach to label propagation was presented in [PMT10]. This work proposes a method for naming faces in videos based on labeled and unlabeled examples. In order to do this an iterative label propagation approach is used on a graph of faces and face-name pairs. In this study the labels are generated from video transcript. The faces are detected using the popular Viola-Jones approach [VJS05]. The label propagation is done using random walk process approach. To improve the performance, an anchor detection element is applied following an approach introduced in [DMP+06].

An approach which also tries to recognize faces based on a set of weak labels is presented in [KWRB11]. As in the study described above, the main source of labels is either the transcript or subtitles (more easily available when dealing with scripted TV series). As the proposed solution a semi-supervised multiple instance learning algorithm is introduced. This enables the use of priors as labels for the unlabeled instances.

A label propagation technique for naming speakers in TV broadcast videos was presented in [PBL+12]. In this work three different propagation approaches are proposed, as shown in Figure 2.23. Based on the notation in that figure the following can be defined: a set of speech tracks $T = \{t_1, \ldots, t_K\}$ of size K, a set of speaker clusters $S = \{s_1, \ldots, s_L\}$ of size L and a set of M names $N = \{n_1, \ldots, n_M\}$.

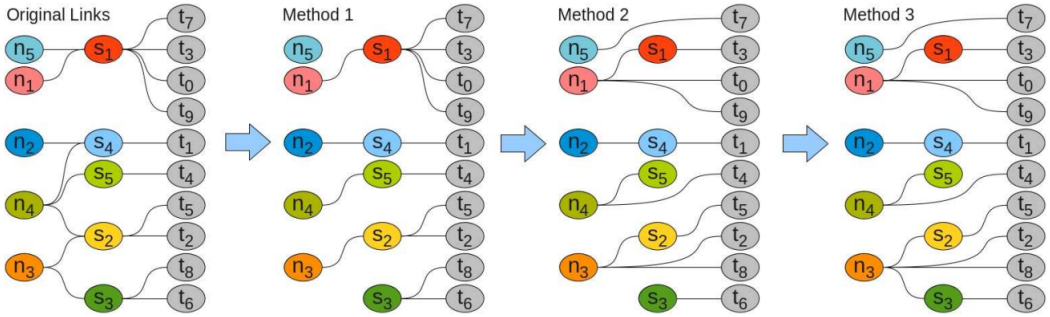


Figure 2.23: The name propagation methods proposed in [PBL+12].

The first method uses a one-to-one mapping between speaker clusters and the names and tries to maximize the co-occurrence (in terms of duration) between the two. It is based on the assumption that speaker diarization results in perfect speaker clustering, i.e. every speech track is correctly assigned to a cluster. The second approach (method 2) assumes that when a name is present on the screen, it usually belongs to the person currently speaking. After labeling every speech track which has a co-occurring name, the rest is treated with the first method. Method 3 takes into account that the output of speaker diarization is not perfect, i.e. the clusters may be over-segmented with more than one belonging to the same speaker. The approach in method 2 is applied first. Afterwards, the remaining speech tracks are labeled cluster-wise, which allows for naming several speaker clusters.

2.6.6 Active learning for multimedia

Throughout the years, AL saw more and more applications to domains such as multimedia and especially videos. This is because video annotation can be prohibitively expensive when applying the same techniques used when labeling single images.

In [SQ12a] an active learning approach for multimedia annotation is proposed. In this work the annotation takes place on the key frame level, i.e. a single image is extracted from every shot. In the context of this work, active learning is used here as a means to efficiently create an automatic indexing system, rather than to annotate a corpus. The experiments are conducted on the TRECVid corpus (see Section 2.8.3 for details) where each image can have multiple concepts at a time. Also, the dataset is heavily imbalanced with most of the concepts appearing in less than 1 % of the corpus. Several different classifiers were tested including SVM and MSVM (both described in detail in Section 2.3) as well as 4 different descriptors (including color histograms and SIFT BoVW). Finally, 4 active learning selection strategies were selected, which includes random sampling, uncertainty sampling and relevance sampling (the last two are described in detail in Section 2.6.2).

The work presented in [AQ08a] describes the collaborative annotation system that was used to generate the annotations for TRECVid 2007 dataset. Contrary to the work presented above, this approach uses active learning to reduce the labeling effort and extract the most useful information from the limited amount of annotation. Three different selection strategies were tested: relevance, uncertainty and random sampling. In order to make the results comparable, a simulated active learning scenario is used where a previously annotated dataset has its labels hidden from the active learning algorithm. The label for a particular instance is revealed once the algorithm selects it. This is the most common approach when evaluating active learning strategies and it is used for near to all experiments in Chapter 3.

2.6.7 Practical application challenges

Even though active learning was for years and still remains an active area of research, the number of published papers where it is used in practice is still somewhat limited. One study [Set11] tries to investigate and highlight the additional difficulties and considerations that arise when active learning is applied to real world applications. Six such challenges are mentioned and discussed:

1. *Querying in batches as opposed to single queries* – most active learning approaches usually select one query at the time. It's not always feasible to do it in practice, however. If a model needs to be retrained or updated after every single new annotation and at the same time is quite complex (e.g., ensemble algorithms, other state-of-the-art methods), then there is a risk that the annotator would be required to idly wait. By presenting several queries to be labeled at once this issue can be avoided or at least minimized. However, this pool-based active learning requires some additional considerations like making sure that the set to be annotated is diverse enough (so

that almost identical instances are not chosen) among others.

2. *The presence of noisy oracles* – one of the assumptions, especially when dealing with a simulated active learning scenario, is that the labels received from the annotators are always correct and noiseless. The issues like fatigue or carelessness are usually not taken into account. A common source of errors is the involvement of non-experts (local volunteers or through Internet-based crowd sourcing), which requires a cleaning or a verification mechanism to correct potential label mistakes.

3. *Varying or unknown labeling costs* – some concepts may be harder or take longer to annotate. Others can have a varying cost that is hard to predict. Because of that, the reduction of the total number of annotations required (one of the main goals of active learning) may not lead to a reduction in time, and therefore in overall cost. The labeling cost may also vary depending on the annotator: people with domain knowledge versus laymen may take a different amount of time to make a decision. In some cases when the labeling cost is completely ignored, a given active learning strategy may not perform significantly better than random selection [ANR09].

4. *Alternative query types* – the most common query type is the membership query where one must decide which class a given instance belongs to. However, other query types exist such as multiple-instance active learning [SCR08] or feature querying. In the case of the latter (first introduced in [RMJ06]), features are proposed to the annotators who in turn can judge their usefulness.

5. *Multi-task active learning* – this is the case when a instance can be labeled differently for multiple tasks. When selecting such labels, a measure of informativeness across tasks should be considered. This could lead to trade-offs between good performance for a particular task versus all the possible tasks.

6. *Shifting model classes* – most active learning algorithms rely on a learner to propose new instances for annotation. When a learner model is changed (for example from a SVM to a neural network) will the already annotated training data be appropriate for the new model (i.e. give better results than the training data generated by random sampling)?

In this thesis the active learning framework was tested in practice. Therefore, some of the above challenges were investigated. For the remaining part of this section a short discussion is presented on how the above points relate to the work done in this thesis. For the first point, the computational constraint (of having an online annotation system) was addressed by introducing a batch of queries to the annotator. Also, many approaches were tested to increase the diversity of each batch, including: taking instances from different videos, querying the unlabeled clusters or outliers and so on.

The presence of noise in labels (point 2) was also considered. In this work two sources of noise can be identified: the annotators and the label propagation. A suggested approach would be to use one source of labels as verification for the other. Additionally, each instance should be labeled twice (three times if there is a conflict). Due to the nature of the annotations (identifying people in videos) the cost of annotation was assumed to be constant across all classes. To some extent this is because of the heavy imbalance between concepts (identities) with some containing only one instance, which makes it complicated to have an accurate cost estimation.

The multimodal corpus used in this thesis allows for multi-task active learning. On the one hand, the same annotation can be used for many different tasks, e.g. face recognition or speaker identification (often a speaking person is also visible). On the other hand, because the people in the video need to be named, some additional information can also be acquired, such as gender or role in the video (guest, anchor, etc.).

2.7 Active and deep learning

Very recently there has been some research emerging, which tries to combine both active and deep learning. In [YZS+15] an approach is proposed, which uses both a deep neural network and a SVM classifier. During the active learning session (this cycle is shown in Figure 2.24), which involves participation of the human annotators, only the SVM model is updated and the CNN is used as a feature extractor. The use of a CNN model directly in such context remains problematic due to the retraining time.

After the image annotations are finished, the CNN is retrained with the combination of both the initial data and the newly labeled one. Not surprisingly, the newly obtained model has better performance than the one trained on the original data.

2.8 Datasets

In this section, a short overview of the principal datasets used in this work is presented. The aim here is to highlight their main features, scope and differences. Most of the datasets have some form of multimodality, which includes images, motion, audio and even text. More details will be provided in the subsequent chapters were the particular datasets will be actually used.

2.8.1 Pascal VOC

The Pascal Visual Object Classes (VOC) [EEVG+15a] is a dataset containing images and their annotations. Additionally, up to 2012 an annual competition was organized using this set. The principal tasks include classification, localization and segmentation. The size of the dataset was not fixed and changed every year. On its final release in 2012, the training and validation set contain 11 530

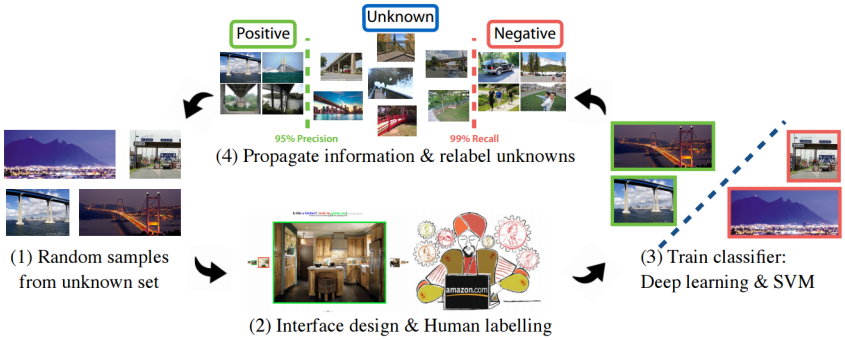


Figure 2.24: The active learning cycle presented in [YZS+15]. (1) Some random images are chosen for annotation. (2) They are then labeled using the proposed interface and the Amazon Mechanical Turk service. (3) A binary SVM classifier is trained next, using the features extracted from the deep net. (4) Finally, the classifier is used to process the unlabeled set of images. The most ambiguous images are chosen for annotation in the next iteration.

annotated images with 27 450 regions of interest and almost 7 000 segmentations. All of this was divided into 20 categories, including persons, different kind of animals, vehicles and indoor objects. The ImageNet dataset may be considered its successor.

2.8.2 ImageNet

The ImageNet dataset [DDS+09] is currently one of the most popular sets of data that focuses on image classification. The training set used in the official challenge consists of 1.2 million images divided into 1000 different categories. It represents a diverse set of categories ranging from man-made objects (*teapot*, *printer*, etc.) to different kinds of animals (*bullfrog*, *goose*), often very challenging like different breeds of dogs. The images were obtained with either the use of Flickr or several search engines and were annotated by hand. This dataset enables the evaluation of both classification and localization tasks. Due to its scope, size and quality, the ImageNet dataset has become the most popular benchmark for new image classification algorithms, especially Deep Learning. It is also considered as one of the main contributing factors in the resurgence of the Deep Learning based methods in recent years.

2.8.3 TRECVid SIN

Semantic indexing is one of the tasks that is a part of the TREC Video Retrieval Evaluation (or TRECVid) [OFS+14]. The goal of this task is to propose semantic tags to various video segments in an automatic manner. The dataset consists of 800 000 videos with the total running time of around 400 hours. Overall, there are 500 different concepts, which were initially selected for this task. This includes a whole variety of different objects (such as *airplane*, *bus*, *dog*) and more complex scenes (*demonstration*, *classroom*, etc.). Out of that, 346 are chosen (because they

have at least 4 annotated positive samples) for the full task. The final evaluation is made on a subset of 60 concepts.

2.8.4 REPERE

The REPERE dataset [GMVS12] consists of a wide range of TV broadcast videos, including talk-shows, news and parliamentary debates. The main goal is to identify different persons present in the videos. In this context, a person can be described by the face, the voice or the written name that appears on the screen. There are 7 different types of TV shows with varying duration (from 15 minutes up to an hour). The dataset contains hundreds of unique faces and voices, and is heavily unbalanced (news anchors being shown frequently while some interviewees appearing only once for a few seconds).

2.9 Conclusion

In this chapter, an overview of the work related to the subject and contribution of this thesis was presented. The two main topics of this thesis are connected to deep learning and active learning. As such they will be addressed separately.

2.9.1 Deep learning

The chapter started with the introduction to the classical pipeline for image classification and retrieval, i.e. engineered features coupled with a trainable classifier. This was than compared with the more recent development of deep learning where the new pipeline learns both the features and the classifier at the same time. The contrast between these two approaches is the source of one of the contributions of this thesis, which is presented in Chapter 4. Even though the deep learning approach outperform the classical approach on most datasets, the fusion between the two was not investigated in detail in the literature.

After the introduction to the general neural network architecture, a more specific convolutional neural network structure was presented along with the overview of its key elements (convolution, activation function, etc.). For each element, both the most commonly used and the state-of-the-art approaches were presented. Afterwards, different CNN applications are showcased, including its use as a feature extractor and its use for object localization. Different state-of-the-art architectures for image classification are presented next. The use of those architectures in the context of audio processing, in particular speaker recognition, is the main subject of investigation in Chapter 5. The use of spectrograms with CNNs for speaker recognition was not extensively explored in the literature. Nor was the combination of the output of such a system with the outputs from other state-of-the-art speaker recognition algorithms.

2.9.2 Person identification

The second part of this chapter gives an overview of active learning. The most common and classical approaches are first introduced, which includes relevance sampling, query by committee and uncertainty sampling, as well as different scenarios were active learning could be applied. Afterwards, the focus is put on the use of active learning, which usually involves the use of a trainable model, with unsupervised approaches such as clustering. Later, some solution which are only based on the unsupervised approaches are introduced. Label propagation algorithms are presented next with emphasis on their application to person identification in videos. The connection between the two topics (active learning and label propagation) is the main subject and contribution in Chapter 3.

Finally, some practical challenges concerning the use of active learning are discussed. The particular issues that concern the research in this thesis are highlighted. Further consideration of possible issues and limitations of using active learning in this context are presented at the beginning of Chapter 3.

Chapter 3

Active learning for multimedia

3.1 Introduction

In this chapter an approach that aims at reducing human annotator involvement is proposed, namely annotation propagation for multimodal data. In particular, this method addresses the problem of how to effectively name numerous persons within a video with the lowest degree of human annotation. The technique presented could be considered as unsupervised active learning, as opposed to supervised active learning, which can make use of a classifier's output to determine samples for annotation [SQ12b, TK02].

Recent years have brought an outstanding and constantly growing amount of heterogeneous data. This immense quantity of videos, available thanks to the widespread availability of TV and the Internet, can be a source of very useful and important information. In order to handle such data and be able to utilize it correctly, its indexing and annotation is required. However, because of the complexity and multimodality of the data a human annotator is usually needed. On the other hand, it is not possible to annotate such a large quantity of videos due to the costs (taking into account both time and manpower) of manual intervention. To address this problem there are techniques being developed that can determine the most suitable instances of the dataset for annotation. Active learning is a group of such methods that try to determine the most informative and relevant samples for manual annotation [AQ07]. An overview and recent developments in active learning are provided in Chapter 2.

In summary, the main contribution presented in this chapter is an efficient strategy for cluster annotation when dealing with the task of multimodal person identification. For this to be possible an unsupervised system for label propagation was developed beforehand and it is shortly described here. However, the application of this system to a simulated manual annotation scenario can be considered as a new insight. The results of this study clearly show the advantages of the use of both the overlaid names (as the source of labels for the cold start) and propagation of annotation within clusters. Additionally, the cross-modal effects are visible when annotation addresses just a single modality. Additional experiments are performed to test this framework, which includes speaker annotation and speaker identification model training. The results for these experiments are

presented in subsequent sections.

The rest of the chapter is organized in the following way. Section 3.2 describes the problem in more detail. What follows in Section 3.3 is some practical considerations and limitations of this approach. Section 3.4 presents the data corpus in more detail. In Section 3.5 the sources of the features are introduced and described. Section 3.6 defines the main evaluation metric. This is followed by the presentation of the results of the first experiment on multimodal propagation in Section 3.7. Next, the speaker annotation experiments are shown and discussed in Section 3.8. The final simulated experiment involving model training as well as the description of a real-life dry run of the system are presented in Section 3.9 and Section 3.10, respectively. Finally, Section 3.11 gives the conclusions and future work.

3.2 Problem overview

Dealing with complex multimedia documents such as videos can be problematic, especially if no annotation is available. Contrary to, for example, image annotation, labeling videos create a set of new and unique challenges. First of all, the division of a video into an arbitrary group of segments, which are somehow meaningful (e.g. individual scenes, shots, locations, etc.), can be problematic. Second, the existence of multiple modalities makes the task dependent on the final use of the annotated data. If the goal is to create a dataset that can be used for training speaker models, the annotation process should concentrate on the speech segmentation, which may differ from the structure of the data for other tasks (e.g. face, object or scene annotation). The set of approaches proposed in this chapter tries to address these problems to some extent.

In a typical scenario involving human annotators, the task is usually binary, i.e. when given an image or a sound sample, one has to determine if a given concept (chair, car, mountain, etc.) is present or not. Such an approach has the advantage of being very efficient. On the other hand, when dealing with person identification the annotator has to provide a specific name if a given person was not seen before. This could be quite time consuming and prone to errors if a way of writing of a name is not standardized. A significant portion of such tasks can be reduced to name checking (or choosing the proper name from the list of candidates) if automatic initial labels and an annotation propagation procedure are used. In a real life scenario and when the proposed system is used, the human annotator would be presented with a single image or a single speech track that represents a corresponding cluster.

3.3 Practical considerations

Any potential solution to the problem presented in the previous section is limited by additional constraints, which have a source in the specific data type that needs to be annotated and in the assumptions of the project itself. The restrictions can be boiled down to the following main points:

1. *Real time response.* The main assumption is that the approach needs to work with the human in the loop. Therefore, the system needs to provide the human annotator with a continuous stream of instances for annotation. This situation creates a restriction on the computation time of the algorithm and may be a reason to exclude some of the established methods that are present in the literature. The approaches based on optimal experimental design (e.g. [ZCB+11] or [YBT06]) can be too complex in terms of required computational resources and are usually limited to selecting one instance per active learning cycle or have scaling issues. The use of a batch mode may be limited in some cases as well. As far as the evaluation of different active learning strategies goes, the computational time constraint is rarely considered in the literature. There is also a very limited number of papers that present the use of active learning in practice (see Chapter 2 for an overview), which incorporates the real human annotator.

2. *Type of concepts to be annotated.* Another issue that is related to the task of person identification is the potential number of concepts that can be present. Here, the concept is equivalent to a given person's identity. Depending on the set of videos, the number of people with a identifiable identity can range from just a few (in the case of a debate program with a presenter and some invited guests) to a few hundred (news shows, celebrity gossip, etc.). Additional challenge is that the number of people present in a given video is not known beforehand. Therefore, new identities may be discovered during the annotation process. This stands in contrast to an annotation scenario [VG14] where a set of concepts that needs to be labeled is established, constant and general (e.g. everyday objects or animals).

3. *Concept imbalance.* The imbalance between the number of instances available for concepts in the dataset can lead to a number of different challenges, especially in relation to model training. For instance, the REPERE corpus is based on real life TV broadcasts, and therefore the time a person is speaking or appears on screen can vary depending on his role and the program type. TV presenters (or anchors) and top level politicians (like the president or the prime minister) are frequently present, while some guests or eyewitnesses (in news programs) can appear only once for a very limited time.

4. *Model-based active learning.* Most approaches used with active learning (see Chapter 2) employ some kind of a learning model that tries to predict to which concept a given instance belongs. After obtaining the prediction scores (or probabilities) a selection method is applied to determine, which instance will be annotated be the human oracle. This can be done, for example, by selecting the most uncertain instance. Because of the reasons discussed in the two previous points, this scenario may be difficult to apply to the dataset used in this study. The potential of discovering new classes means that both the set of models (in this case biometric models, which try to identify a particular person) and the set of concepts should not be

fixed. Additionally, the imbalance between classes makes the quality of the potential models differ significantly. Also, in the case of some models (e.g. for speaker identification) a certain amount of data is required in order to obtain even the basic performance. This particular aspect will be explored later in this chapter.

5. *Automatic segmentation.* In order to efficiently label a video , it needs to be segmented into parts small enough that each represents (ideally) a single concept (in this case the same face or the same voice), but also big enough so that annotation is feasible. Annotating every single frame of a video separately and manually is not economically possible and, on the other hand, labeling only full shots or scenes can drastically reduce the future utility of the data. To that end, the instances are usually presented in the form of the face tracks (which contain the same face across subsequent frames) or speech segments (the same voice over a short period of time). Such segmentation is mostly done automatically, but can be refined with the help of the annotators. However, in essence these are unsupervised process and the resulting segmentation may contain some impurities (e.g. two voices, not the whole face). On the other hand, there may also be false positives, i.e. face tracks that do not contain a face and speech segments with just noise or music.

6. *Noise.* Another problem is the quality of the dataset itself. For example, in the REPERE corpus high level of noise is present for both speech and face. For speech, there are several potential sources of disturbances. Varying quality of the recorded voices can be problematic. This can range from debates in TV studio environments to conversations carried out over the telephone. Background noise and music can be present during speech. Also, during some lively debates, more than one person can speak at the same time. All of those aspects can have a significant influence on the quality of the models trained with such data. When dealing with visual data such as faces, the noise may results from, for example, obscured parts of the face, the angle under which a given face appears and the size of the face itself, which can make one face indistinguishable from the others. Some of those problems are addressed in the preprocessing stage.

Due to the reasons mentioned above, some classical approach to active learning may have problems when dealing with the data. This is also the reason why the model-based approaches were avoided in the preliminary experiments. It was, therefore, decided that unsupervised approaches could potentially be more suited for this task.

3.4 Dataset presentation

The REPERE challenge [GCM+12a] was designed to help evaluate person identification in videos[1]. When given a video of a TV show (either news broadcast or a talk-show) the aim is to identify who is speaking at a given moment and who can be seen at a particular time. This evaluation also provides the participants with the data, which consists of a series of shows from the French TV channels BFM TV and LCP. Not all frames of the video are annotated, but rather one every 10 seconds. There are 7 different types of shows. The series differ in length (from around 15 minutes to an hour) and therefore, also in number of annotation available for each (from around 20 to more than 100). There are also significant differences in terms of the number of participants (from a political debate with 3 persons, to dozens within a single video). Table 3.1 presents the names of the TV programs present in the corpus, the name of the French TV channel on which they appear and the approximate duration of each show.

Show	Channel	Duration (min)
BFM Story	BFM	60
Planete Showbiz	BFM	15
Ca vous regarde	LCP	15
Entre les lignes	LCP	15
Pile et Face	LCP	15
LCP Info	LCP	30
Top Questions	LCP	30

Table 3.1: TV shows present in the dataset.

In Figure 3.1 a set of screenshots from different videos are presented. The videos in the dataset were selected to represent a varied set of TV broadcasts. There are significant differences when it comes to the camera movement and angle, light conditions, head sizes and their orientations, prepared and spontaneous speech, number of people visible in each frame and so on.

3.5 Feature sources and components

In order to propagate labels throughout the dataset, the method tries to find the most promising cluster to annotate, rather than single tracks. As a source of initial labels the optical character recognition (OCR) approach is used, which utilizes the overlaid text visible in TV broadcasts (e.g. when a person is presented for the first time his or her name is shown at the bottom of the screen).

In this section different elements of the proposed system are described in detail. The system in question is composed of the following three modules:

- Text detection and recognition algorithms, which provide the initial labeling in an unsupervised way.

[1] http://www.defi-repere.fr

(a) BFMStory (b) Planète Showbiz

(c) Top Questions (d) Entre Les Lignes

Figure 3.1: Frames taken from different videos to illustrate the diversity of the dataset: (a) news shows with invited guests, (b) interviews with celebrities, (c) broadcasts from the national assembly and (d) TV debates.

- An agglomarative clustering algorithm, based on the multimodal distance matrix between tracks (face vs face, speaker vs speaker and face vs speaker) to form multimodal clusters. This step is described in detail in [Poi13].

- Selection strategy that chooses tracks for annotation in order to reduce human annotator involvement.

Figure 3.2 gives a overview of the system used in this study. First, both speaker and face tracks are extracted from the videos. In order to create multimodal cluster, the distance between tracks of different modalities are normalized, so that they can be comparable. The output of a multilayer perceptron classifier, based on lip activity and other temporal characteristics, is used to establish the association between face and speaker tracks. Using the names obtained by the OCR, the multimodal clusters are initially labeled. Next, the active learning cycle is introduced. Here, based on the multimodal cluster structure and already available annotation, a given selection strategy chooses a set of unlabeled samples for human annotation. Once the new labels are obtained, cluster recalculation and annotation propagation takes place. During this procedure some clusters may be combined. This gives rise to a slightly modified cluster structure, which is used for the next iteration of the active learning cycle.

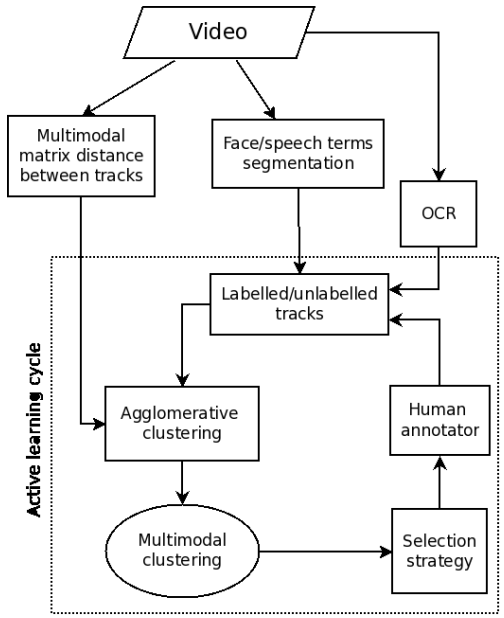


Figure 3.2: System overview.

3.5.1 OCR system

Optical character recognition (OCR) system is following the design proposed in [PBQT12]. Its main purpose is to introduce alternative sources of information for person identification apart from voice and appearance. Often, guests and speakers, when introduced on a given television show to the viewer, are presented alongside overlaid text containing their name. In the context of this work, the OCR system is used to generate automatic initial annotation, which would be later improved and expanded by human annotators.

This module is composed of two parts: text detection and text recognition. For text detection a two step approach following [AGP10] is adopted. The coarse detection is obtained through a Sobel filter and dilatation/erosion as in [WJC02]. Additionally, to overcome the shortcomings of binarization, several binarized images are extracted of the same text, but temporally shifted. This is done to filter out false positive text boxes. For the text recognition part a publicly available OCR system from Google called `Tesseract`[2] was used.

3.5.2 Speech clustering

Speaker diarization is done in the following way. After splitting the signal into acoustically homogeneous segments, the calculation of the similarity score matrix between each pair of speech tracks is done with the use of the BIC criterion [CG98] with single full-covariance Gaussians. Next, the distances are normalized to have values between 0 and 1.

[2]http://code.google.com/p/tesseract-ocr/

3.5.3 Face tracking

Face detection and tracking follows the particle-filter framework using detector-based face tracker introduced in [BBF+10]. The initialization of the face tracks is done by scanning the first and the fifth frame of every shot. Three detectors are included: frontal, half-profile and profile, the goal of which is to prevent the face detector to be dependent on the initial pose. Tracking is done on-line, i.e. with the use of the information from the previous frame, the location and head pose of the current frame are established.

Afterwards, a 9-point mesh is imposed on the image of the face (2 point per eye, 3 for the nose and 2 for the lips). A confidence score helps to determine if a given face can be successfully used [ESZ06]. If so, a HOG descriptor with 490 dimensions is calculated on that face image [DT05]. After such a descriptor is made for every suitable image in the sequence, an average descriptor is then established for the whole sequence. This is then projected to a reduced space of 200 dimensions thanks to the LDML approach [GMVS12]. Next, the Euclidean distance is computed between each track. Finally, the output is normalized (values between 0 and 1).

3.5.4 Multimodal clusters

In order to make use of both modalities (face and speech) at the same time and to connect the face tracks that co-occur with speech segments, additional features are used, such as lip activity, head size, etc. A multilayer perceptron is then trained on those features. The output of the model (with values in the range of 0 and 1) is then treated as the distance between speech and face tracks. This is used to produce initial multimodal clusters, i.e. clusters constructed from both the speech and face tracks.

When some labels are available (for example extracted from the overlaid names seen on the screen), some constraints are set to forbid merging the clusters (denoted as c) with different names (i.e. n) associated to them (i.e. $c(n)$). Note that clusters can contain more than one person name at this step. The agglomerative clustering algorithm is used for this purpose. The full list of constraints is as follows (based on [PBB+13]). The cases that allow two clusters c_1 and c_2 to be merged are:

- $c_1(\emptyset) \cup c_2(\emptyset) \Rightarrow c_3(\emptyset)$
- $c_1(n_1) \cup c_2(\emptyset) \Rightarrow c_{new}(n_1)$
- $c_1(n_1, n_2) \cup c_2(\emptyset) \Rightarrow c_{new}(n_1, n_2)$
- $c_1(n_1, n_2) \cup c_2(n_1) \Rightarrow c_{new}(n_1)$

An example of a case where two clusters can not be merged is when they do not share a common label, but do have at least one label assigned. This can be presented as:

- $c_1(n_1, n_3) \cup c_2(n_2) \Rightarrow \emptyset$

Apart from that, there are also temporal constraints. For example, two different clusters with the same name assigned to them cannot have a co-occurring face track, i.e. faces that appear at the same time in a video can not belong to the same cluster.

3.6 Evaluation metric

In order to be able to compare the effectiveness of different approaches the F-measure was adopted as evaluation criteria. It is calculated in the following way:

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.1}$$

When calculating the metric, faces with less than 2000 pixels are not considered. Also for the evaluation purpose, persons, which were not identified in the corpus, are not included in the score. In other words, a given face may be annotated correctly by the algorithm, but will not be counted as such due to the lack of full reference in the corpus.

3.7 Multimodal propagation

In this problem the propagation approach is tested on the video set using different modalities. The interaction between these different types of data are also explored.

3.7.1 Data corpus

The experiments are performed on the REPERE corpus. For this study, the test set, with the running time of around 3 hours, is used for evaluation. For the test set there are 1229 annotated frames in total.

3.7.2 Proposed solutions

In this section the proposed approaches are introduced.

Four different annotation selection strategies are explored and evaluated:

- Random – the basic baseline, which chooses random annotation for every show. It does, however, sometimes yield decent performance, as in [HJZ10].

- Chronological – chooses the annotation according to its time of appearance in a given show starting from the beginning. Due to the nature of some of the shows (e.g. political debates with a limited number of people, which are introduced by the presenter at the very beginning) this approach can be quite effective.

- Biggest cluster first – this strategy makes use of the multimodal cluster structure of the tracks (both the face tracks and speaker tracks). Let n_t be the number of tracks within a given cluster and a_t be the number of annotations already assigned to that cluster. The score S_c is calculated as:

$$S_c = \frac{a_t}{n_t} \tag{3.2}$$

  and the cluster with the minimum score is selected. Afterwards, a track for manual annotation from the cluster is chosen in a chronological manner.

- Biggest cluster probability – a modification of the previous algorithm, which rather than selecting the cluster with the lowest S_c score, assigns a probability to be chosen for annotation, which is proportional to the score at a given step.

3.7.3 Simulated run results and discussion

Figure 3.3 presents the results of the active learning simulation. Four different scenarios were tested. The main observations about the results are listed below:

- As an additional experiment the random selection strategy was launched without the use of the annotation propagation after every step (called 'No prop rand' in the figures). This was done mainly to show the effectiveness of the propagation mechanism, which use seems to be very beneficial to the overall performance of the system. Also, when applying annotation from a different modality the score of 'No prop rand' does not change, due to the lack of cross–modal effect.

- The chronological strategy seems to give the worst score, which is not surprising given its simplicity. However, choosing samples at random is also not much better.

- A more promising strategy is to pick the biggest clusters for annotation first. It is very effective at the beginning, but the increase in performance at the later steps is not quite satisfactory. When using head annotation for the face error it tends to be at times significantly worst than random.

- Amongst the four tested strategies the BigProb tends to display the best performance overall. It is also the most consistent. It is significantly better than Random and Chronological strategies at the beginning of the simulation, but also manages to keep this advantage in the following steps.

- One of the most interesting findings in this work is the increase of performance observed on one modality when using the annotation from the other. This is due to the use of multimodal clusters and the annotation propagation within them. In other words, while annotating speakers one can also significantly increase the performance of face annotation. This could be used in a practical active learning scenario, where annotation of different

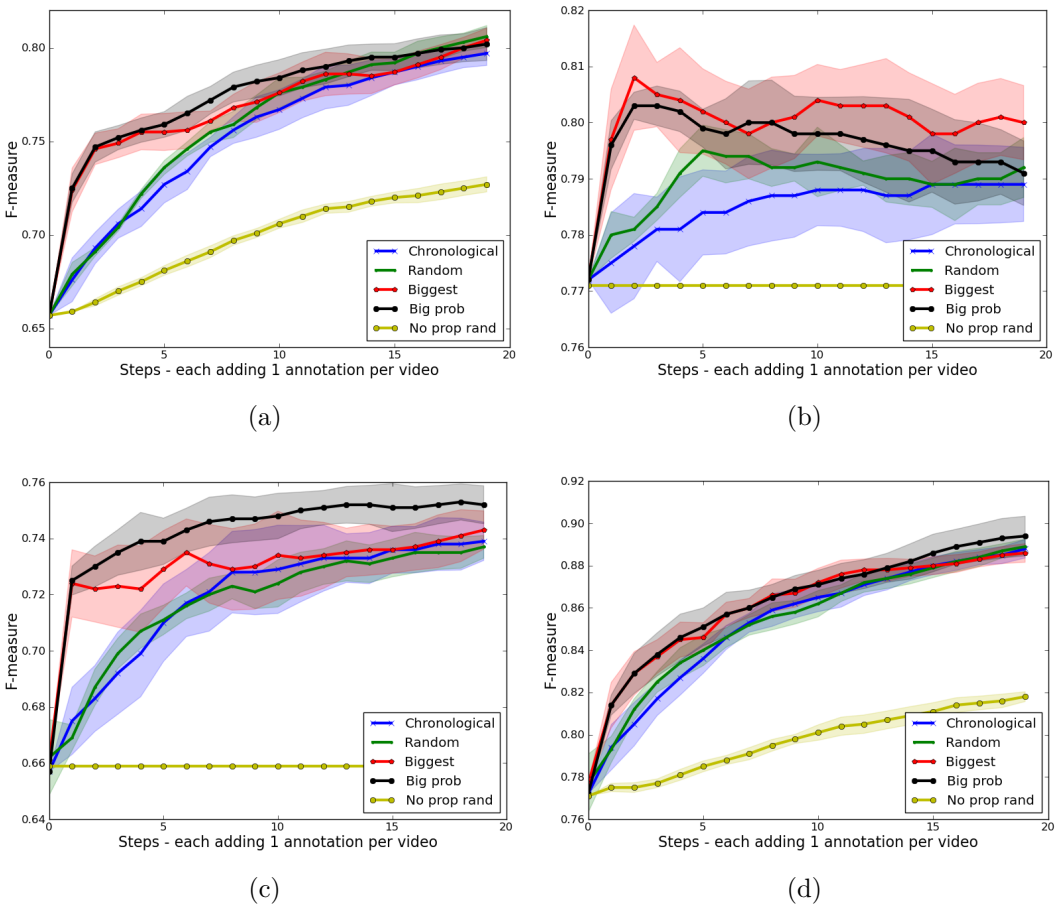


Figure 3.3: The results of runs using different modalities for annotations and evaluation. (a) Face tracks used for both annotation and evaluation. (b) Face tracks used for annotation and speech segments for evaluation. (c) Speech segments used for annotation and face tracks for evaluation. (d) Speech tracks used for both annotation and evaluation.

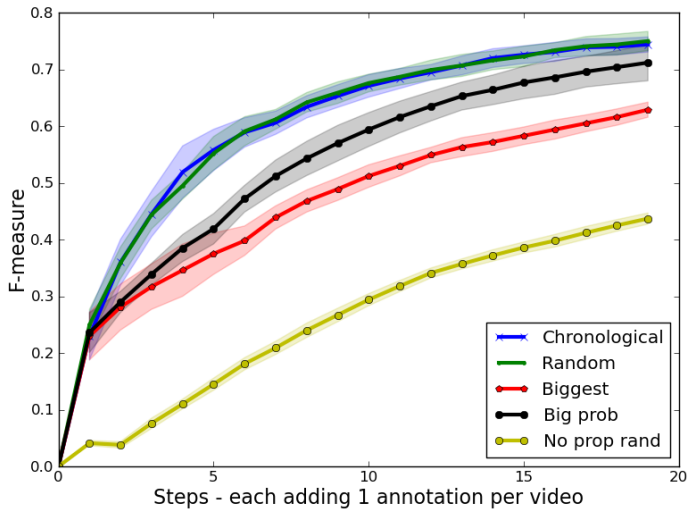


Figure 3.4: The F-measure score without the initialization done by the OCR labels. Face score using the head annotation.

modalities has a different cost (time, difficulty for the human annotator) associated with them. In this case some of the modalities may be preferable for annotation, while the labeling of others can be reduced without a major drop in performance.

- The selection strategies display similar relative performance independently from the used annotation. There is, however, a difference in overall performance achievable by each modality, which is what is to be expected. One of the key differences between the speaker and the head modality is that, usually, in the case of the former just one voice is heard at any given moment, i.e. two or more voices rarely occur simultaneously and for a significant period of time. While, in the case of the later, several faces visible in a single shot is often the case. This could help to explain the slightly erratic behaviour observable in Fig. (b), which shows the speaker score using head annotation.

- In a case where there is no prior labels available (from the OCR system or otherwise) the behaviour of the strategies changes significantly. Figure 3.4 shows the result of the simulation using the head annotation without the OCR initialization, the annotation is propagated with each step and the score is calculated for the face. Without any additional prior knowledge the multimodal clusters are constructed based on the distances alone. This means that there is a higher probability that they will have lower purity, i.e. containing tracks from different people, compared to their counterparts with the OCR labels. When a given track is annotated and the label spreads to the other tracks in the cluster, the tracks describing other people are also annotated. Therefore, strategies that make use of the cluster structure of the data and emphasize larger clusters (which probably have lower purity) perform worse than those that ignore this information.

3.8 Speaker annotation

Another set of experiments were done this time focusing on speaker annotation. The goal was to generate enough labeled data to train individual speaker models. This would have to be done with the least amount of human involvement possible. Creating such a dataset can be challenging, given that a decent amount of speech time is needed as well as the speech itself needs to be of a decent purity.

As was mentioned before, a typical active learning approach usually involves a trained model to generate relevance or uncertainty scores. The drawback of this approach is that when dealing with speaker identification, the list of classes (i.e. individual speakers) is not known beforehand and new classes may appear during the annotation process (see Section 3.3). This can be seen in the case of video annotation where propagating the available labels can be as efficient as training a model [PMT10].

3.8.1 Proposed method

In this section the proposed method to address the problem is presented in detail. The assumption is to use an unsupervised approach to active learning, which is based mostly on the data structure, i.e. the (monomodal or multimodal) clusters, and the length of the speech tracks. Several strategies were tested. Including benchmark approaches, which consist of random selection of speech tracks for labeling from the still unlabeled pool of tracks. Also, a chronological selection of tracks, i.e. according to their order of appearance in the video, was tested. The best performing approach is presented in detail in Algorithm 1.

Data: A set of speech tracks $S = \{s_1, \ldots, s_N\}$.
Result: A set of annotated speech tracks $Ann = \{a_1, \ldots, a_M\} \subseteq S$, $M \leq N$.
$Ann \leftarrow \emptyset$;
Initialise a set for propagated annotation: $Ann_{prop} \leftarrow \emptyset$;
while $|S| \neq |Ann| + |Ann_{prop}|$ **do**
  $s_{temp} = \max S \setminus (Ann \cup Ann_{prop})$;
  $Ann \leftarrow Ann \cup \{s_{temp}\}$;
  $Ann_{prop} \leftarrow propagate(Ann)$;
end
return Ann;

Algorithm 1: Active learning cycle with longest track selection

3.8.2 Data corpus

As in the previous experiment, the REPERE corpus was used. However, the number of videos were extended. For this study it consists of 205 videos, which sums up to the total length of around 40 hours. As before, it contains recordings of 7 different TV shows from the French TV channels BFM TV and LCP. In this

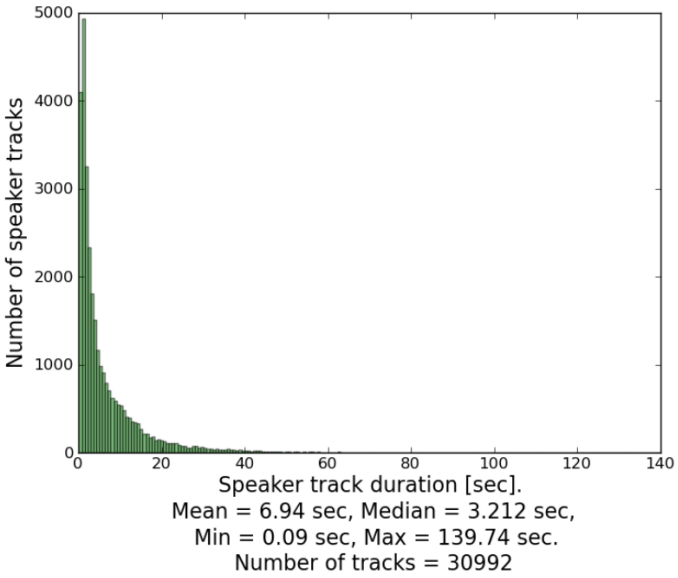


Figure 3.5: Speaker track distribution based on duration and key statistical values of the speech track corpus.

study the focus is on the speech modality, while the face tracks are not used. Figure 3.5 shows some basic statistics about the speech segments as well as the duration distribution.

3.8.3 Evaluation metrics and experimental settings

Contrary to other studies in this chapter where the F measure is used, here two different metrics are used. First, the identification error rate (IER) evaluates the overall performance increase at every step of the active learning simulation. Additionally, for assessing the quality of the annotation for every speaker (in order to train speaker biometric models as presented in Section 3.9), the purity is calculated. Then, every set of speaker track with purity score above 90% and above a given duration threshold is counted at each step of the experiment.

The experimental settings follow the ones introduced in Section 3.7. As before the evaluation is done using a simulated active learning where all the labels provided by human annotators are initially unknown and are revealed for a given speech track when the selection method chooses it. At each step of the simulation (consisting of 20 steps in total) a single track is selected for labeling for every show as long as the new annotation is available. The whole experiment is repeated 10 times, at each time 80 % of the annotation per show is randomly selected, while the rest is not used in any way. In addition for all subsequent plots in this section, the shaded area around the curves (with a corresponding color) is the standard deviation at each point.

3.8.4 Results and discussion

In this work two tasks were taken into account. On the one hand the efficiency of the annotation process is considered, in which the error reduction at each step is measured. Additionally, the ability to produce speaker corpora, which can later be used to train biometric models is also investigated. In the case of the monomodal experiments, only the speech segments extracted from the videos are used and, at the beginning of the simulation, no annotation is available.

Figure 3.6(c) presents the IER results for the monomodal speaker annotation experiment. In addition to the strategies already mentioned, a strategy not making use of the label propagation is presented for reference. In this case, the selection of the tracks for annotation is done randomly.

Figures 3.6(a) and 3.6(b) show the number of obtained speaker corpora with purity score above 90% and with total speech duration (the sum of all annotated tracks for a given speaker) above 20 and 60 seconds, respectively. For the 20 second condition, the proposed strategy works better than random at every step. Moreover, both approaches that make use of the annotation propagation are far better than the standard, no propagation method. The gap is even bigger when the 60 second condition is considered. Here the standard approach requires more than 9 steps (9 annotations per video) to produce any annotated speaker data meeting the criteria; and after 20 steps, it is still lower than when compared to the best strategy after a single step.

Figure 3.6(d) presents the results of an experiment where the co-occurring overlaid names were extracted from the video and used as an initial annotation for speakers. Afterward the annotation was further refined with the use of active learning. When compared to the monomodal scenario this approach seems to be beneficial, also for the number of generated speaker corpora with the duration of 60 seconds or longer, which is equal to 315 after 10 steps for the longest track strategy against 190 for the corresponding approach without overlaid names.

Finally, an additional experiment was done with the use of the head annotation only. In this scenario the human annotator would be asked to label faces rather then speech tracks. In this case, the speakers are annotated indirectly, through the use of multimodal clusters, which contain both the speech and face tracks. By labeling a face track, all the speech tracks in the cluster are also annotated. Figures 3.6(e) and 3.6(f) show the identification error rate measured on the speaker annotation exclusively with and without the overlaid names and with the random selection strategy. The results of annotation with the use of speaker tracks are provided in the corresponding plots for reference. The advantage of such an approach is that usually the process of face annotation is faster than speaker labeling. It is possible to present to the annotator a set of faces at the same time, while speech terms need to be heard one by one. The proposed approach makes it possible to produce annotations for two different modalities by presenting to the annotator just a single annotation task.

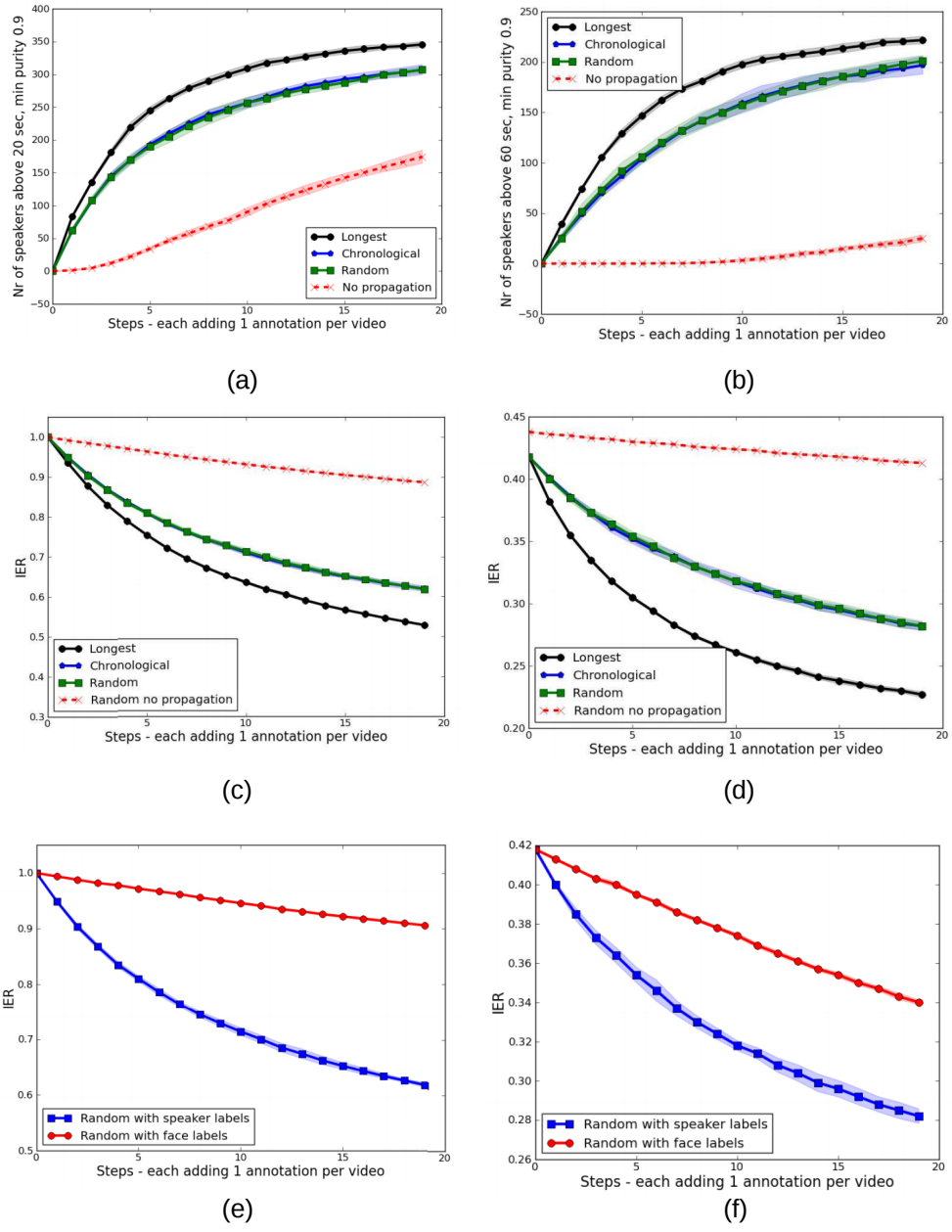


Figure 3.6: Different simulation results. **(a)** Monomodal experiment showing the number of speakers with annotated tracks longer than 20 seconds. **(b)** Similar, but with the total duration of annotated track longer than 60 seconds. **(c)** Monomodal experiment showing the IER scores. **(d)** As in (c), but with the overlaid names as cold start. **(e)** Speaker annotation using face labels vs speaker labels. **(f)** As in (e), but with overlaid text as an additional modality.

3.9 Model training with propagation

The goal here is to investigate how much annotation is needed to produce an accurate speaker models. This serves as an extension and verification of the previous experiment presented in Section 3.8. Here, the effectiveness of the active learning and annotation propagation methods is evaluated for a true speaker identification task.

The comparison is made between models that were trained in an unsupervised manner (with the only source of labels being the overlaid names present in the video) and those that gradually use more human annotation, i.e. they become more and more supervised. The quality of the generated speaker models is evaluated through speaker identification tests on a separate set of unseen data.

3.9.1 Speaker identification system

In order to evaluate the active learning runs, a state-of-the-art speaker identification system was chosen. To that end an automatic speaker identification system based on front-end factor analysis [DKD+11], also known as Total Variability Space (TVS), is chosen for this study. In this section a short overview of such a speaker modeling system is presented.

Gaussian Mixture Models (GMM) are usually used as generative models for the representation of the acoustic feature space in speaker recognition systems. A general model based on the GMM is called Universal Background Model (UBM) and is first trained on speech data from multiple speakers (to better represent general, person independent feature characteristics present in the corpus) and is speaker-independent. Afterwards, speaker-specific models are obtained using Maximum a Posteriori (MAP) adaptation, which are also GMMs.

After the MAP adaptation, specific speaker models can be represented as high-dimensional supervectors of means of distributions. Using factor analysis on these supervectors, speaker models can be represented as a low-dimensional identity vector (i-vector).

In this approach, an i-vector w_s can be calculated by the equation $M_s = M_0 + Tw_s$, where M_s is the mean supervector of speaker model, M_0 is the mean supervector of the UBM, and T is the low-rank rectangular matrix representing the variability space of the i-vectors learned in an unsupervised manner. In the case of having multiple tracks for speaker modeling, i-vectors extracted from each speech segment are averaged and the average i-vector is used as the new speaker model. The averaging can be done in a weighted manner according to the logarithm of the duration of the tracks. Length normalization is done prior to generation of speaker models [GREW11].

Identification is done by extraction of an i-vector of a target track and calculation of the cosine similarity score of the extracted i-vector with all the speaker models (each represented with its own i-vector). The speaker corresponding to the model with the highest score is finally chosen.

The UBM consists of 1024 gaussians and is learned on the training data for the GMM-UBM system. A UBM of the same size is used for training the T

matrix. TVS is then learned using Expectation-Maximization (EM) algorithm on the segmented training data. The dimension of the output i-vectors is set to 400 and weighted averaging is done for generation of speaker model i-vectors. A modified version of MSR Identity Toolbox [SSH13] is used for experiments.

MFCC features

For feature extraction, Energy and Mel-Frequency Cepstral Coefficients (MFCCs) of 13 dimensions are extracted every 10 ms with a window length of 20 ms. These features along with their delta and delta-delta coefficients are concatenated. Feature warping [PS01] is done on these features and static energy feature is discarded resulting a 41 dimensional feature vector per frame. Sound activity detection is done using bi-gaussian distribution on frame log-energies. The segmentation is done in a similar manner as described for speaker diarization. Segments with less than 150 ms of voice activity are not used.

3.9.2 Data corpus and experimental protocol

As before, the REPERE corpus [GCM+12a] was used for evaluation of the annotation propagation methods. And just like in the previous experiments, this corpus consists of seven types of shows from the French TV channels BFM-TV and LCP, and is aimed for development of person identification methods on broadcast data. The corpus division into train, validation and test sets follows the official REPERE challenge guidelines. The train set consists of 58 videos and has a total duration of approximately 28 hours. The development set and test set, with a total annotated duration of ~8 hours and ~13 hours, containing 57 and 90 videos respectively, are both used for evaluation. Only parts of the videos corresponding to specific shows are annotated. Length of these series differ, ranging from 3 minutes to half an hour, and therefore, the number of annotations per video vary widely from twenty to several hundreds.

In this study, simulated active learning is done on the training set, where all the manual annotations are available initially, but considered initially as unknown to the system. The labels are revealed to the system as the selection algorithm selects them for annotation. Simulation is done in 20 cycles and on each cycle only one annotation per show is selected. For each selection strategy, there are 10 replicas and randomly 20% of the data is completely left-out for each replica.

For every replica of each cycle and selection strategy, speaker models are created based on the output annotations of the active learning system (only on the training data). The speaker identification performance is evaluated on the development and test sets separately using standard F-measure. However, due to the fact that all the data is not annotated, the performance is only evaluated on the annotated tracks. The numbers of annotated tracks used in this study are 3490 and 4779 for development and test, respectively.

Tests are done in an open-set manner, which means that there may be speakers in the test set that do not appear in the train set, making them impossible to predict correctly. Results with correct labels corresponding to the maximum

possible F-measure that can be achieved as well as results with a fully supervised system are reported for comparison. Tests with and without OCR cold start are also reported.

3.9.3 Results and discussion

Figure 3.7(a) and 3.7(c) give the F-measure results with the standard deviation starting with no available annotation. With a good enough selection strategy one can get close to the fully supervised performance in just a few steps. Overall, the best approach is to select the longest track first, which is not surprising given that a long uninterrupted speech segment from a single speaker would lead, in most cases, to a good speaker model.

The results with the overlaid names as a cold start can be seen in Figure 3.7(b) and 3.7(d) for the development and test set, respectively. This approach gives an initial boost in performance, which can be further increased with just a few additional annotations. Regardless of the use of the OCR, both approaches are able to arrive at the same level of performance (especially in the case of the longest strategy), even though the use of OCR makes it faster for every selection approach. This would indicate that the approach presented here could be successfully employed on different datasets where the OCR may not always be available.

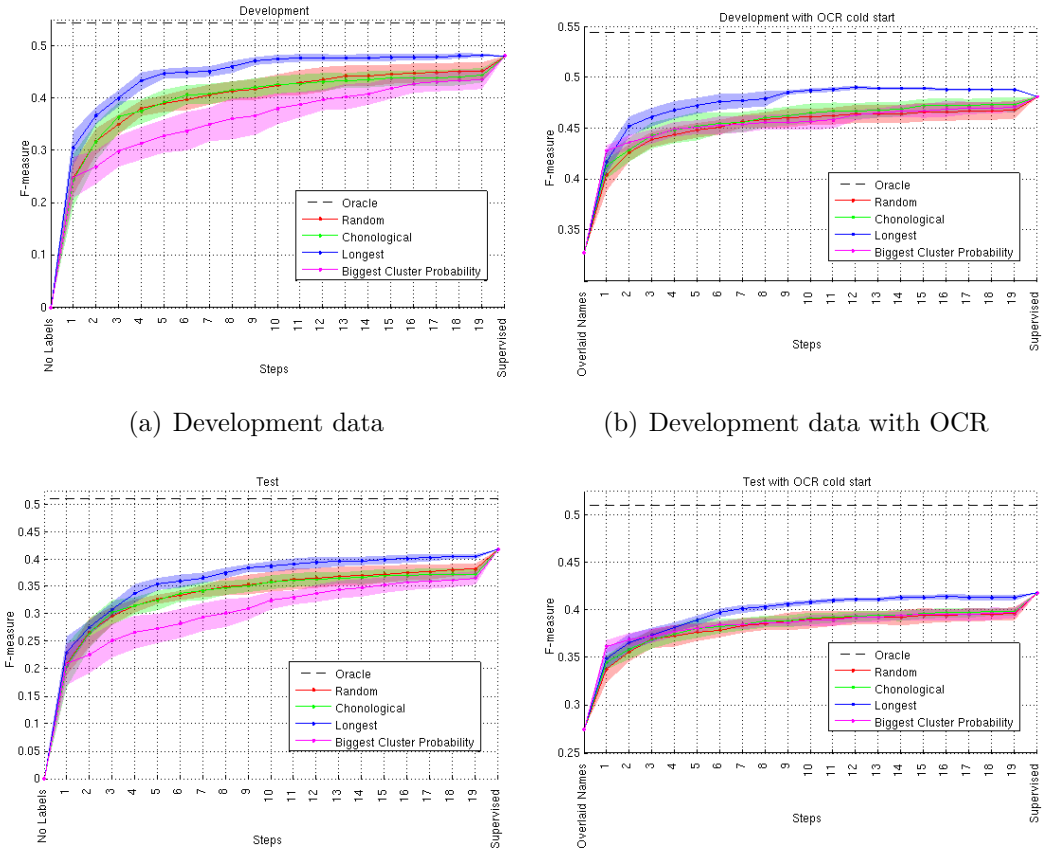
After around 6 steps (when using OCR) the performance increases only slightly. Without OCR it takes around 9 steps for the best strategy to get to a comparable level. It seems, therefore, that the use of such an active learning system can greatly reduce the number of annotation needed to produce competitive speaker models. Manual annotation is often expensive and time consuming. This approach can help to reduce this burden, especially when the final goal is to have reliable speaker models.

3.10 Dry run of a real-life active learning task

To verify the usefulness of the proposed methods, a limited real life run with human annotators participation was organized. The task consisted of annotating speech tracks extracted automatically following the approach presented in [BZMG06]. Each participant was given a video fragment corresponding to the time frame of a speech track and was asked to name the person speaking at the moment. Due to the nature of the videos (TV news broadcasts), most people were presented either by an overlaid text or by a spoken name.

3.10.1 Overview

9 users were involved in the dry run. The annotations were done simultaneously and lasted for around 1.5 hours per user. In this run only the speaker annotation scenario was tested (the faces of people present in the video were not annotated). The corpus consisted of 62 videos from the REPERE dataset [GCM+12b], which



(a) Development data

(b) Development data with OCR

(c) Test data

(d) Test data with OCR

Figure 3.7: Performance of the speaker identification system on the respective sets of data with and without the inclusion of the OCR system in terms of F-measure. The results with supervised speaker modeling as well as maximum possible F-measure in the open-set setup are also reported for comparison.

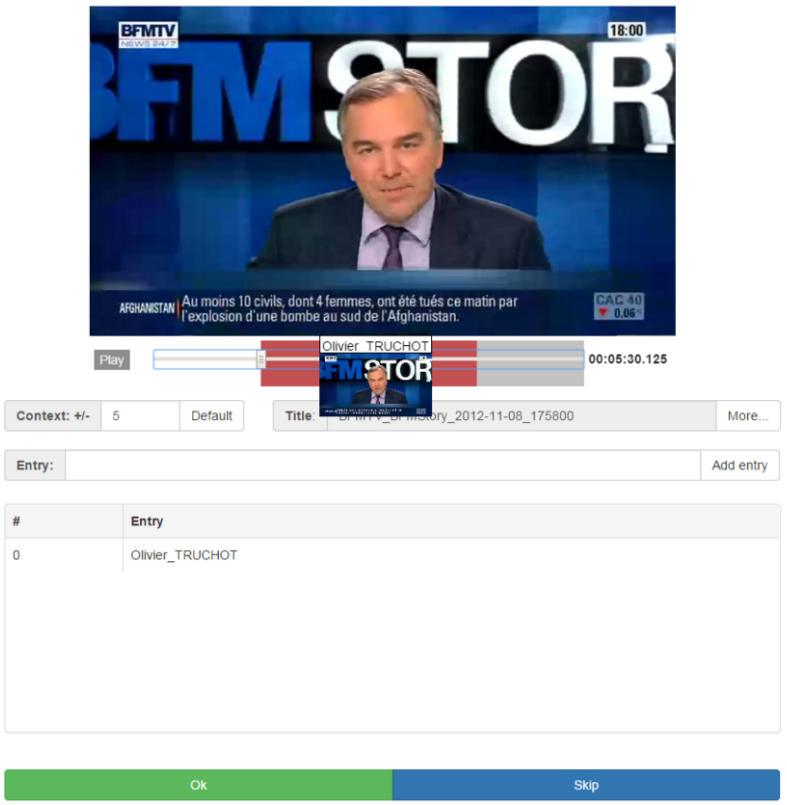


Figure 3.8: The graphical user interface used for the dry run.

included TV debates, news programs and parliamentary broadcasts among others. The annotations were done using the Camomile project framework[3] and using the graphical user interface (GUI) developed within the same project. An example of the interface used by the participants of the dry run can be seen in Figure 3.8.

During the run, a total of 716 speech tracks (with the total duration of 81 minutes) were annotated. Additionally, 654 tracks (total of 68 minutes) were marked as skipped (tracks which do not contain speech, but music, external noises, silence, etc.). The median annotation time is equal to 10.8 seconds. Additionally, because of the clustering present in the system, the annotations were propagated to the corresponding clusters. This produced a total number of 3504 labeled tracks (including the 716 annotated manually) with the total time equal to 7.81 hours. Additionally, the use of the multimodal clusters during the dry run enabled to get face annotation (1973 head annotations, for a total duration of 5.47 hours).

Due to its limited scope, as well as duration, it is hard to draw any substantial conclusions based on this dry run. Some attempts have been made however. The number of discovered speakers was measured depending on the strategy and can be seen in Figure 3.9. However, this result is limited, due to the fact that during the actual annotation process there was no time to test additional selection strategies and, therefore, only the best one was chosen (following the

[3]https://github.com/camomile-project/

results obtained in Section 3.8). So in order to have a baseline, the order of the tracks was randomized several times and the resulting new curve is used for comparison. It is true that the baseline strategy is not truly random as it is applied on a preselected subset of tracks.

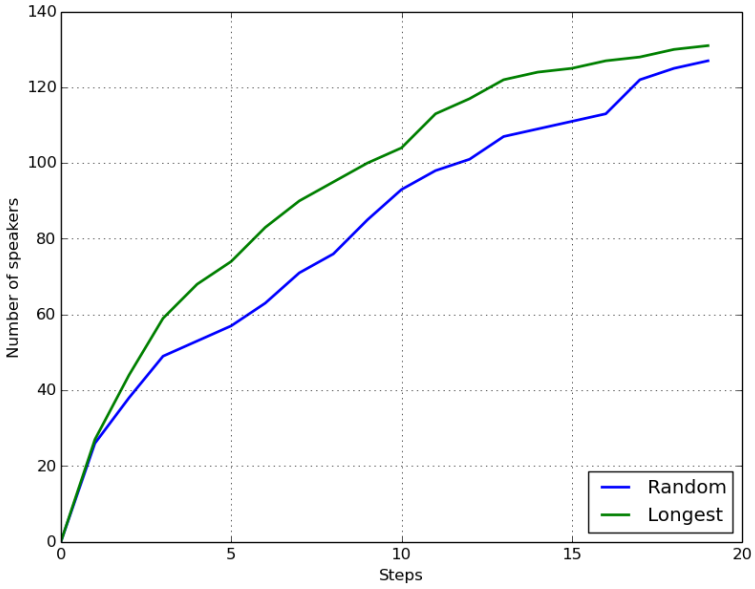


Figure 3.9: Speaker discovery during the dry run.

Overall, the real-life application of this framework showed its potential usability and benefits. This is especially true when one considers the amount of annotation obtained through automatic means of label propagation. It remains clear, however, that a longer and more thorough evaluation under real-life conditions would be required to fully evaluate the usefulness of such an approach. This would include the use of different selection strategies and more controlled human annotator participation.

3.11 Conclusion

In this chapter, a novel approach to the annotation of multimedia documents has been proposed. An active learning framework was designed and developed in a way that it can reduce the amount of work done by human annotators. To that end, the application of clustering was used, which combined different modalities present in a video document, such as faces and speech. This was connected with a number of different selection strategies.

A series of experiments were made in order to evaluate the proposed framework. These included studies involving both mono- and mulitmodal scenarios and different objectives. The latter included efficiently preparing the data for

training models for speaker identification. Among other things, the crossover effect of the use of different modalites was observed and investigated. Finally, the whole framework was evaluated in a real-life scenario, which was a limited dry run.

Chapter 4

Deep learning for multimedia

4.1 Introduction

Deep Convolutional Neural Networks (DCNN) have recently made a significant breakthrough in image classification [KSH12]. This has been made possible by a conjunction of factors (some of which are discussed in Chapter 2 in detail) including: findings about how to have deep networks effectively and efficiently converge [OM98], the use of convolutional layers [Fuk80] [LBBH98], the availability of very powerful parallel architectures (GPUs), findings about how exactly a network should be organized for the task [KSH12], and the availability of huge quantity of cleanly annotated data [DDS+09].

Not to minimize the importance of the hardware progress and of algorithmic breakthroughs, the availability of a large number of image examples for a very large number of concepts was crucial as DCNNs really need such amount of training data for actually being efficient. Data augmentation (e.g. multiple crops of training samples) can further help but only when a huge amount of data is already available. Such amount of training data is currently available only with ImageNet, which corresponds to a single type of application and only for still images. In the case of video documents for instance, several annotated collections exist but with much smaller number of concepts and/or much smaller number of examples. Trying to train DCNNs on such data generally leads to results that are less good than those obtained using "classical" engineered features (or descriptors) combined with more traditional and well established machine learning methods (typically SVMs), these being more suitable when small to moderate amounts of training data are available.

Two strategies have been considered for making other domains benefit from the success of the DCNN/ImageNet combination. The first one consists in pre-training a DCNN using ImageNet data and annotations as a source collection and then partly retrain or fine-tune it on a different destination collection [CSVZ14] [YCBL14]. Generally, only the last layers are retrained, the exact number of which, as well as the learning parameters, being experimentally determined by cross-validation. Though this strategy can produce much better results than by training the DCNN only on the destination data, it does not necessarily compete with classical approaches and it could lead to gains that are much less important

than in the ImageNet case.

The second strategy consists of employing a DCNN pre-trained on ImageNet as a source collection, applying it to a different destination collection and then using the final ImageNet concept detection scores or the output of the hidden layers as features for training classifiers and making prediction on the destination collection. In [RASC14a] a successful application of this strategy is presented and applied to a number of test collections for both image indexing and image retrieval.

In this chapter, these strategies are explored and their performance is evaluated in the context of video indexing. Also, there is an additional investigation on how they can be combined with classical methods based on engineered features and how they can be combined with other video-specific improvement methods like temporal re-scoring [SQ11]. Experiments have been carried out in the context of the semantic indexing task at TRECVid [SOK06] [OAM+15]. Additionally, an evaluation of the DCNN features / SVM classifiers combination is made on the object classification task of VOC 2012 [EEVG+15b]. In this chapter, the following contributions and observations are presented:

1. The results obtained for still images in the case of video shot indexing are confirmed: features learned from other training data generally outperform engineered features for concept recognition.

2. Directly training SVM classifiers using these features does better than partially retraining the DCNN for adapting it to the new data.

3. Even though learned features outperform the engineered ones, fusing them performs even better, indicating that engineered features are still useful, at least in this case.

4. Temporal and conceptual re-scoring methods, as well as the use or multiple key frames within video shots, also improve classification results obtained with DCNN features.

5. The DCNN features / SVM classifiers combination is very efficient for still images too and it was evaluated on the VOC 2012 object classification task.

The chapter is organized as follows: Section 4.2 describes related work; Section 4.3 describes the features and methods used for conducting the experiments; Section 4.4 presents comparative results on the TRECVid semantic indexing task; Section 4.5 presents results obtained on the VOC 2012 object classification task; and Section 4.6 concludes the chapter.

4.2 Related work

Semantic features are not restricted to DCNN and had already been used for multimedia classification and retrieval. The work presented in [SNN03] introduced

them as "model vectors". These provide a semantic signature for multimedia documents by capturing the detection of concepts across a lexicon using a set of independent binary classifiers. Ayache et al. [AQG07] proposed to use local detection scores of visual categories on regular grids or to use topic detection on ASR transcriptions for video shot classification. Su et al. [SJ12] also proposed to use semantic attributes obtained with supervised learning either as local or global features for image classification.

In all of these works and many other similar ones, semantic features are learned on source collections different from the destination one and generally for source concepts or categories different from those searched for on the destination collection. Hamadi et al. [HMQ15] used the approach using the same collection and the same concepts both for the semantic feature training and for their use in a further classification step. In this variant, called "conceptual feedback", a given target concept is learned both directly from the "low-level" features and from the detection scores of the other target concepts also learned from the same low-level features (the training of the semantic features has to be done by cross-validation within the training set so that it can be used for the second training step both on the training and test sets).

Concerning the first DCNN transfer strategy (DCNN re-training), Yosinski et al. [YCBL14] et al showed that the features corresponding to the output of the hidden layers are well transferable from one collection to another and that re-training only the last layers is very efficient both for comparable or for dissimilar concept types. Their experiments were conducted only within the ImageNet collection however. Similar results were obtained by Chatfield et al. [CSVZ14] on different data.

Concerning the second DCNN transfer strategy (classical training with features produced by DCNNs), Razavian et al. [RASC14b] showed that it works very well too, for several test collection, some of which are close to ImageNet and some of which are quite different both in terms of visual contents and in terms of target concepts. They also showed that this type of semantic features can be successfully used both for categorization tasks and for retrieval tasks. Finally, they showed that in addition to the score values produced by the last layer, the values corresponding to the output of all the hidden layers can be used as feature vectors. The semantic level of the layers output values increases with the layer number from low-level, close to classical engineered features for the first layers, to fully semantic for the last layers. Their experiments showed that using the last but one and last but two layers' outputs generally gives the best results. This is likely because the last layers contain more semantic information while the last one has lost some useful information as it is tuned to different target concepts. There is generally no equivalent to the output of the hidden layers in classical learning methods (e.g. SVMs) and these can only produce the final detection scores as semantic features.

Many variants of the "classical" approach exist. Most of them consist in a feature extraction step followed by a classification step. As several different features can be extracted in parallel and different classification methods can also be used in parallel, a step has to be considered. Fusion is called "early" when it is per-

formed on extracted features, "late" when it is performed on classification scores or "kernel" when it is performed on computed kernel within the classification step (for kernel-based methods); many combinations can also be considered.

A very large variety of engineered features has been designed and used for multimedia classification. Some of them are directly computed on the whole image (e.g. color histograms), some of them are computed on image parts (e.g. SIFTs) [Low04]. In the latter case, the locally extracted features need to be aggregated in order to produce a single fixed-size global feature. Many methods can be used for that, including the "bag of visual words" (BoW) approach [SZ03b] [CBDF04] or the Fisher Vector (FV) [SPMV13] and similar ones like Super Vectors (SV) and Vectors of Locally Aggregated Descriptors (VLAD) [JPD+12] or Tensors (VLAT) [PG13]. Some of them may reach their maximum efficiency only when they are highly dimensional, typically the FV, VLAD and VLAT ones. Two different strategies can be considered for dealing with them: either use linear classifiers combined with compression techniques [SPMV13] or using dimensionality reduction techniques combined with non-linear classifiers [SDQ15]. In the case of video indexing, engineered features have been proposed also for the representation of audio and motion content.

The comparison of methods presented here has been conducted in the context of the Semantic Indexing (SIN) task à TRECVid [SOK06][OAM+15]. It differs from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [RDS+15a] in many respects. Indeed, the indexed units are video shots instead of still images. The quality and resolution of the videos are quite low (512 and 64 kbit/s for the image and audio streams respectively). The target concepts are different: 346 non-exclusive concepts with generic-specific relations. Many of them are very infrequent in both the training and test data. The way the collection has been built is also very different. In ImageNet, a given number of sample images have been selected and checked for each target concept resulting in a high quality and comparable example set size for all concepts. In TRECVid SIN, videos have been randomly selected from the Internet Archive completely independently of the target concepts; the target concepts have been annotated a posteriori resulting in very variable number of positive and negative examples for the different concepts. Most of the concepts are very infrequent and also not very well visible. Compared to ImageNet, the positive samples are much less typical, much less centered, of smaller size and with a much lower image quality. The task is therefore much more difficult than the ILSVRC one but it may also be more representative of indexing and retrieval tasks "in the wild". An active learning method was used for driving the annotation process for trying to reduce the imbalance class effect in the training data and also ensure a minimum number of positive samples for each target concept [AQ08b]. The resulting annotation is sparse (about 15% in average) and consists in 28,864,844 concept $\times$ shots judgments. All of these differences probably explain why training DCNNs directly on TRECVid SIN data gives much poorer results than on ImageNet data and why the two considered adaptation strategies are needed (or perform much better) in this case.

4.3 Methods

This section presents the various elements used for the evaluation: features (or descriptors), classifiers and fusion methods, as well as several other processing steps used for further improving the overall system performance: use of multiple frames, feature optimization, temporal re-scoring and conceptual feedback.

4.3.1 Engineered features

For the engineered features, a series of features is used, which contains 13 different feature types contributed and shared by the participants of the IRIM group of the French GDR ISIS [BGP+15]. They vary significantly in terms of performance but several of them were quite competitive and at the level of the state-of-the-art before the availability of DCNN-based features. Here is a description of the six best performing types:

CEALIST/bov_dsiftSC_[8192|21504] : bag of visual terms [SL12]. Dense SIFT are extracted every 6 pixels. The codebook of size 1024 is built with K-means. Bags are generated with soft coding and max pooling. The final signatures result from a three-level spatial pyramid with two variants: $1024 \times (1 + 2 \times 2 + 1 \times 3) = 8192$ and $1024 \times (1 + 2 \times 2 + 4 \times 4) = 21504$ dimensions.

ETIS/global_<feature>[<type>]x<size> : (concatenated) histogram features [GCPF08], where:

- <**feature**> is chosen among lab (CIE Lab colors) and qw (quaternionic wavelets with 3 scales and 3 orientations)
- <**type**> can be m1x1 (histogram computed on the whole image), m1x3 (histogram for 3 horizontal parts) or m2x2 (histogram on 4 image parts)
- <**size**> is the dictionary size: 256, 512 or 1024.

For instance, with <type>=m1x3 and <size>=256, the final feature vector has $3 \times 256 = 768$ dimensions.

ETIS/vlat_<desc type>_dict<dict size>_<size> : compact Vectors of Locally Aggregated Tensors (VLAT [PG13]), where:

- <**desc type**> is low-level descriptor type, for instance hog6s8 = dense histograms of gradient every 6 pixels, 8×8 pixels cells.
- <**dict size**> is the dictionary size.
- <**size**> is the size of the feature for one frame: 4096 dimensions.

LIG/opp_sift_<method>[_unc]_1000 : Bag of Words, Opponent SIFT, generated using Koen Van de Sande's software [vdSGS10a]: 1000 dimensions (384 dimensions per detected point before clustering; clustering on

535117 points coming from 1000 randomly chosen images). <**method**> method is related to the way by which SIFT points are selected: **har** corresponds to a filtering via a Harris-Laplace detector and **dense** corresponds to a dense sampling; the versions with **__unc** correspond to the same with fuzziness introduced in the histogram computation.

LIRIS/OCLPB_DS_4096 : Dense sampling OCLBP [ZBC11] Bag of Words descriptor with 4096 k-means clusters. The orthogonal combination of local binary patterns (OCLBP) is extracted to reduce original LBP histogram size and at the same time preserve information on all neighboring pixels. Instead of encoding local patterns on 8 neighbors, encoding is performed on two sets of 4 orthogonal neighbors, resulting in two independent codes. Concatenating and accumulating two codes leads to a final 32 dimensional LBP histogram, compared with original 256 dimensions. 4096-dimensional bag-of-words descriptors are finally generated using a pre-trained dictionary.

LISTIC/SIFT_* : Bio-inspired retinal preprocessing strategies is applied before extracting Bag of Words of Opponent SIFT features (details in [SBL13]) using the retinal model from [BCDH10]. Features extracted on dense grids on 8 scales (initial sampling=6 pixels, initial patch=16x16pixels), using a linear scale factor 1.2. K-means clustering is used for producing dictionaries of 1024 or 2048 visual words. The proposed descriptors are similar to those from [SBL13] except that multi-scale dense grids are used. Despite showing equivalent mean average performance, the various pre-filtering strategies present different complementary behaviors that boost performances at the fusion stage [SBL14b].

Early fusions of features of the same type were also taken into account.

4.3.2 Learned or semantic features

The following learned or semantic feature types were considered:

XEROX/ilsvrc2010: Attribute type descriptor constructed as a vector of classification scores obtained with classifiers trained on external data with one vector component per trained concept classifier. For XEROX/ilsvrc2010, 1000 classifiers were trained using annotated data from the ILSVRC 2010 challenge. Classification was done using Fisher Vectors [SPMV13].

XEROX/imagenet10174: Attribute type descriptor similar to XEROX/ilsvrc2010 but with 10174 concepts trained using the ImageNet annotated data.

LIG/alexnet1000: The AlexNet model trained on the ImageNet data only [KSH12] has been applied unchanged on the TRECVID key frames, both on training and test data, providing detection scores for 1000 concepts. These are accumulated into a 1000 dimension semantic feature vector.

LIG/alexnet_fc<level>_<size> : This descriptor corresponds to the LIG/alexnet1000 one and was also computed using the AlexNet model [KSH12] but is made of the values of the output of the last three hidden layers (fully connected fc5, fc6 and fc7). The vector size if of 4096 dimensions for fc6 and fc7 and of 43,264 for fc5.

LIG/googlenet_pool5b_1024 : This descriptor is obtained by extracting the output of the penultimate layer (pool5) of the GoogLeNet model [SLJ+14] with 1024 dimensions.

LIG/vgg_all_fc8 : This descriptor is obtained by extracting the output of the last layer of the VGG19 model [CSVZ14][SZ14] before the last normalization stage and also has 1000 dimensions.

The Xerox features are of semantic type by construction but they also belong to the engineered type as the low-level features they rely upon (Fisher Vectors) have been explicitly designed using human expertise rather than having been built from learning like the DCNN-based features. Early fusions of features of the same type were considered.

4.3.3 Use of multiple key frames

All features (except audio and motion ones) have been computed on the reference key frames provided in the master shot segmentation. Additionally, some of them have been computed on all the I-frames extracted from the video shots (typically one every 12 video frames and about 13 per shot in average). Classification scores are computed in the same way both for the regular key frames and all the additional I-frames; a max pooling operation is then performed over all the scored frames within a shot [SWG+05]. This max pooling operation is performed right after the classification step and before any fusion operation (though it would probably have been better to postpone it after).

4.3.4 Feature optimization

The feature (descriptor) optimization consists of a PCA-based dimensionality reduction with pre- and post-power transformation [SDQ15]. Optionally, a L_1 or L_2 unit length normalization can also be performed before the PCA-based dimensionality reduction. This method allows to simultaneously reduce the dimensionality of the feature vectors (by factors from 2 to 50) and significantly improves the classification performance. It can also be used to transform feature vectors not naturally suited for the use of Euclidean distance into feature vectors suited for it, greatly simplifying (or speeding up) the classification process.

4.3.5 Classification

Two different classifiers have been used and their predictions were fused producing a globally better result [SDH+14]:

KNN: The first classifier is kNN-based. It is directly designed for simultaneously classifying multiple concepts with a single nearest neighbor search. A score is computed for each concept and for each test sample as a linear combination of 1's for positive training samples and of 0's for negative training samples (non-annotated or skipped samples are ignored) with weights chosen as a decreasing function of the distance between the test sample and the training sample. As the nearest neighbor search is done only once for all concepts, this classifier is quite fast for the classification of a large number of concepts. It is generally less good than the SVM-based method but it is much faster.

MSVM: The second one is based on a multiple learner approach with SVMs. The multiple learner approach is well suited for the imbalanced data set problem [SQ10] [SQ15], which is the typical case in the TRECVid SIN task, in which the ratio between the number of negative and positive training samples is generally higher than 100:1.

FUSE: Fusion between the two above classifiers. The fusion is simply done by a MAP weighted average of the scores produced by the two classifiers. Their output is naturally (or by construction) normalized in the [0:1] range. Even though the MSVM classifier is often significantly better than the KNN one, the fusion is frequently even better, probably because they are very different in terms of information type they capture. The MAP values used for the weighting are obtained by a two-fold cross-validation within the development set.

Classification scores are always produced both on the development set (by cross-validation) and on the test set (by prediction) so that they can be used in the higher levels of fusion.

4.3.6 Fusion

Several early and late fusions of features of the same type were also considered [SBL+14a]. Hierarchical late fusion was made successively on:

- variants of the same feature,
- variants of classifiers on results from the same features,
- different types of features,
- the selection of groups of features.

4.3.7 Temporal re-scoring and conceptual feedback

At the end, temporal re-scoring (re-ranking) and conceptual feedback are performed. Temporal re-scoring consists of modifying the detection score of a given video shot for a given concept according to the detection scores of adjacent video

shots for the same concept [SQ11]. Conceptual feedback consists in modifying the detection score of a given video shot for a given concept according to the detection scores of other concepts for the same video shot [HMQ15]. This is done by building an additional semantic feature constituted of the prediction scores (by cross-validation within the development set) and adding it to the pool of other engineered or learned features for inclusion in the global fusion process.

4.4 Evaluation on the TRECVid 2013-2015 semantic indexing task

Experiments were conducted on the 2013, 2014 and 2015 issues of the TRECVid semantic indexing task [OAM+15]. This task is defined as follows: "Given the test collection, master shot reference, and concept definitions, return for each target concept a list of at most 2000 shot IDs from the test collection ranked according to their likelihood of containing the target". In the 2013-2015 test collections each include about 200 hours of video contents from the IACC collection; they respectively include 112,677, 107,806 and 113,467 video shots which are the units to be indexed. Participants are asked to provide results for a set of 60 concepts, of which only a subset was actually evaluated (38, 30 and 30 respectively in 2013, 2014 and 2015). The Mean (Inferred) Average Precision (MAP) is used as the official metric. The average of these measures over the three years was also considered as it is expected to be more stable.

Additionally, a development set with the annotation of 346 concepts (including the 60 ones for which results should be submitted) was provided to the participants for training their systems [AQ08b].

4.4.1 Engineered features versus semantic and learned features

In this section, a comparison is presented of the performance of engineered features and semantic and learned features. For the engineered features, a series of features is used, which is composed of shared features by the participants of the IRIM group of the French GDR ISIS [BGP+15]. As the IRIM participants did not all provide prediction scores on the I-frame set, results are shown here only for the key frames (only one per shot).

Table 4.1 shows the performance of several types of engineered features. Performance is shown for the six best groups of feature types as well as the fusion of the seven less good ones and the overall fusion. In several cases, the result is shown for already a combination of variants of the same feature type, for instance corresponding to a pyramidal image decomposition. Performance is given as the Mean (Inferred) Average Precision on the 2013, 2014 and 2015 editions of the TRECVid SIN task as well as their mean. The task was a bit harder in 2014 than in 2013 and a bit harder still in 2015 than in 2014. This is because the set of evaluated concepts was different, including more complex and difficult ones. One

Table 4.1: Performance of low-level engineered features

Feature type	2013	2014	2015	Mean
IRIM bottom seven fused	0.1890	0.1444	0.1311	0.1548
LIRIS OC-LBP	0.1156	0.0811	0.0773	0.0915
LIG BoW opponent SIFT	0.1423	0.1104	0.0981	0.1169
CEA-LIST pyramidal BoW dense SIFT	0.1605	0.1203	0.1107	0.1304
ETIS pyramidal BoW lab and qw	0.1563	0.1191	0.1171	0.1307
LISTIC BoW retina SIFT	0.1663	0.1255	0.1122	0.1346
ETIS VLAT	0.1801	0.1369	0.1201	0.1457
IRIM all engineered fused	0.2300	0.1786	0.1554	0.1879

can see that the fusion of all features does significantly better than the single best one. Also, fusion of the seven least performing IRIM feature types does slightly better that the best individual one.

Table 4.2: Performance engineered and learned features

Feature type	2013	2014	2015	Mean
IRIM all engineered fused	0.2300	0.1786	0.1554	0.1879
Xerox ILSVRC 1000 features	0.2190	0.1749	0.1539	0.1824
Xerox ImageNet 10174 features	0.2258	0.1839	0.1570	0.1886
Xerox semantic features	0.2291	0.1862	0.1613	0.1920
IRIM and Xerox fused	0.2573	0.2070	0.1793	0.2145
AlexNet fc5	0.2214	0.1781	0.1610	0.1868
AlexNet fc6	0.2330	0.2001	0.1751	0.2027
AlexNet fc7	0.2277	0.1968	0.1717	0.1985
AlexNet out	0.2114	0.1925	0.1703	0.1910
GoogLeNet pool5	0.2633	0.2234	0.2062	0.2309
VGG-19 out	0.2550	0.2283	0.2042	0.2291
Learned (DCNN) features fused	0.2995	0.2637	0.2350	0.2660
Engineered and DCNN fused	0.3190	0.2849	0.2553	0.2863

Table 4.2 shows the performance of engineered and learned features as well as of their combinations. The first row reproduces the result of the fusion of all the IRIM engineered features from Table 4.1. The next three rows show the performance of the two Xerox semantic features as well as their fusion. Both have a performance similar to the performance of the fusion of all IRIM features and their fusion has an even higher performance. The Xerox semantic features are very good thanks to their state-of-the-art use of Fisher Vector and to their training on ImageNet data, which the other IRIM features did not benefit from. The next row shows the performance of the fusion of IRIM and Xerox features,

which is significantly higher than that of each of them taken separately. This performance is the best that could be achieved using only engineered features (as Xerox features also fall in this category even though they include some learning).

The next six rows of Table 4.2 show the performance obtained with the features extracted from the AlexNet, GoogLeNet and VGG deep neural networks. It is detailed for the four last layers in the case of AlexNet. The best DCNN feature for each of the three networks already has a performance comparable to the IRIM/Xerox fusion or even higher. Again, this is due to the use of the ImageNet data and to the very good effectiveness of DCNNs. The second to last row shows the performance of the fusion of the best three DCNN features, which is significantly higher than that of each of them taken separately. This indicates that the three networks extract complementary information. Finally, the last row shows the performance of the fusion of non-DCNN-based features and of DCNN-based features. This performance is once again significantly higher than that of both of them taken separately, even if the performance of non-DCNN-based features is significantly lower than that of DCNN-based features, indicating that engineered features are still useful, even with a lower overall performance.

4.4.2 Partial DCNN retraining versus use of DCNN layer output as features

Several trials were made for retraining the last layers of the pre-trained AlexNet, GoogLeNet and VGG-19 implementation using the Caffe framework. The following was tested: the retraining of the last one, last two or last three layers. Much care was taken to try and select the optimal training parameters in each case. Actually, due to the design of the inception module in the GoogLeNet architecture, it was not easy to retrain only the last two or last three layers so, alternatively, the two layers were retrained by adding another last fully connected layer. A complete three layers retraining or any equivalent was not tested. The best performance was obtained by cross-validation when retraining only the last two layers for AlexNet and only the last layer for GoogLeNet and VGG-19.

For these features, the evaluation was done both using only one key frame per shot and using additionally all the available I-frames within the shot. In both cases, the training was done using only the key frames as the collaborative annotation was done mostly only on them while the assessment for the evaluation was done on the basis of the full shots [OAM+15].

Table 4.3 shows the performance obtained with the AlexNet, GoogLeNet and VGG-19 implementations as well as for the fusion of their predictions. The first (resp. second) half of the table shows results using only the key frames (resp. using the key frames and the I-frames) for the prediction. Results are displayed using the classical KNN/MSVM learning approach applied to the best extracted features for each implementation and by retraining these implementations. It can be observed that:

- the classical KNN/MSVM learning consistently perform better than the retraining of the last layers. This may be because the last layers actually

Table 4.3: Partial DCNN retraining versus use of DCNN layer outputs as features

	2013	2014	2015	Mean
AlexNet fc6 + classifiers	0.2330	0.2001	0.1751	0.2027
GoogLeNet pool5 + classifiers	0.2633	0.2234	0.2062	0.2309
VGG-19 out + classifiers	0.2550	0.2283	0.2042	0.2291
Classifiers fused	0.2995	0.2637	0.2350	0.2660
AlexNet, 2 layers retrained	0.2172	0.1834	0.1647	0.1884
GoogLeNet, 1 layer retrained	0.2331	0.2016	0.1926	0.2090
VGG-19, 1 layer retrained	0.2230	0.1948	0.1778	0.1985
Retrained fused	0.2768	0.2406	0.2208	0.2460
AlexNet fc6 + classifiers with I-frames	0.2553	0.2631	0.2233	0.2472
GoogLeNet pool5 + classifiers with I-frames	0.2953	0.2911	0.2733	0.2865
VGG-19 out + classifiers with I-frames	0.2828	0.2958	0.2657	0.2814
Classifiers fused with I-frames	0.3213	0.3296	0.3004	0.3170
AlexNet, 2 layers retrained with I-frames	0.2534	0.2579	0.2216	0.2442
GoogLeNet, 1 layer retrained with I-frames	0.2721	0.2787	0.2594	0.2700
VGG-19, 1 layer retrained with I-frames	0.2608	0.2675	0.2433	0.2571
Retrained fused with I-frames	0.3107	0.3170	0.2895	0.3057

implement only a one or two-layer perceptron, because there is not enough training data for a good neural network learning (while KNN and MSVM are more robust to this) and/or because they have difficulties with highly imbalanced training data (cost sensitive training was also tried but brought no improvement);

- the differences between 2013, 2014 and 2015 collections and between using or not I-frames are smaller in the case of retrained networks, indicating a better generalization capability despite a lower global performance.

4.4.3 Combining with improvement methods

Considering the same features, classifiers and fusion methods, several methods can be used to further improve the overall system performance. The three following ones were evaluated: the temporal re-scoring method proposed by Safadi et al. [SQ11], the conceptual feedback method proposed by Hamadi et al. [HMQ15], and the use of multiple frames proposed by Snoek et al. [SWG+05].

As previously mentioned, only a few of the IRIM engineered features were computed by the participants on the I-frame set; the Xerox features were not available either on this set. It was therefore not possible to evaluate and compare all the combinations and some fusions are only partial. Table 4.4 shows the effect of the temporal re-scoring (TRS) and conceptual feedback (CF) methods for the fusions of the engineered (IRIM and Xerox) features, the DCNN-based features and their combinations (All). The effect of additionally using the I-frames is also

Table 4.4: Effect of improvement methods: temporal re-scoring (TRS), conceptual feedback (CF) and use of multiple frames (I-frames)

	2013	2014	2015	Mean
IRIM and Xerox fused	0.2573	0.2070	0.1793	0.2145
IRIM and Xerox with TRS	0.2691	0.2207	0.1822	0.2239
IRIM and Xerox with TRS and CF	0.2844	0.2474	0.2013	0.2443
DCNN features fused	0.2995	0.2637	0.2350	0.2660
DCNN features with TRS	0.3216	0.2903	0.2491	0.2869
DCNN features with TRS and CF	0.3288	0.3021	0.2533	0.2947
DCNN features with I-frames fused	0.3213	0.3296	0.3004	0.3170
DCNN features with I-frames with TRS	0.3293	0.3346	0.2974	0.3204
DCNN features with I-frames with TRS and CF	0.3421	0.3416	0.2935	0.3257
All features fused	0.3190	0.2849	0.2553	0.2863
All features with TRS	0.3343	0.3039	0.2625	0.3002
All features with TRS and CF	0.3407	0.3151	0.2670	0.3075
All features with I-frames fused	0.3408	0.3265	0.2917	0.3196
All features with I-frames with TRS	0.3473	0.3365	0.2938	0.3258
All features with I-frames with TRS and CF	0.3539	0.3460	0.2933	0.3310

shown, except in the case of the engineered features since not enough of them were available for doing better that using the key frames alone. These were, however, included in the "All" fusion when possible. It can be observed that:

- all three improvement methods are always effective, even when combined though the "TRS" and "I-frames" ones do not accumulate well; this is probably because both search information in the neighborhood, either within the current shot or within adjacent shots and such information may be redundant;

- fusing engineered features and DCNN-based features always lead to an improvement, even if engineered ones are less good and even if less of them were available in the I-frames case.

The five combinations with TRS and CF were the LIG (or Quaero) official submissions with the respective identifiers: 2C_M_A_Quaero.13_1, 2C_M_D_LIG.15_4, 2C_M_D_LIG.15_2, 2C_M_D_LIG.15_3 and 2C_M_D_LIG.15_1. For the 2015 issue of the semantic indexing task, the best MAP was of 0.3624. However, the participant who obtained this result used additional annotations that were not shared with other TRECVid participants. The following participants, ranked second, third and fourth, obtained best MAPs of 0.3086, 0.2987 and 0.2947. The 2C_M_D_LIG.15_1 submission ranked as fifth with a MAP of 0.2933. The fourth participant was the IRIM group that used a submission very close to the 2C_M_D_LIG.15_1 one (differing only in the last level of late fusion).

4.5 Evaluation on the VOC 2012 object classification task

The DCNN features and KNN/MSVM combination were also tested on the VOC 2012 classification competition. The official deadline for this competition has passed but the annotations on the test set were kept hidden and an evaluation server is left permanently opened, allowing for new participation in conditions similar to those of the original competition. The submission count is limited so that no tuning on the test set is possible. The goal of the task is, for each of twenty predefined classes, to predict the presence/absence of an example of that class in the test image. More generally, a classification with a real value is expected, allowing to sort the test samples according to their likeliness of containing an example of the target class. The official metric is the average precision (AP) by concept and the overall mean average precision (MAP) over the 20 target classes [EEVG+15b].

A single submission was made with a single feature for the KNN/MSVM classification. This feature is an early fusion of the AlexNet fc6 layer output (last but two layer, 4096 dimensions), the GoogLeNet pool5 output (last but one layer, 1024 dimensions), and the final output of the VGG-19 network (1000 dimensions). These are the same as those used in the TRECVid semantic indexing task, and the same feature optimization 4.3.4 parameters were used, reducing the dimensionality of the DCNN features to 662, 660 and 609 respectively. Early fusion is then performed by concatenating the three optimized and normalized features resulting in a unique feature of 1931 dimensions. A second feature optimization step is performed reducing it further to 294 dimensions. This is done again with the same parameters as those computed for the TRECVid semantic indexing task. No feature optimization was performed on the VOC 2012 data.

The KNN and MSVM classifiers were trained only using the development data ("train" and "val" sets) and annotations. No other data and annotation was used directly in the training. ImageNet data and annotations were used indirectly for training the DCNN systems from which the features were extracted. TRECVid SIN data and annotations were used indirectly for optimizing these features and their early fusion. However, these data and annotations were not used directly for training the classifiers on the VOC 2012 data. Therefore, the submission was made in the "comp1" category as other teams did in similar conditions. This submission obtained a MAP of 85.4% while the second best performance obtained a MAP of 82.9%.

4.6 Conclusion

In this chapter, the comparison was made between the use of "traditional" engineered features and learned features for content-based semantic indexing of video documents. An extensive comparison was made of the performance of learned features with traditional engineered ones as well as with combinations of them. Comparison was made in the context of the TRECVid semantic indexing task.

The results confirm those obtained for still images: features learned from other training data generally outperform engineered features for concept recognition. Additionally, it seems that directly training KNN and MSVM classifiers using these features does better than partially retraining the DCNN for adapting it to the new data. Even though the learned features performed better that the engineered ones, the fusion of both of them still perform significantly better, indicating that engineered features are still useful, at least in this case. Additionally, the improvement methods based on temporal re-scoring, on conceptual feedback and on the use of multiple frames per shot are effective on both types of features and on their combination. Finally, the combination of DCNN features with KNN and MSVM classifiers was applied to the VOC 2012 object classification task where it currently obtains the best performance with a MAP of 85.4%.

Chapter 5

Deep learning for speaker identification

5.1 Introduction

In this chapter, a further application of deep neural networks is investigated, namely the use of convolutional neural networks (CNN) for speaker identification. Contrary to the classical approach of most speaker identification methods, which are based on the MFCC features (or on a modified version of them), here the CNN is used with raw spectrograms. This enables us to treat the speech signal in its 2D representation, i.e. just like an image. This could create an opportunity to have a more generalized CNN model, which could be applied to a wider range of signal processing related tasks. Additionally, a challenging dataset was selected, containing both noise and unbalanced amount of speaker data. This CNN approach is compared to more commonly used methods. Despite the lower CNN performance, the use of this deep complementary features in fusion improves on the state-of-the-art.

In the past few years, Convolutional Neural Networks (CNN) became widely used in image related domains providing state-of-the-art performance [SLJ+14]. At the same time Deep Neural Networks (DNN) were being applied more and more to mono-dimensional signals for tasks like language recognition [MZN+14], speech recognition [DLH+13] or speaker identification [RRD15b]. Lately, there was an increasing number of studies trying to address some of the related tasks (notably automatic speech recognition) with the use of CNN based systems with only spectrograms as input [GHT+14, UW15]. However, such systems have not yet been widely explored for speaker identification. The work done and presented in this chapter tries to give additional insight into the efficient use of CNN for this particular biometric task. The current limitations are also mentioned and analyzed.

The structure of this chapter is as follows. Section 5.2 gives the overview of the baseline methods for speaker identification while Section 5.3 describes the structure of the neural networks used. This is followed by the presentation of the experimental framework in Section 5.4. The results are presented in Section 5.5. Section 5.6 contains the concluding remarks.

5.2 Baseline speaker identification systems

Gaussian Mixture Model-Universal Background Model (GMM-UBM) [RQD00] and Total Variability Space (TVS) [DKD+11] speaker recognition systems are used in this study as baselines. The TVS setting and description follow the setup used in Chapter 3. For convenience, the introduction to the algorithm as well as the parameters used for these experiments are repeated here.

In the GMM-UBM approach, a Universal Background Model (UBM) is first trained on speech features extracted from multiple speakers using the Expectation-Maximization (EM) algorithm. Speaker-specific models are then obtained using Maximum a Posteriori (MAP) mean adaptation. Similarity scoring is done by calculation of log-likelihood ratio (LLR) on these models. Given a sequence of feature vectors X extracted from a test segment, LLR is computed as $\Lambda(X) = log\ p(X|\lambda_{hyp}) - log\ p(X|\lambda_{ubm})$, where λ_{hyp} and λ_{ubm} represent speaker-specific GMM and UBM model respectively.

After adaptation, models can also be represented as high-dimensional supervectors of means of distributions. These supervectors can be represented as low-dimensional identity vectors (i-vector) using factor analysis. In this approach, mean supervectors M_s and M_0 representing speaker-specific model and UBM respectively are extracted, and an i-vector, w_s, is calculated using $M_s = M_0 + Tw_s$. The low-rank rectangular matrix, T, representing the variability space of the i-vectors, is learned in an unsupervised manner using Expectation-Maximization (EM) algorithm. In case of having multiple tracks for speaker modeling, i-vectors extracted from each speech track are typically averaged and the average i-vector is used as the speaker model. Scoring can be done with cosine similarity metric, while pre-processing i-vectors before scoring can result in better performance.

In this setup, speaker identification is done by scoring a test segment versus all the speaker models. The speaker identity regarding the highest similarity score is chosen as the result of the identification test.

A UBM consisting of 1024 gaussians is trained on the training data for both systems. T matrix is then trained on the segmented training data. Segmentation outputs of conventional BIC-criterion [DW00] are used. The dimension of output i-vectors is set to 500. MSR Identity Toolbox [SSH13] is used for all experiments. Similarity scoring is done by cosine similarity scoring between the test segment and the i-vector representing target identity. Length normalization [GREW11] is used for an increased performance.

5.3 Convolutional neural network structure

5.3.1 Initial approach and tests

In Figure 3.2 the general way in which the CNN algorithm is applied can be seen. For any given speech segment (Fig. 3.2a) the spectrograms are first extracted. Because they have a fixed size there are usually several overlapping spectrograms representing each segment. Next (Fig. 3.2b) , each spectrogram is fed to the convolutional neural network separately. This in turn produces an individual

vector of potential speaker identities for every input (Fig 3.2c). Finally, to obtain a single vector for the speech track, the individual vectors are averaged.

Several different architectures were tested in these experiments. Here, the most promising initial setup is presented along with the justification for that choice. Most commonly used CNN architectures (such as AlexNet [KSH12]) may be too complex for the given task. First of all, the input images have a smaller resolution and are grayscale. There is also far less diversity in terms of observable patterns and textures than in datasets containing images of everyday object, animals, etc., such as ImageNet [RDS+15b]. Therefore, the setups used for ordinary images may have too many redundant parameters, leading to unnecessarily long training times and the risk of overfitting.

Initially, the input data size was wider with the spectrogram dimensions being equal to 137×129, which is equivalent to 0.685 seconds. However, this approach caused an unnecessarily large overlap between neighboring spectrograms. Also, the initial system was trained on a training set containing 375 speakers, around 800k spectrograms and close to 3200 speech segments. The final results were obtained using an extended set, which details are described in Section 5.4.1.

The initial structure of the net can be seen in Table 5.1. The input to the net is a central crop of the input spectrogram.

Name	Type	Patch size /Stride	Output size
input: 128×128 grayscale image of a spectrogram			
conv1	convolution	$3 \times 3/1$	$126 \times 126 \times 64$
pool1	ave pooling	$2 \times 2/2$	$63 \times 63 \times 64$
conv2	convolution	$3 \times 3/1$	$61 \times 61 \times 128$
pool2	ave pooling	$2 \times 2/2$	$30 \times 30 \times 128$
conv3	convolution	$3 \times 3/1$	$28 \times 28 \times 256$
conv4	convolution	$3 \times 3/1$	$26 \times 26 \times 256$
pool3	ave pooling	$2 \times 2/2$	$13 \times 13 \times 256$
conv5	convolution	$3 \times 3/1$	$11 \times 11 \times 512$
conv6	convolution	$3 \times 3/1$	$9 \times 9 \times 512$
pool4	ave pooling	$2 \times 2/2$	$4 \times 4 \times 512$
fc6	full connected		$1 \times 1 \times 4096$
fc7	full connected		$1 \times 1 \times 4096$
fc8	full connected		$1 \times 1 \times 375$

Table 5.1: The initial structure of the network tested on a smaller subset of the data containing 375 individual speakers.

The test set in the initial evaluation was composed of around 400k individual spectrograms, which were divided into roughly 1850 segments of varying lengths. This was a closed set with 71 speakers appearing in both the train and the test set.

This introductory setting helped to give insight into the training of a convo-

lutional neural network for this particular task and served as a justification for some of the design choices in the final approach.

Epochs	3	6	9	12	15	18	21	24	27	30
Spectrogram	61.5	61.0	64.0	63.8	64.2	64.3	64.3	64.3	64.3	64.3
Segment	70.7	69.0	71.7	70.8	71.1	71.0	71.1	71.4	71.2	71.2

Table 5.2: The performance of the initial CNN system after a particular number of epochs. The accuracy is given for both individual spectrograms and the whole segments. The latter value should be compared to the baseline systems trained on MFCC features with GMM-UBM and i-vector systems having 73% and 74.4% accuracy, respectively.

The initial model was tested after every 3 epochs[1] and the results can be seen in Table 5.2. The convergence seems to be achieved quite quickly. After around 9 epochs no significant change in performance seems to take place. Therefore for the final evaluation and after accounting for an extended dataset, the training lasting 12 epochs was chosen.

At this stage the accuracy results for the CNN approach are not far from the baselines. This led to the belief that this system could benefit from additional training data. The final results on the full dataset can be found in Section 5.5. Even at this stage, however, the accuracy gain from fusion between i-vector (74.4%) and CNN (71.7%) scores is visible. For the mean of normalized scores from both these systems the accuracy is equal to 77.9% – well above the baseline score.

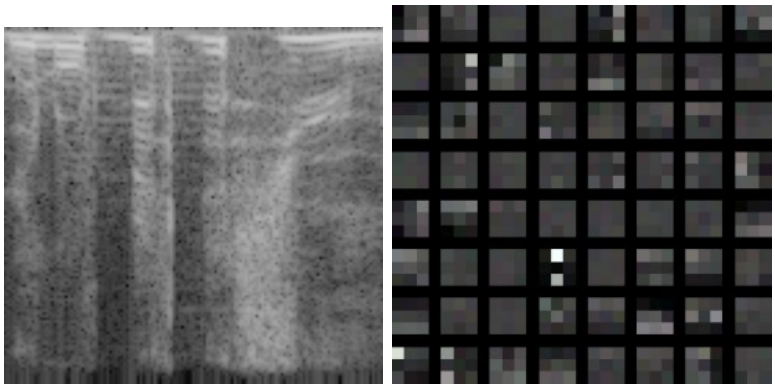
These observations led to the design and execution of a new set of experiments on a bigger and possibly more representative dataset. Additionally, some insight into the features extracted by the CNN network could be made based on the initial structure. They are presented in the following subsection.

5.3.2 Layer visualization

To better understand what is going on within the CNN structure it is possible to visualize some of its components. There are many ways to achieve this. One of the more insightful approaches is presented in [ZF14]. There it was made possible to map back to an image the response of the net even for the layers at the end of the net where the input vector was 1 dimensional. In the case of spectrograms this seems to not be necessary, because they are not easily readable to humans anyway (contrary to normal images containing everyday object for example). Instead, the visualization of the outputs of the first two convolution layers is presented. Their resolution is still high enough to see what the network is looking for.

Figure 5.1(a) is an example of a grayscale spectrogram of human speech that is used as an input to the CNN. Figure 5.1(b) presents the first convolution layer

[1]Where a epochs is a single pass through the whole training set.



(a) (b)

Figure 5.1: (a) An example of a spectrogram used as input to the network. (b) Visualization of the 64 filters from the first convolutional layer.

filters that are obtained through supervised training. The filters are small with the resolution of just 3×3. Note that only first layer filters are shown. The number of filters with each subsequent layer increases significantly and they become more and more abstract and hard to interpret. The result of the convolution operation between the input spectrogram and one of the filters can be seen in Figure 5.2, which shows corresponding feature maps. It shows that most of the filters tend towards some kind of an edge detector, be it horizontal, vertical or other. Some examples of low pass (which result in the blurring of the image) and high pass filters (making the detail more visible) are also present. Also, a lot of feature maps are dark, suggesting a low response. On the other hand some may be redundant.

This finding is in line with what can be seen when a CNN is trained on color images (see Chapter 2 for an example). With the exception that additionally there are also color blobs that are detected, but for a monochrome image this is not the case. Further results of the second convolution can be seen in Figure 5.3. The size of the feature map is reduced due to the 2×2 max pooling layer between the two convolution layers. As the input image passes through the network the filters become more specialized to detect particular shapes and pattern in an image. This can be seen in the output of the second convolution layer. Here, most filters seem to focus on the detection of vertical lines.

5.3.3 Final architecture

The network used in this study is inspired by the general design proposed in [SZ14] for image recognition. It was chosen as a starting point due to its relative structure simplicity and state-of-the-art performance. However, several changes were made in order to adapt it to this specific speaker identification task. The detailed structure can be found in Table 5.3.

The network was trained from scratch on a set of grayscale spectrogram images with non-square dimensions. The model was trained for around 12 epochs. Every convolutional layer was followed by a rectified linear unit (ReLU), which serves

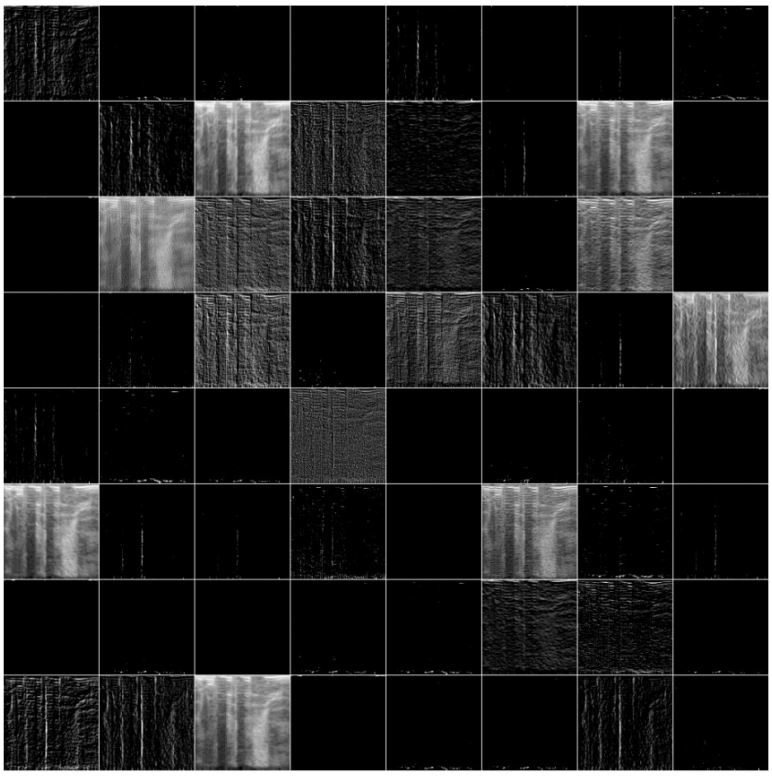


Figure 5.2: The output of the first convolutional layer in the network. These are the results of a convolution between the input spectrogram presented in Figure 5.1(a) and the respective filters seen in Figure 5.1(b).

as an activation function and is defined as $f(x) = \max(0, x)$. The first two fully connected layers (fc6, fc7) are followed by ReLU and dropout with the rate of 0.5. The output of the last fully connected layer (fc8) is used with the softmax function.

Different to the initial design for image recognition, the proposed structure has fewer convolutional layers (from 8 down to 5), however the filter size for the first two is expanded. Adding additional convolutional layers did not improve performance. Average pooling layers were chosen instead of max pooling. The input to the network is a 48×128 pixel grayscale image of a spectrogram. Due to the overlap between the images, no random cropping or rotation is used during training. Caffe framework [JSD+14] was used for training and testing the net.

The network gives predictions based on individual spectrograms. In order to be able to fuse the output with the output of the TVS system (which assigns a speaker identity for the whole speech segment), the spectrograms are mapped to bigger speech segments. The mapping is done by averaging the scores of every spectrogram contained within a given segment.

In Figure 5.5(a) an example of a spectrogram used for training is shown. Figure 5.5(b) represents the saliency map, i.e. a heatmap representing the most significant regions of the image used by the CNN to predict a given speaker. In this case it represents speaker with the highest response from the top layer. Note the heavy reliance on horizontal patterns, which stands in contrast to MFCC

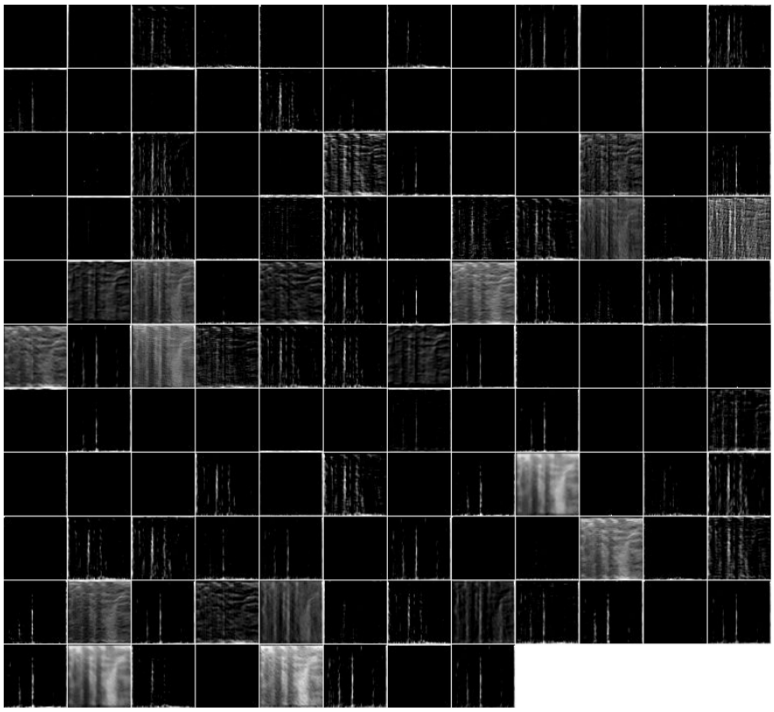


Figure 5.3: The output of a subset of filters after the second convolutional layer.

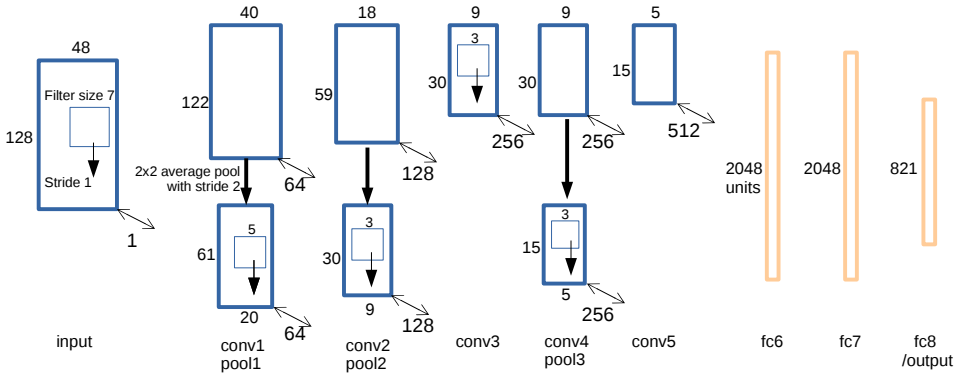


Figure 5.4: The visualization of the CNN used in this study. A spectrogram is taken as input and is convolved with 64 different filters (with the size of 7×7) at the first layer with the stride equal to 1. The resulting 64 feature maps are then passed through the ReLU function (not visible in the figure) and downsampled using average pooling. A similar process continues up to the fully connected layers (f6, which takes conv5 as input, and fc7) and the final output layer (corresponding to the number of speakers in the train set).

based methods that take into account the whole frequency spectrum. This was obtained by backpropagating the correct output from the last layer to the input layer in a similar way as it was done in [ZF14]. To put it into perspective and better understand the output Figure 5.6(a) shows a color image that may be used to train a CNN system for image recognition. The output of the net shown in Figure 5.6(b) is equivalent to Figure 5.5(b). In other words, it highlights

Name	Type	Patch size /Stride	Output size
Input: 48×128 grayscale image of spectrogram			
conv1	convolution	$7 \times 7/1$	$40 \times 122 \times 64$
pool1	ave pooling	$2 \times 2/2$	$20 \times 61 \times 64$
conv2	convolution	$5 \times 5/1$	$18 \times 59 \times 128$
pool2	ave pooling	$2 \times 2/2$	$9 \times 30 \times 128$
conv3	convolution	$3 \times 3/1$	$9 \times 30 \times 256$
conv4	convolution	$3 \times 3/1$	$9 \times 30 \times 256$
pool3	ave pooling	$2 \times 2/2$	$5 \times 15 \times 256$
conv5	convolution	$3 \times 3/1$	$5 \times 15 \times 512$
fc6	full connected		$1 \times 1 \times 2048$
fc7	full connected		$1 \times 1 \times 2048$
fc8	full connected		$1 \times 1 \times 821$

Table 5.3: The structure of the network.

the regions that help the network to determine that an image contains a given concept, in this case a cat.

5.3.4 Fusion

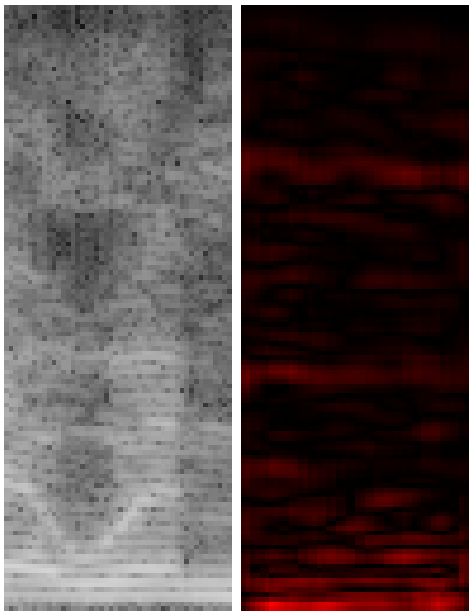
Fusion is often used to enhance results for speaker recognition systems, for example in [MLSF14]. Even if by itself a system gives inferior results, it still can help to improve the baseline performance. In this article, several attempts were made to fuse the CNN results with the output of the TVS system. Both early and late fusions were considered.

5.3.4.1 Late fusion

A standard late fusion of normalized predictions taken from both systems was proposed. A weighted sum of both outputs was also tested.

5.3.4.2 Duration-based fusion

This strategy was proposed to give the CNN scores higher weights for short duration segments and lower ones for the long speech segments. CNN seems to produce comparable results to the TVS system on short segments, while the difference in performance grows with increasing duration. Fusion on the longer segments also seems to be less beneficial. This is illustrated in the bottom plot of Figure 5.8. In this case, fusion was calculated as $s = (1 - \tanh(d))s_{cnn} + s_{ivec}$, where s corresponds to the scores provided by each system and d is the segment duration.



(a) (b)

Figure 5.5: (a) An example of a spectrogram used in this study. (b) A saliency map representing the networks response to this spectrogram.

(a) (b)

Figure 5.6: (a) A cat. (b) A saliency map representing the networks response to the picture of a cat.

5.3.4.3 Support Vector Machines

This strategy serves as early fusion, where a linear SVM is used for the final classification decision based on a concatenated normalized outputs of CNN's last hidden layer and i-vectors. Principal component analysis is applied to the CNN output in order to match the i-vector dimensionality (500 for each).

5.4 Experimental setup

5.4.1 Dataset

The REPERE corpus [GCM+12b] was used for evaluation. The dataset contains a set of videos from two French television channels (LCP and BFM). There are 7 types of videos, ranging from news shows, debates to celebrity gossip and culture programs. Only the audio track was used in the experiments.

The dataset is quite challenging. The recording takes place both inside a studio setting and outside in public and noisy environments. Apart from this, music if often played in the background during certain presentations or interviews. Additionally, there is a significant imbalance between speakers, with anchors and top politicians both being often over-represented in the dataset. Total amount of speech per speaker for speakers present in both train / test sets helps to illustrate this and it is shown in Figure 5.7. Normalized histogram of number of segments for each duration bin is also shown in Figure 5.8 for training and test data. It is important to mention that a big portion of speech segments fall below 2 seconds of speech.

Experiments were done in a closed-set manner, where all the speakers in the training data are used for training models, while performance is evaluated only on test segments from speakers overlapping between training and test data. There are 821 speakers available in the training data, from which only 113 are observed in the test data.

Training data includes 9377 speech segments from 148 videos, while the test data contains 2410 segments from 57 videos. Training data and test data contain around 22 hours and around 6 hours of active speech respectively.

5.4.2 Features

Two types of features are used in this study. First of all, the classical MFCC features, which are the most commonly used, when dealing with speech related tasks. As an alternative spectrograms are used. Their use and extraction for such tasks are not yet normalized, so the choice concerning the dimensions and frequency are somewhat arbitrary.

5.4.2.1 Mel-Frequency Cepstral Coefficients (GMM-UBM, TVS)

Energy feature and Mel-Frequency Cepstral Coefficients (MFCCs) of 19 dimensions are extracted every 10 ms with a window length of 20 ms. These features

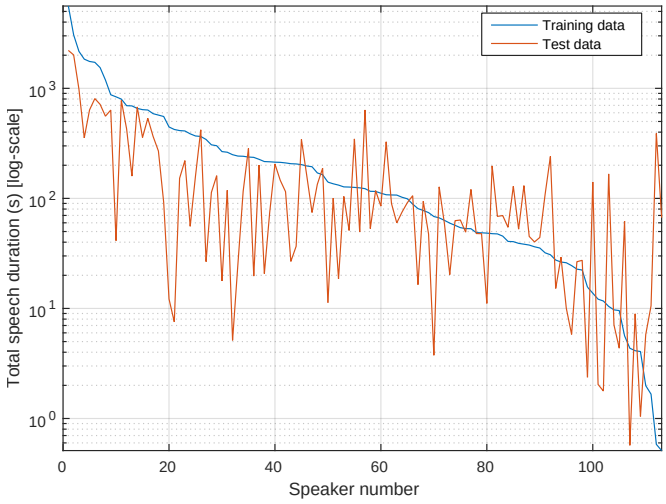


Figure 5.7: Total amount of speech per speaker for speakers present in both train / test sets of REPERE corpus. Speakers are sorted according to total speech duration in training set.

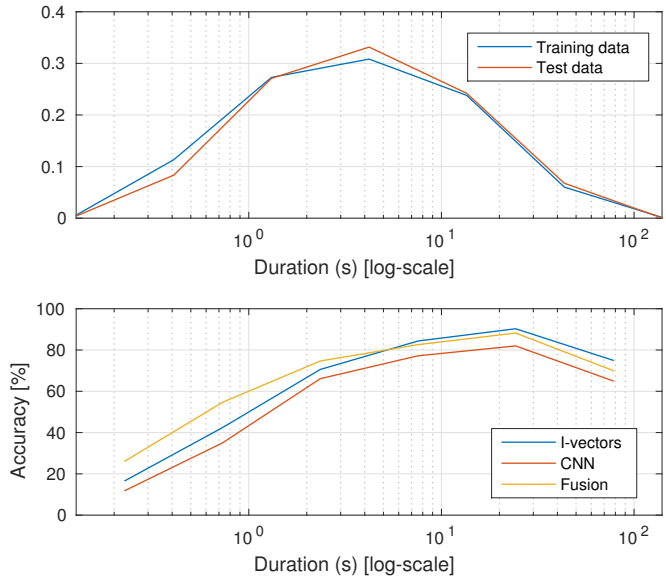


Figure 5.8: Normalized histogram of speech segments for different duration bins for training and test data from REPERE corpus on top, and test set accuracy of each system along with their late-fusion for corresponding duration bins in bottom.

along with their delta and delta-delta coefficients are concatenated. Static energy is used for silence removal using bi-gaussian distribution of frame log-energies. For each frame, a 59 dimensional feature vector is then obtained after application of feature warping [PS01] on remaining features.

5.4.2.2 Spectrograms (CNN)

Spectrograms were extracted on 240 ms duration with a frequency of 25 Hz. This results an overlap of 200 ms (83%) between neighboring spectrograms. For each spectrogram, first the corresponding audio segment was windowed every 5 ms with a window length of 20 ms. Then on each window, after applying Hamming window of 256 samples, log-spectral amplitude values were extracted. By discarding the symmetric part and the value corresponding to the highest frequency, a 48 by 128 matrix of values was obtained, which was utilized for input of the CNN as an image. Spectrograms containing speech from multiple speakers were discarded along with the ones containing no speech in both training and testing phases.

5.5 Results and discussion

Table 5.4 gives the accuracy for the baseline systems and the CNN. The segment accuracy for the CNN is generated as explained in section 5.3. We see that CNN is slightly lower in performance than baseline approaches for speaker identification. In Table 5.5 the results of fusion are presented. The last two columns give partial results for segments shorter and longer than 2 seconds, respectively. Apart from the standard accuracy, a duration based accuracy is also given, i.e. the duration of the data predicted correctly versus the total duration of the data.

Method	Accuracy	Trained on
CNN	67.41	Spectrorgams
PLDA	70.50	MFCC
GMM-UBM	71.16	MFCC
TVS	72.78	MFCC

Table 5.4: CNN and baseline accuracy (% on the test set) estimated at the speaker segments level.

Method	Stand. Acc.	Dur. Acc.	$<= 2s$ Acc.	$> 2s$ Acc.
CNN	67.41	76.00	40.93	76.32
TVS	72.78	83.74	48.99	81.58
SVM-CNN+TVS	69.05	75.27	51.63	75.41
CNN+TVS	75.89	83.61	58.45	82.27
durCNN+TVS	75.10	84.07	56.12	82.04

Table 5.5: Fusion results with standard accuracy and duration based accuracy (on test set).

The rather poor performance of the single CNN approach when compared to TVS may be attributed to several different factors. First of all, the unbalanced speaker dataset where some speakers (like high level politicians or news anchors and presenters) are heavily over represented, while others may appear for just a few seconds. The second factor could be connected to the nature of the corpus. Live and mostly spontaneous (especially in the case of debates) TV broadcasts usually come with significant noise (street noises, crowds, other voices) or background music. This may, in fact, disproportionally affect the raw spectrograms over the MFCC features.

A relatively low performance was given also by the PLDA approach, even though a grid search was done in order to choose the best hyper-parameters possible. This can be explained by the dependency of PLDA performance on the availability of a large training set (as discussed for example in [Aro14]). In the training data used in this study, only 375 speakers out of 821 had more than two segments, whereas thousands of multi session speakers are usually used for successful estimation of the PLDA hyper-parameters.

The late fusion approaches represented by CNN+TVS and durCNN+TVS seem to work much better than the early fusion based on the SVM. Both late fusion approaches were able to be better than both the CNN and the TVS. Based on the partial accuracy results, it seems that the main improvement of fusion is for the shorter speech segments. The duration based accuracy revels the underlining imbalance of the dataset, where the improvement of the number of segments correctly classified does not necessarily imply a higher duration score.

5.6 Conclusions

In this chapter, an approach was proposed which uses the output of a CNN network trained on spectrograms to improve the performance of a TVS system based on MFCC features. The tests were carried out on a broadcast TV dataset, which included such real-life issues like noisy environments and imbalance between speakers, with encouraging results.

Chapter 6

Conclusions and perspectives

In this last chapter, a final overview of the work done during this thesis is summarized with an emphasis on the proposed contributions. Also, at the end, some general perspectives based on the presented research are given.

6.1 Conclusion

Overall, the main topics of this thesis were active learning and deep learning applied in the context of multimedia documents. The emphasis was put on the applications involving images, video and audio.

The related work chapter started with the introduction of the classical approach to concept detection in images, namely engineered features (including low level descriptors and several aggregation methods) coupled with a classifier which is trained using supervised learning. This formula is then compared to the alternative deep learning approach. The basic building blocks of deep neural networks are then presented, focusing on the convolutional neural network and its components. Further on, different applications of the CNN architecture are explored (including feature extraction, multi-label prediction and localization). Afterwards, the active learning framework was presented. This included the most common methods and scenarios where this approach can be used. Several methods that use active learning in combination with unsupervised approach are also presented and discussed. An interesting approach and a potential source of labels is through label propagation, which has the means to make active learning even more cost effective. The exploration of the intersection between active learning and label propagation was one of the major points of this thesis.

6.1.1 Active learning

In terms of active learning and label propagation, the following contributions and observations were made during this thesis:

- An active learning framework was developed that incorporates label propagation to speed up the acquisition of labeled instances. This includes the evaluation of several different modalities (including speech and faces) as

well as the use of overwritten names as initial labels. Additionally, four different selection strategies were evaluated. According to the experimental evaluation the best approach involves the selection of biggest clusters with the probability corresponding to their size and using overlaid names as initial annotation when available.

- One of the more interesting findings is the positive effect of the cross-modal annotation. When annotating one modality, the annotation can be propagated to another (from faces to voices for example), which results are an additional rudimentary set of annotations.

- The proposed active learning framework was additionally tested in the context of speaker identification model training. The experiments show that satisfactory performance can be achieved with just a few active learning steps and full annotation of the dataset is not necessary. After around 10 steps with the best selection strategy no significant gain in performance is observed.

6.1.2 Deep learning for multimedia

Here, the main contribution and conclusions are made involving the use of deep learning for video indexing (limited to Chapter 4):

- An experimental validation is given, which confirms the superiority of the learned deep learning based features over the more classical engineered ones. A relatively wide range of classical and learned features has been tested, including fusion.

- Fusion between learned and engineered features seems to improve the overall performance, which would indicate that the use of engineered features may still be advantageous.

- Additional performance improvements (in the case of video indexing) can be obtained through the use of multiple key frames or temporal and conceptual re-scoring methods. Also, it seems that the use of an SVM combined with the learned features may outperform the retraining of the last few layers of the DCNN.

6.1.3 Deep learning for speaker recognition

Finally, the main contributions connected to the use of convolutional neural networks for speaker identification are summarized here:

- A proposed exploratory CNN architecture that takes as input raw spectrograms and outputs the speaker identity.

- Despite displaying a weaker performance than the state-of-the-art, the fusion of the CNN output with the best performing method (in this case

TVS) leads to a significant boost in performance. The improvement is even more visible for short utterances (less than 2 seconds in duration), which are usually more challenging to improve upon.

6.2 Perspectives

Some of the potential research perspectives based on the work done throughout this thesis will be pointed out in this section. Apart from using more complex deep learning models, there are many interesting areas of research when dealing with multimodal data or trying to combine active and deep learning.

6.2.1 Active and deep learning

With the exception of the work presented in [YZS+15] (discussed in Chapter 2), there is not much work that tries to combine the active learning framework with deep learning. The long term goal in the deep learning community is to develop an unsupervised approach to learning complex models. However, this is still not the case and most approaches rely heavily on quality noise-free labeled data to achieve high performance. An intermediate step towards that long term goal could include the use of active learning. There are at least to potential challenges: the scale that is required and a reasonable computation time so it can be used in practice.

As for the scale, most deep learning algorithms requires thousands of annotated instances to obtain a model with competitive performance. Most active learning approaches assume that labeling is done one instance at a time, which may turn out to be too inefficient or costly. One of the possible solutions would be to limit the role of the annotator to verification, i.e. checking or cleaning the labels proposed by the learning algorithm, which is usually faster than identification. Another approach could involve label propagation or annotation of clusters rather than single instances. To reduce the issue of the cold start problem, a pretrained model (ideally trained on a related dataset) could be used, which would also require less annotated data to converge.

The second problem is the computation time. In the classical active learning approaches the learner is retrained every time new annotations are available. This may not be feasible when dealing with deep learning models. The approach in [YZS+15] suggests the use of a smaller model (e.g., SVM) which would interact with the human annotators and then use the more complex deep learning model once enough new data is acquired. A direct use of the DL model is still problematic. Potential solutions may involve the use of smaller and more specialized models (concentrated on learning just the subset of classes). Otherwise, fine-tuning (or retraining just the last few layers) may be a viable alternative.

6.2.2 Multimodal deep learning

Another potential research direction is a multimodal approach to deep learning. Given a dataset with several modalities such as videos, some instances can be

described by information coming from more than one source. This would allow the learning algorithm to have more than one input. This approach, for example, could potentially be used for person identification in videos where each person can be defined by both their face and their voice.

This approach can potentially be used to identify people that speak and can be seen at the same time or just based on one of the modalities. Additionally, it may provide a more robust representation that can deal with partial occlusion of the face or presence of noise in the speech track.

One of the potential difficulties is the synchronization between the two inputs. Any design decisions have to take into account both the nature of the input (e.g., pure audio signal, MFCC or spectrogram) and its relative size and frequency (a new input with every frame of the video, every second frame, etc.).

Appendices

Appendix A

Publications

1. M. Budnik, E.-L. Gutierrez-Gomez, B. Safadi, D. Pellerin, G. Quenot, *Learned features versus engineered features for multimedia indexing*, **accepted to** MTAP Journal.

2. M. Budnik, A. Khodabakhsh, L. Besacier, C. Demiroglu, *Deep complementary features for speaker identification in TV broadcast data*, Odyssey 2016.

3. M. Budnik, A. Khodabakhsh, L. Besacier, C. Demiroglu, *OCR-Aided Person Annotation and Label Propagation for Speaker Modeling in TV shows*, 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5570-5574).

4. M. Budnik, L. Besacier, J. Poignant, H. Bredin, C. Barras, M. Stefas, P. Bruneau, T. Tamisier, *Collaborative Annotation for Person Identification in TV Shows*, Interspeech 2015 (demo paper).

5. M. Budnik, B. Safadi, L. Besacier, G. Quénot, A. Khodabakhsh, C. Demiroglu, *LIG at MediaEval 2015 Multimodal Person Discovery in Broadcast TV Task*, MediaEval 2015 Workshop.

6. M. Budnik, E. L. Gutierrez Gomez, B. Safadi, G. Quénot, *Learned features versus engineered features for semantic video indexing*, CBMI 2015.

7. E. Demir, Z. Cataltepe, U. Ekmekci, M. Budnik, L. Besacier, *Unsupervised Active Learning for Video Annotation*, ICML Active Learning Workshop 2015.

8. M. Budnik, J. Poignant, L. Besacier, G. Quénot, *Active Selection with Label Propagation for Minimizing Human Effort in Speaker Annotation of TV Shows*, SLAM 2014.

9. B. Safadi, N. Derbas, A. Hamadi, M. Budnik, P. Mulhem, G. Quénot, *LIG at TRECVid 2014: Semantic Indexing*, Proceedings of TRECVID 2014.

10. P. Bruneau, M. Stefas, M. Budnik, J. Poignant, H. Bredin, T. Tamisier, B. Otjacques, *Collaborative Annotation of Multimedia Resources*, CDVE 2014.

11. M. Budnik, J. Poignant, L. Besacier, G. Quénot, *Automatic propagation of manual annotations for multimodal person identification in TV shows*, CBMI 2014.

Appendix B

Résumé en Français

B.1 Introduction

Au cours des dernières années, de plus en plus de documents multimédias sont disponibles grâce à la présence croissante et l'utilisation de caméras, smartphones et autres appareils d'enregistrement. L'utilisation d'Internet et de nombreux médias sociaux a permis d'accéder facilement à un nombre sans précédent de données diverses. Par ailleurs, en raison du partage et de la distribution en ligne, le volume de ce type de données augmente rapidement. Avec cette croissance rapide vient la nécessité de rendre les données plus utiles et accessibles aux utilisateurs potentiels. L'annotation et l'indexation permettent de rechercher ce contenu multimédia. Cependant, l'annotation manuelle d'une telle quantité de données est prohibitivement coûteuse. Pour résoudre ce problème, de nombreuses solutions potentielles ont été créées.

Une solution possible serait l'indexation multimédia automatique. Ceci est fait avec l'utilisation de diverses méthodes d'apprentissage automatique. Un tel système serait capable d'attribuer des étiquettes correspondant au contenu sémantique d'un document multimédia donné (qu'il s'agisse d'un objet pouvant être vu, d'une personne ou d'un locuteur d'une piste audio) sans intervention humaine, ce qui le rend identifiable et traçable à un utilisateur potentiel. Pour être en mesure de construire un tel système, qui aurait également un niveau satisfaisant de précision, et de le former avec succès, un grand nombre de données déjà annotées est nécessaire. Dans la plupart des cas, ces données doivent être annotées à la main par des annotateurs humains. Cependant, ce processus est toujours limité par les coûts en temps et en ressources. Comme toutes les données ne peuvent pas être étiquetées, il est nécessaire de hiérarchiser et de sélectionner certaines des instances de données par rapport à d'autres. Cela peut aider à éviter les redondances et conduire à un ensemble potentiellement plus représentatif d'instances marquées, ce qui peut à son tour augmenter la performance globale du modèle d'apprentissage automatique.

L'apprentissage actif représente un ensemble d'algorithmes conçus pour sélectionner des instances appropriées à partir d'un ensemble non marqué de données, compte tenu d'un certain critère. Il ya beaucoup de critères potentiels, qui dépendent de la tâche à accomplir, mais le but principal est de prédire l'utilité

d'une nouvelle instance pour un modèle donné. D'autres méthodes peuvent être utilisées pour augmenter le nombre total d'échantillons annotés grâce à la propagation d'étiquettes. En outre, l'apprentissage actif peut aider à éviter d'annoter l'instance redondante, c'est-à-dire celles qui ne portent pas de nouvelles informations utiles.

À mesure que la quantité de données annotées disponibles augmente, des modèles plus grands et plus complexes peuvent être formés. Cela conduit à la possibilité d'utiliser de nouveaux classificateurs tels que l'apprentissage profond. Les modèles les plus avancés de cette famille d'algorithmes nécessitent une grande quantité de données annotées pour atteindre la meilleure performance.

Dans cette thèse, certaines stratégies d'apprentissage actif sont proposées qui aident à la propagation de l'étiquette, ce qui peut augmenter la quantité de données étiquetées utiles et par conséquent la performance globale des modèles formés. En outre, plusieurs expériences différentes avec l'apprentissage profond ont été entrepris pour explorer l'utilité de l'ajustement des réseaux, la fusion et d'autres aspects lorsqu'il est appliqué au traitement du multimédia. Enfin, une application supplémentaire de cette classe d'algorithmes à la reconnaissance des locuteurs est explorée.

B.1.1 Apprentissage actif pour le multimédia

La plupart des applications utilisant des documents multimédias bénéficient ou même nécessitent une certaine quantité d'annotations manuelles. Compte tenu de la complexité de certains concepts (qu'il s'agisse d'une personne particulière ou d'un objet spécifique), l'intervention humaine est nécessaire. Avec une annotation appropriée, un grand nombre d'applications potentielles sont rendues possibles, comme des modèles de formation pour la reconnaissance ou la récupération. Cependant, le processus d'annotation (à condition qu'aucune source antérieure d'étiquettes ne soit donnée) peut être prohibitif, ce qui devient encore plus évident lorsqu'il s'agit de documents multimédias tels que des vidéos. Dans le cas de ce dernier, souvent en donnant une étiquette, certaines actions supplémentaires sont également nécessaires, par ex. fournir l'emplacement de l'objet dans une trame d'une vidéo (par une simple boîte ou une forme plus complexe) ou ajouter un timestamp pour indiquer quand un concept donné est visible. Ces étapes supplémentaires ne composent que le coût potentiel de chaque annotation.

Par conséquent, l'utilisation de méthodes actives liées à l'apprentissage peut être très bénéfique dans ce contexte. Lorsque le temps et les ressources sont limités, la possibilité de sélectionner les segments les plus informatifs des données pour l'annotation peut être inestimable. En outre, toute source probable d'étiquettes faibles telles que les noms superposés qui apparaissent sur l'écran ou sous-titres peut réduire considérablement le coût de l'annotation. En extrayant automatiquement les étiquettes probables et en les affectant à l'instance la plus probable, le processus d'annotation passe de l'identification à une tâche de vérification, généralement plus rapide et plus facile à exécuter.

B.1.2 Apprentissage profond

L'apprentissage profond a émergé récemment comme un ensemble très efficace d'algorithmes qui sont capables de résoudre des problèmes de reconnaissance complexes compte tenu de suffisamment de données d'entraînement. En particulier, le réseau de neurones convolutif est très bien adapté lorsqu'il s'agit de données composées d'images ou de vidéos et il est capable de trouver des concepts de haut niveau qui apparaissent dans les données.

Contrairement à l'approche traditionnelle où les caractéristiques décrivant une image sont "faites à la main", c'est-à-dire conçues explicitement pour extraire certaines caractéristiques d'une image telle que la couleur ou la texture, les réseaux neuronaux convolutifs sont capables d'apprendre les caractéristiques les plus appropriées pour un ensemble de données. C'est de loin l'élément le plus important qui contribue à la performance supérieure par rapport aux méthodes plus traditionnelles.

En raison d'une performance globale nettement meilleure dans de nombreuses tâches basées sur la vision, les méthodes de réseaux neuronaux convolutifs sont devenues les méthodes les plus utilisées pour résoudre des problèmes tels que l'indexation et la récupération d'images, la classification d'images et bien plus encore.

En raison de ce potentiel, une partie importante de cette thèse est consacrée à une évaluation supplémentaire de ses performances par rapport au paradigme plus traditionnel utilisé pour la classification des images. De plus, une exploration plus poussée des applications potentielles du réseau neuronal convolutif à d'autres domaines est étudiée.

B.2 Description du problématique

Dans cette section, une présentation plus détaillée du problème pour chaque aspect de cette thèse est définie. Cela inclut le défi central et certains des principaux problèmes.

B.2.1 Apprentissage actif pour le multimédia

Le principal défi consiste à créer un cadre d'apprentissage actif capable de traiter un ensemble de problèmes et de contraintes. En bref, ils peuvent être définis comme suit:

- Incorporer les données provenant de différentes sources, c'est-à-dire l'utilisation de données multimédia.
- Utiliser des étiquettes faibles, y compris un moyen de vérifier son exactitude.
- L'utilisation potentielle en pratique, qui impose des contraintes sur le temps d'exécution de toute solution proposée.

- Présence de séries de données fortement déséquilibrées, où l'utilisation d'un modèle formé peut être limitée au moins pour certains des concepts les moins fréquents.

B.2.2 L'apprentissage profond et ses applications

L'application de l'apprentissage profond dans le contexte de l'indexation vidéo pose plusieurs défis. Les principaux problèmes potentiels sont:

- Les données très déséquilibrées et bruyantes, ce qui rend difficile de former le modèle à partir de zéro.

- La présence de concepts multiples par cadre, ce qui nécessiterait une approche différente de celle utilisée dans la plupart des méthodes de classification d'image.

- Gestion du volume des données vidéo.

B.3 Contributions

Les contributions suivantes ont été faites tout au long des travaux sur cette thèse. Ils sont liés soit à l'application de l'apprentissage profond dans le multimédia ou à l'apprentissage actif et la propagation des étiquettes pour l'identification de la personne.

- Méthode d'annotation multimédia efficace utilisant l'apprentissage actif et la propagation d'étiquettes. Plusieurs stratégies d'échantillonnage différentes sont proposées. En outre, différentes sources d'information (visages, audio, noms écrits) sont utilisées. Des expériences ont montré l'utilité de cette approche pour la formation des modèles des locuteurs. Les détails sont présentés au chapitre 3.

- Une comparaison entre les caractéristiques classiques (non apprises) et les caractéristiques basées sur l'apprentissage profond a été faite au chapitre 4. L'utilisation des modèles d'apprentissage profond comme extracteurs de caractéristiques a été testée ainsi que le réglage fin dans le contexte de l'indexation et de la récupération d'images. La fusion entre les traits classiques et profonds a également été explorée.

- Une approche pour l'identification d'un locuteur basée sur un réseau de neurones convolutif formé sur des spectrogrammes est présentée au chapitre 5. Plusieurs techniques de fusion différentes ont également été proposées impliquant la sortie de la CNN et d'autres approches à la fine pointe de l'identification des locuteurs.

B.4 Apprentissage actif pour le multimédia

Le traitement de documents multimédias complexes tels que des vidéos peut être problématique, surtout si aucune annotation n'est disponible. Contrairement, par exemple, à l'annotation d'images, les vidéos d'étiquetage créent un ensemble de défis nouveaux et uniques. Tout d'abord, la division d'une vidéo en un groupe arbitraire de segments, qui sont en quelque sorte significatifs (par exemple, des scènes individuelles, des plans, des emplacements, etc.) peut être problématique. Deuxièmement, l'existence de multiples modalités rend la tâche dépendante de l'utilisation finale des données annotées. Si l'objectif est de créer un jeu de données pouvant être utilisé pour la formation de modèles d'enceintes, le processus d'annotation devrait se concentrer sur la segmentation vocale, qui peut différer de la structure des données pour d'autres tâches (par exemple, annotation de visage, d'objet ou de scène). L'ensemble des approches proposées dans ce chapitre tente d'aborder ces problèmes dans une certaine mesure.

Dans un scénario typique impliquant des annotateurs humains, la tâche est généralement binaire, c'est-à-dire lorsqu'on lui donne une image ou un échantillon sonore, on doit déterminer si un concept donné (chaise, voiture, montagne, etc.) est présent ou non. Une telle approche présente l'avantage d'être très efficace. D'autre part, lorsqu'il s'agit de l'identification de la personne, l'annotant doit fournir un nom spécifique si une personne donnée n'a pas été vue auparavant. Cela pourrait prendre beaucoup de temps et être sujet à des erreurs si un moyen d'écrire un nom n'est pas standardisé. Une partie importante de ces tâches peut être réduite à la vérification des noms (ou au choix du nom propre de la liste des candidats) si des étiquettes initiales automatiques et une procédure de propagation d'annotation sont utilisées. Dans un scénario de la vie réelle et lorsque le système proposé est utilisé, l'annotateur humain serait présenté avec une seule image ou une seule piste de parole qui représente un cluster correspondant.

Un cadre d'apprentissage actif (comme le montre la figue B.1) a été développé qui incorpore la propagation d'étiquettes pour accélérer l'acquisition d'instances étiquetées. Cela inclut l'évaluation de plusieurs modalités différentes (y compris la parole et les visages) ainsi que l'utilisation de noms écrasés comme étiquettes initiales. De plus, quatre stratégies de sélection différentes ont été évaluées. Selon l'évaluation expérimentale, la meilleure approche consiste à sélectionner les plus grands groupes avec la probabilité correspondant à leur taille et à utiliser les noms superposés comme annotation initiale lorsqu'ils sont disponibles.

L'un des résultats les plus intéressants est l'effet positif de l'annotation intermodale. En annotant une modalité, l'annotation peut être propagée à une autre (des visages aux voix par exemple), ce qui donne un ensemble rudimentaire supplémentaire d'annotations.

Le cadre d'apprentissage actif proposé a également été testé dans le contexte de la formation sur le modèle d'identification des locuteurs. Les expériences montrent que des performances satisfaisantes peuvent être obtenues avec seulement quelques étapes d'apprentissage actives et que l'annotation complète de l'ensemble de données n'est pas nécessaire. Après environ 10 étapes avec la meilleure stratégie de sélection, aucun gain significatif de performance n'est

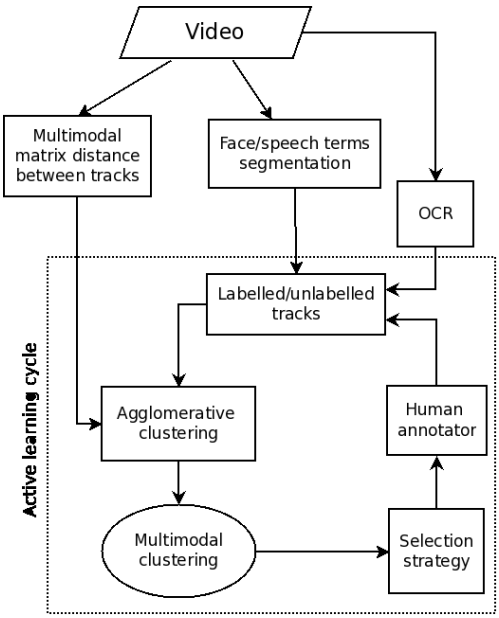


Figure B.1: Présentation du système.

observé.

B.5 L'apprentissage profond et ses applications

La comparaison a été faite entre l'utilisation de caractéristiques «traditionnelles» et des fonctions apprises pour l'indexation sémantique de contenu de documents vidéo. Une comparaison étendue a été faite de la performance des caractéristiques apprises avec ceux d'ingénierie traditionnels ainsi que des combinaisons d'entre eux. La comparaison a été effectuée dans le contexte de la tâche d'indexation sémantique TRECVid.

Les résultats confirment ceux obtenus pour les images fixes: les caractéristiques apprises à partir d'autres données d'apprentissage surpassent généralement les caractéristiques techniques pour la reconnaissance de concept. De plus, il semble que la formation directe des classificateurs KNN et MSVM utilisant ces fonctionnalités fait mieux que de recycler partiellement le DCNN pour l'adapter aux nouvelles données. Même si les caractéristiques apprises ont mieux fonctionné que les caractéristiques techniques, la fusion des deux a encore une meilleure performance, ce qui indique que les caractéristiques techniques sont toujours utiles, du moins dans ce cas.

De plus, les méthodes d'amélioration basées sur le re-scoring temporel, sur le retour conceptuel et sur l'utilisation de plusieurs trames par coup sont efficaces sur les deux types de caractéristiques et sur leur combinaison. Enfin, la combinaison des fonctions DCNN avec les classificateurs KNN et MSVM a été appliquée à la tâche de classification d'objets VOC 2012 où elle obtient actuellement la meilleure performance avec un MAP de 85,4 %.

Une autre application des réseaux neuronaux profonds a été étudiée, à savoir

l'utilisation de réseaux neuronaux convolutionnels (CNN) pour l'identification des locuteurs. Contrairement à l'approche classique de la plupart des méthodes d'identification des locuteurs, qui sont basées sur les caractéristiques MFCC (ou sur une version modifiée de celles-ci), ici le CNN est utilisé avec des spectrogrammes bruts. Ceci nous permet de traiter le signal vocal dans sa représentation 2D, c'est-à-dire juste comme une image. Cela pourrait créer une opportunité d'avoir un modèle CNN plus généralisé, qui pourrait être appliqué à un plus large éventail de tâches liées au traitement du signal. De plus, un jeu de données stimulant a été sélectionné, contenant à la fois du bruit et une quantité déséquilibrée de données de haut-parleurs. Cette approche CNN est comparée à des méthodes plus couramment utilisées. Malgré les performances CNN inférieures, l'utilisation de ces fonctionnalités complémentaires profondes dans la fusion améliore l'état de l'art.

Bibliography

[ABGM13] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. *arXiv preprint arXiv:1310.6343*, 2013.

[ACL90] Les E Atlas, David A Cohn, and Richard E Ladner. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, pages 566–573, 1990.

[AGP10] Marios Anthimopoulos, Basilis Gatos, and Ioannis Pratikakis. A two-stage scheme for text detection in video images. *Image and Vision Computing*, 28(9):1413–1426, 2010.

[AHESK10] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379, 2010.

[Ang88] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[ANR09] Shilpa Arora, Eric Nyberg, and Carolyn P Rosé. Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 18–26. Association for Computational Linguistics, 2009.

[AQ07] Stephane Ayache and Georges Quenot. Evaluation of active learning strategies for video indexing. *Signal Processing: Image Communication*, 22(7):692–704, 2007.

[AQ08a] Stéphane Ayache and Georges Quénot. Video corpus annotation using active learning. In *Advances in Information Retrieval*, pages 187–198. Springer, 2008.

[AQ08b] Stéphane Ayache and Georges Quénot. Video Corpus Annotation using Active Learning. In *European Conference on Information Retrieval (ECIR)*, pages 187–198, Glasgow, Scotland, mar 2008.

[AQG07] Stephane Ayache, Georges Quénot, and Jérôme Gensel. Image and video indexing using networks of operators. *EURASIP Journal on Image and Video Processing*, 2007(1):056928, 2007.

[Aro14] Hagai Aronowitz. Inter dataset variability compensation for speaker recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4002–4006. IEEE, 2014.

[AV] Namrata Anand and Prateek Verma. Convoluted feelings convolutional and recurrent nets for detecting emotion from audio data.

[BBE+04] Tamara L Berg, Alexander C Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller, and David A Forsyth. Names and faces in the news. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–848. IEEE, 2004.

[BBF+10] Martin Bauml, Keni Bernardin, Mika Fisher, Hazim Kemal Ekenel, and Rainer Stiefelhagen. Multi-pose face recognition for person retrieval in camera networks. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 441–447, 2010.

[BCDH10] A. Benoit, A. Caplier, B. Durette, and J. Herault. Using human visual system modeling for bio-inspired low level image processing. *Computer Vision and Image Understanding*, 114(7):758 – 773, 2010.

[BGP+15] Hervé Le Borgne, Philippe Gosselin, David Picard, Miriam Redi, Bernard Mérialdo, Boris Mansencal, Jenny Benois-Pineau, Stéphane Ayache, Abdelkader Hamadi, Bahjat Safadi, Nadia Derbas, Mateusz Budnik, Georges Quénot, Boyang Gao, Chao Zhu, Yuxing Tang, Emmanuel Dellandrea, Charles-Edmond Bichot, Liming Chen, Alexandre Benoit, Patrick Lambert, and Tiberius Strat. IRIM at TRECVid 2015: Semantic Indexing. In *Proceedings of TRECVID 2015*. NIST, USA, 2015.

[BL92] Eric B Baum and Kenneth Lang. Query learning can work poorly when a human oracle is used. In *International Joint Conference on Neural Networks*, volume 8, 1992.

[BLJ04] 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*, page 6. ACM, 2004.

[Bre96] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[BZMG06] Claude Barras, Xuan Zhu, Sylvain Meignier, and J Gauvain. Multistage speaker diarization of broadcast news. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1505–1512, 2006.

[CAL94] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[CBDF04] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[CG98] Scott Chen and Ponani Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, page 8. Virginia, USA, 1998.

[CH67] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[CSVZ14] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531, 2014.

[CUH15] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[DAHG+15] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[DAHY13] Li Deng, Ossama Abdel-Hamid, and Dong Yu. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6669–6673. IEEE, 2013.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[DE95] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157, 1995.

[DH08] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[DKD+11] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(4):788–798, 2011.

[DLH+13] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Mike Seltzer, Geoffrey Zweig, Xiaodong He, Julia Williams, et al. Recent advances in deep learning for speech research at microsoft. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8604–8608. IEEE, 2013.

[DMP+06] Leandro D'Anna, G Marrazzo, Gennaro Percannella, Carlo Sansone, and Mario Vento. A multi-stage approach for anchor shot detection. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 773–782. Springer, 2006.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[DW00] Perrine Delacourt and Christian J Wellekens. Distbic: A speaker-based segmentation for audio data indexing. *Speech communication*, 32(1):111–126, 2000.

[EEVG+15a] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

[EEVG+15b] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

[ESZ06] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy–automatic naming of characters in tv video. In *Proceedings of the 17th British Machine Vision Conference*, 2006.

[ESZ09] Mark Everingham, Josef Sivic, and Andrew Zisserman. Taking the bite out of automated naming of characters in tv video. *Image and Vision Computing*, 27(5):545–559, 2009.

[FS95] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[FS+96] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996.

[Fuk80] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.

[GCM+12a] Aude Giraudel, Matthieu Carré, Valérie Mapelli, Juliette Kahn, Olivier Galibert, and Ludovic Quintard. The repere corpus: a multimodal corpus for person recognition. In *LREC*, pages 1102–1107, 2012.

[GCM+12b] Aude Giraudel, Matthieu Carré, Valérie Mapelli, Juliette Kahn, Olivier Galibert, and Ludovic Quintard. The repere corpus: a multimodal corpus for person recognition. In *LREC*, pages 1102–1107, 2012.

[GCPF08] Philippe Henri Gosselin, Matthieu Cord, and Sylvie Philipp-Foliguet. Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. *Computer Vision*

and Image Understanding, 110(3):403 – 417, 2008. Similarity Matching in Computer Vision and Multimedia.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[GHT+14] Sriram Ganapathy, Kyu Han, Samuel Thomas, Mohamed Omar, Maarten Van Segbroeck, and Shrikanth S Narayanan. Robust language identification using convolutional neural network features. In *Proc. INTERSPEECH*, 2014.

[Gir15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[GMVS12] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Face recognition from caption-based supervision. *International Journal of Computer Vision*, 96(1):64–82, 2012.

[GREW11] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Proc. INTERSPEECH*, pages 249–252, 2011.

[GSK+15] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *arXiv preprint arXiv:1503.04069*, 2015.

[GWFM+13] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C Courville, and Yoshua Bengio. Maxout networks. *ICML (3)*, 28:1319–1327, 2013.

[Har54] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[HCC+14] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deepspeech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[Heb05] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[HJZ10] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *Advances in neural information processing systems*, pages 892–900, 2010.

[HMQ15] Abdelkader Hamadi, Philippe Mulhem, and Georges Quénot. Extended conceptual feedback for semantic multimedia indexing. *Multimedia Tools and Applications*, 74(4):1225–1248, 2015.

[HOT06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HZRS15a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[HZRS15b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[JDSP10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[JPD+12] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, September 2012.

[JSD+14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[JXYY13] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KWRB11] Martin Kostinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Learning to recognize faces from videos and weakly related information cues. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 23–28. IEEE, 2011.

[Lap05] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LCDH+90] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

[LG94] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[LGT16] Chen-Yu Lee, Patrick W Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *International Conference on Artificial Intelligence and Statistics*, 2016.

[LKF10] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[LLH15] Xinchao Li, Martha Larson, and Alan Hanjalic. Pairwise geometric matching for large-scale object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5153–5161, 2015.

[LMGDP+14] Ignacio Lopez-Moreno, Jorge Gonzalez-Dominguez, Oldrich Plchot, David Martinez, Joaquin Gonzalez-Rodriguez, and Pablo Moreno. Automatic language identification using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5337–5341. IEEE, 2014.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[LPLN09] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009.

[Mam98] Naoki Abe Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Machine Learning: Proceedings of the*

Fifteenth International Conference (ICML'98), page 1. Morgan Kaufmann Pub, 1998.

[MHN13] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.

[Mit82] Tom M Mitchell. Generalization as search. *Artificial intelligence*, 18(2):203–226, 1982.

[MLSF14] Mitchell McLaren, Yun Lei, Nicolas Scheffer, and Luciana Ferrer. Application of convolutional neural networks to speaker recognition in noisy conditions. In *Proc. INTERSPEECH*, 2014.

[MM96] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8):837–842, 1996.

[MM15] Dmytro Mishkin and Jiri Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

[MMK00] Ion Muslea, Steven Minton, and Craig A Knoblock. Selective sampling with redundant views. In *AAAI/IAAI*, pages 621–626, 2000.

[Moz89] Michael C Mozer. A focused back-propagation algorithm for temporal pattern recognition. *Complex systems*, 3(4):349–381, 1989.

[MSM16] Dmytro Mishkin, Nikolay Sergievskiy, and Jiri Matas. Systematic evaluation of cnn advances on the imagenet. *arXiv preprint arXiv:1606.02228*, 2016.

[MZN+14] Pavel Matejka, Le Zhang, Tim Ng, HS Mallidi, Ondrej Glembek, Jeff Ma, and Bing Zhang. Neural network bottleneck features for language identification. *Proc. IEEE Odyssey*, pages 299–304, 2014.

[NH10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[NS04] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.

[OAM+15] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, Georges Quénot, and Roeland Ordelman. Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2015*. NIST, USA, 2015.

[OFS+14] Paul Over, Jon Fiscus, Greg Sanders, David Joy, Martial Michel, George Awad, Alan Smeaton, Wessel Kraaij, and Georges Quénot. Trecvid 2014–an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, page 52, 2014.

[OM98] Genevieve B. Orr and Klaus-Robert Mueller, editors. *Neural Networks : Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*. Springer, 1998.

[OPH96] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[OT01] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[PBB+13] Johann Poignant, Hervé Bredin, Laurent Besacier, Georges Quénot, and Claude Barras. Towards a better integration of written names for unsupervised speakers identification in videos. In *First Workshop on Speech, Language and Audio in Multimedia, SLAM*, 2013.

[PBL+12] Johann Poignant, Hervé Bredin, Viet-Bac Le, Laurent Besacier, Claude Barras, and Georges Quénot. Unsupervised speaker identification using overlaid texts in tv broadcast. In *Interspeech 2012-Conference of the International Speech Communication Association*, page 4p, 2012.

[PBQT12] Johann Poignant, Laurent Besacier, Georges Quénot, and Franck Thollard. From text detection in videos to person identification. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 854–859. IEEE, 2012.

[PC+15] Dimitri Palaz, Ronan Collobert, et al. Analysis of cnn-based speech recognition system using raw speech as input. In *Proc. INTERSPEECH*, 2015.

[PD07] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[PG13] David Picard and Philippe-Henri Gosselin. Efficient image signatures and similarities using tensor products of local descriptors. *Computer Vision and Image Understanding*, 117(6):680–687, March 2013.

[PM04] Dan Pelleg and Andrew W Moore. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2004.

[PMT10] Phi The Pham, Marie-Francine Moens, and Tinne Tuytelaars. Naming persons in news video with label propagation. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 1528–1533. IEEE, 2010.

[Poi13] Johann Poignant. *Identification non-supervisée de personnes dans les flux télévisés*. PhD thesis, Université de Grenoble, 2013.

[PS01] Jason Pelecanos and Sridha Sridharan. Feature warping for robust speaker verification. *IEEE Odyssey: The Speaker and Language Recognition Workshop*, pages 213–218, 2001.

[RASC14a] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[RASC14b] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 512–519, June 2014.

[RDS+15a] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.

[RDS+15b] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[RHW88] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[RMJ06] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7(Aug):1655–1686, 2006.

[RMN09] Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, pages 873–880. ACM, 2009.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[RQD00] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000.

[RRD15a] Fred Richardson, Douglas Reynolds, and Najim Dehak. Deep neural network approaches to speaker and language recognition. *IEEE SIGNAL PROCESSING LETTERS*, 22(10):1671, 2015.

[RRD15b] Fred Richardson, Douglas Reynolds, and Najim Dehak. A unified deep neural network for speaker and language recognition. *arXiv preprint arXiv:1504.00923*, 2015.

[SBL13] S.T. Strat, A. Benoit, and P. Lambert. Retina enhanced sift descriptors for video indexing. In *Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on*, pages 201–206, June 2013.

[SBL+14a] Sabin Tiberius Strat, Alexandre Benoit, Patrick Lambert, Hervé Bredin, and Georges Quénot. Hierarchical late fusion for concept detection in videos. In Bogdan Ionescu, Jenny Benois-Pineau, Tomas Piatrik, and Georges Quénot, editors, *Fusion in Computer Vision*, Advances in Computer Vision and Pattern Recognition, pages 53–77. Springer International Publishing, 2014.

[SBL14b] S.T. Strat, A. Benoit, and P. Lambert. Retina enhanced bag of words descriptors for video classification. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pages 1307–1311, Sept 2014.

[SCR08] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[SDBR14] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[SDH+14] Bahjat Safadi, Nadia Derbas, Abdelkader Hamadi, Mateusz Budnik, Philippe Mulhem, and Georges Quénot. LIG at TRECVid 2015: Semantic Indexing. In *Proceedings of TRECVID*, Orlando, United States, November 2014.

[SDQ15] Bahjat Safadi, Nadia Derbas, and Georges Quénot. Descriptor optimization for multimedia indexing and retrieval. *Multimedia Tools and Applications*, 74(4):1267–1290, 2015.

[SDW01] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *Advances in Intelligent Data Analysis*, pages 309–318. Springer, 2001.

[Set09] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[Set11] Burr Settles. From theories to queries: Active learning in practice. *Active Learning and Experimental Design W*, pages 1–18, 2011.

[Set12] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[SEZ+13] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[Sha01] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[SHK+14] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[SJ12] Yu Su and Frédéric Jurie. Improving image classification using semantic attributes. *International Journal of Computer Vision*, 100(1):59–77, 2012.

[SL12] A. Shabou and H. LeBorgne. Locality-constrained and spatially regularized coding for scene categorization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3618–3625, June 2012.

[SLJ+14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[SMB00] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, 2000.

[SNN03] J.R. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 2, pages II–445–8 vol.2, July 2003.

[SOK06] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[SOS92] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[SPMV13] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[SQ10] Bahjat Safadi and Georges Quénot. Evaluations of multi-learner approaches for concept indexing in video documents. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pages 88–91, Paris, France, France, 2010. Le centre de hautes études internationales d'informatique documentaire.

[SQ11] Bahjat Safadi and Georges Quénot. Re-ranking by Local Rescoring for Video Indexing and Retrieval. In Ian Ruthven Craig Macdonald, Iadh Ounis, editor, *CIKM 2011 - International Conference on Information and Knowledge Management*, pages 2081–2084, Glasgow, United Kingdom, October 2011. ACM. Poster session: information retrieval.

[SQ12a] Bahjat Safadi and Georges Quénot. Active learning with multiple classifiers for multimedia indexing. *Multimedia Tools and Applications*, 60(2):403–417, 2012.

[SQ12b] Bahjat Safadi and Georges Quénot. Active learning with multiple classifiers for multimedia indexing. *Multimedia Tools and Applications*, 60(2):403–417, 2012.

[SQ15] B. Safadi and G. Quenot. A factorized model for multiple svm and multi-label classification for large scale multimedia indexing. In *Content-Based Multimedia Indexing (CBMI), 2015 13th International Workshop on*, pages 1–6, June 2015.

[SS02] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[SSH13] Seyed Omid Sadjadi, Malcolm Slaney, and Larry Heck. Msr identity toolbox v1. 0: A matlab toolbox for speaker recognition research. *Speech and Language Processing Technical Committee Newsletter*, 2013.

[SWG+05] C. G. M. Snoek, M. Worring, J. M. Geusebroek, D. C. Koelma, and F. J. Seinstra. On the surplus value of semantic video analysis beyond the key frame. In *IEEE International Conference on Multimedia & Expo*, 2005.

[SWS05] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM, 2005.

[SZ03a] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[SZ03b] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1470–, Washington, DC, USA, 2003. IEEE Computer Society.

[SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[TC01] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118. ACM, 2001.

[TK02] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[TYRW14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lars Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014.

[UW15] Lior Uzan and Lior Wolf. I know that voice: Identifying the voice actor behind the voice. In *Biometrics (ICB), 2015 International Conference on*, pages 46–51. IEEE, 2015.

[VD02] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

[VdSGS08] Koen EA Van de Sande, Theo Gevers, and Cees GM Snoek. A comparison of color features for visual concept classification. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 141–150. ACM, 2008.

[vdSGS10a] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[VDSGS10b] Koen Van De Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1582–1596, 2010.

[VG14] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1-2):97–114, 2014.

[VJS05] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[VLBM10] Viet-Vu Vu, Nicolas Labroche, and Bernadette Bouchon-Meunier. Active learning for semi-supervised k-means clustering. In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 12–15. IEEE, 2010.

[VTBE15] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[WA95] Dietrich Wettschereck and David W Aha. Weighting features. In *International Conference on Case-Based Reasoning*, pages 347–358. Springer, 1995.

[WJC02] Christian Wolf, Jean-Michel Jolion, and Francoise Chassaing. Text localization, enhancement and binarization in multimedia documents. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, pages 1037–1040, 2002.

[Wol92] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[WXH+14] Yunchao Wei, Wei Xia, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. Cnn: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*, 2014.

[XWCL15] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[YBT06] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning*, pages 1081–1088. ACM, 2006.

[YCBL14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

[YZS+15] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[ZBC11] Chao Zhu, Charles-Edmond Bichot, and Liming Chen. Color orthogonal local binary patterns combination for image region description. *Rapport technique RR-LIRIS-2011-012, LIRIS UMR*, 5205:15, 2011.

[ZBC13] Chao Zhu, Charles-Edmond Bichot, and Liming Chen. Image region description using orthogonal combination of local binary patterns enhanced with color information. *Pattern Recognition*, 46(7):1949–1963, 2013.

[ZCB+11] Lijun Zhang, Chun Chen, Jiajun Bu, Deng Cai, Xiaofei He, and Thomas S Huang. Active learning based on locally linear reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(10):2026–2038, 2011.

[ZF13] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[ZF14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.