Unsupervised discovery of Interpretable Visual Concepts

Caroline Mazini Rodrigues, Nicolas Boutry, Laurent Najman

Unsupervised discovery of Interpretable Visual Concepts

Caroline Mazini Rodrigues[a,b], Nicolas Boutry[a], Laurent Najman[b]

[a]*Laboratoire de Recherche de l'EPITA – LRE, 14-16, Rue Voltaire, Le Kremlin-Bicêtre, 94270, France*
[b]*Laboratoire d'Informatique Gaspard Monge – LIGM, 5 Boulevard Descartes, Marne-la-Vallée, 77454, France*

Abstract

Providing interpretability of deep-learning models to non-experts, while fundamental for a responsible real-world usage, is challenging. Attribution maps from xAI techniques, such as Integrated Gradients, are a typical example of a visualization technique containing a high level of information, but with difficult interpretation. In this paper, we propose two methods, *Maximum Activation Groups Extraction* (MAGE) and *Multiscale Interpretable Visualization* (Ms-IV), to explain the model's decision, enhancing global interpretability. MAGE finds, for a given CNN, combinations of features which, globally, form a *semantic* meaning, that we call *concepts*. We group these similar feature patterns by clustering in "concepts", that we visualize through Ms-IV. This last method is inspired by Occlusion and Sensitivity analysis (incorporating causality), and uses a novel metric, called *Class-aware Order Correlation* (CaOC), to globally evaluate the most important image regions according to the model's decision space. We compare our approach to xAI methods such as LIME and Integrated Gradients. Experimental results evince the Ms-IV higher localization and faithfulness values. Finally, qualitative evaluation of combined MAGE and Ms-IV demonstrates humans' ability to agree, based

on the visualization, on the decision of clusters' concepts; and, to detect, among a given set of networks, the existence of bias.
Keywords:
explainable artificial intelligence, interpretability, convolutional neural networks, global artificial concepts

1. Introduction

The use of machine learning (ML) in real-world applications increased the need for explaining decisions to non-computer experts. However, providing model explanations of isolated features is challenging. Consider the explanation in Fig. 1(b) which is pixel-level importance annotated: we do not directly understand the model's knowledge. It is not easily *interpretable*.

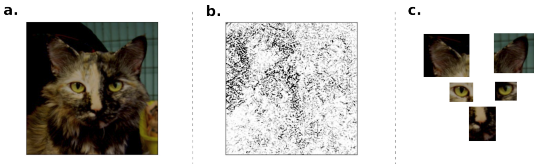


Figure 1: Pixel-level importance is more difficult to interpret than components one. From left to right: an image, its Integrated gradients [41] and an easier-to-interpret visualization.

Interpretability, compared to *explainability*, is more subjective as it involves semantics and the idea of how Humans understand signals [2; 15]: the process of interpretation is a *translation of knowledge* that depends not only on the information semantics, but also on how it is transmitted and received [35].

Methods like LIME [30] and KernelSHAP [26] propose visualizations based on interpretable components rather than isolated pixels. These components facilitate the human interpretation of how a model understands a

sample. However, instead of understanding the **complete** model's knowledge, they explain the behaviour of the convolutional neural network (CNN) to an **individual** image.

Works such as TCAVs [22] and Explanatory graphs [49] aim to translate the model's knowledge and how it behaves given input changes, into interpretable concepts. Apart from increased interpretability, the required supervision can affect how effectively a model is explained. In the case of TCAVs, we need to know the concepts that we are testing the model against. In the case of Explanatory graphs, we need to train a time-expensive model to approximate a graph of activation patterns for a set of images.

These supervisions' constraints can impact on how well a model can be explained, i.e., if the explanation provided is complete enough to represent the model reasoning. To solve this problem, ACE [13] was proposed to use image segments, represented by internal activations, clustered as *concepts*. In this way, TCAV no longer requires a user *concept* supervision. However, as an example-based technique, it depends on how and what images are segmented, i.e., if we do not use images containing all concepts, some of the concepts could be left out.

We use a similar idea to cluster concepts, but instead of clustering segments' activations, we cluster *internal units' activation patterns*. Doing this, we are able to provide a more global and complete set of concepts.

Our methodology tackles three aspects of CNN's explainability problem: i) we represent the models' knowledge as **completely** and **globally** as possible without supervision; ii) we obtain explanations based on how humans understand **concepts** (patterns groups with similar *semantics*), and; iii) we

provide **interpretability** to the explanations, enabling the use of intelligent systems by non-experts.

Our main contributions are:

- Maximum Activation Groups Extraction (MAGE), that constructs a novel feature-maps representation based on activation patterns **localization** in **multiple** images, instead of, the normal activation vectors from individual images;
- The Class-aware Order Correlation (CAOC) metric, to determine the impact of **occlusions**, not only in a single image activation, but also how, according to the model, this image relates to the others (**dataset relation)**, and;
- A Multiscale Interpretable Visualization (Ms-IV), using $\mathcal{CAOC}$ to have an occlusion-based visualization accounting for **dataset relations**; and presenting **hierarchical selection** of important image regions (from the complete image to the smallest defined patches' size) to focus human sight on gradually highlighted image parts.

Section 2 is a literature review of xAI methods, Section 3 presents the intuition of our method, Section 4 shows qualitative and human-based experiments and Section 5 concludes the paper.

2. Literature review and Motivation

xAI methods can be broadly categorised as *intrinsic*, *model-specific*, or *post-hoc* and *model-agnostic*. Examples of *intrinsic* methods are decision

trees [29], some attention networks [14] and joint training with text explanation [28]. They are called *intrinsic* because they do not need an extra mechanism to provide some level of explanation. For these methods, we have the explanations directly from the analysed learning model.

The *specific-methods* can also cover *intrinsic* models. However, they refer to explanations specifically applied to some determined architectures. For example, the Deconvolution [47], CAM [50] and Grad-CAM [33] techniques are firstly designed to explain Convolutional Neural Networks. Nevertheless, they are not *intrinsic* but *post-hoc* models, as they are applied to a pre-trained model.

According to recent xAI surveys, the mentioned classical methods, such as LIME [30], SHAP [26], DeconvNet [47], CAM [50], Grad-CAM [33], Guided Grad-CAM [33], DeepLIFT [38], Integrated Gradients [41], Guided-Backpropagation [40], Saliency maps, TCAV [22] still are the most referenced ones [12; 3; 32]. Their application is disseminated through different domains and are presented in recent researches.

The medical domain is one of the biggest applications of xAI techniques. Some literature reviews in this domain mention the use of TCAV as a concept analysis technique [9; 31; 37]; CAM, Grad-CAM, LRP [5], SHAP and LIME as visual-based model explanations [31; 10]; ACE [13], Network dissection [6] and some supplementary techniques for visualization such as t-SNE and UMAP [9]. Tim Hulsen [21] mentions that most of the papers in this area are based on visual explanation, for different purposes, such as: lungs ultrasound [8] and breast cancer X-rays [36] using CAM; ulcerative colitis colonoscopy using Grad-CAM [42]; COVID-19 detection in chest CTs [25]

using Grad-CAM; lung X-ray [17] using Grad-CAM and LIME, and; chest X-ray images [1] using LIME, Integrated Gradients and SHAP.

In areas such as network security, models as LIME, SHAP and induced decision trees are used to explain the detection of malicious domains [4]. There are also applications in forecasting within the manufacturing domain [11] using methods such as recursive feature elimination (RFE) [16], and SHAP.

However, although their frequent use, we do not believe that all currently used xAI techniques are sufficiently interpretable for non-experts analysis, and we do not believe that they fully explain the reasoning of models.

2.1. Model-agnostic methods and interpretability

The *model-agnostic* methods are generally *post-hoc* methods, and can explain multiple types of architectures. Some examples of *post-hoc* and *model-agnostic* methods are Layer-wise Relevance Propagation [5] and Integrated Gradients [41], but also LIME [30] and its numerous derivatives [30; 46; 51; 45; 24; 34], TreeView [44], and Explanatory graphs [48]. We describe some methods with a higher level of interpretability and their differences.

LIME [30]: is a *model-agnostic* method which introduces the idea of explaining by using interpretable components. The method decomposes each data sample into human understandable parts. If a data sample is an image, these decomposed parts can be, for example, superpixels or patches, not necessarily expressed as it is inputted in the model. After we have these parts (or components), LIME measures their importance to a decision. This approach is more human-friendly than showing each individual feature importance, especially in high-dimensional data. However, despite presenting high interpretability, these explanations are generally local, *i.e.*, they are

sample-based or rely on local explanations to explain the model behaviour.

Explanatory Graphs: proposed by Zhang et al. [48], represent a CNN knowledge hierarchy through convolutional layers. Each node in the graph represents candidate patterns of the object's parts, summarising the knowledge from feature maps. Edges connect nodes from adjacent layers. The method proposes to disentangle object parts from a single filter without ground-truth part annotations. It mines highly activated image patterns from the last convolutional layer (high-level semantics) to the first (simpler structures). This process relies on the complete dataset to optimise the graph of hierarchically connected patterns that best fit network feature maps.

Testing with Concept Activation Vectors (TCAV): proposed by Kim et al. [22], aims to, use a set of low-level features to provide human-friendly, interpretable concepts. In more detail, CAVs' method is part of TCAV, which analyses how sensitive a model's prediction is to a user's predefined concept. The idea is to learn a linear classifier to separate, based on the model internal activations and a given class, the response to the concept's given examples and random given examples. TCAV ultimately shows the images most similar to a concept.

Besides the improvement in interpretability, Explanatory Graphs and TCAVs require a level of supervision, to generate the knowledge graphs or to indicate the concepts. We also want a more global network's explanation but in an unsupervised way.

2.2. Our motivation and similar literature methods

We propose to extract interpretable visual concepts from a model. We already mentioned some similar ideas to our proposal: TCAVs and Ex-

planatory Graphs (described in Section 2.1). However, a paper proposed by Tan el al. [43], has a similar idea: to identify semantic concepts within networks. The authors suggest inducing, during training, neighboring neurons (or feature maps) to exhibit similar activations. The objective is to have an easier interpretation of the activation maps visualizations, showing similar regions of activations for similar semantic concepts. The method, called Locality Guided Neural Network (LGNN), conditions during training, the filters' topology to facilitate manual inspection. However, the difference from our approach is that this method focuses on changing the learning algorithm, i.e., it should be used during the training process to change the model. That is not our objective, as we expect to explain general, already trained, CNNs.

Another work, proposed by Li et al. [23], presents a network, PatternNet, to mine visual patterns that are discriminative and representative. They consider these patterns should be popular (*representative*), i.e., activated in a considerable number of images from the analyzed class; and unique (*discriminative*) for this class (not appearing in the rest of the classes). However, this is also a supervised task. The authors train PatternNet to find these patterns. Therefore, it is a dataset explanation technique and not a model explanation technique, as we expect to propose.

The method ACE [13] proposes a way to mine concepts, without direct supervision. It uses image segments with different scales to cluster similar patterns, represented by the segments' network activations. TCAV is then used to measure the importance of each cluster. We have a similar proposal, however, we aim to represent units as their activation patterns in the complete dataset. In this way, we include a global view of network behaviour

whilst decomposing the network into units with similar concepts.

We believe our methods can fill the gap by providing an unsupervised means of mining semantic patterns inside a pre-trained model, through decomposing the model's *global* knowledge into interpretable concepts.

3. Proposed methods

Our intuition is: if we can decompose networks' knowledge into different *concepts* (used for the network decision), we can *translate* them into human-understandable visualizations (patterns). We describe these two tasks.

Concepts' decomposition: We define *concepts* as combinations of features which form a *semantic meaning*. This generally implies *spatial proximity*. For animals' images (cats & dogs), a concept can be nearby features that compose a muzzle, ears or eyes at a given location. Together, these concepts induce the animal's presence perception in the image at this same location.

For digital images, these features are pixels. For the human visual system, the process of grouping pixels into concepts seems automatic, but it is not the same for machines. In the case of artificial neural networks, the patterns that indicate these concepts are learned by internal units during training.

As previously investigated by other works, we observe different learned patterns inside a CNN by looking into convolutional layers' activations. These patterns can be determined by structures (in the input) that most activate each feature map. It is quite similar to mapping human brain activity. We give a stimulus and look at what lights up in the brain.

The problem is that CNNs have high-dimensional feature maps, and reasoning based on them is humanly infeasible. Our solution is to group similar

responses of feature maps' dimensions to provide easier analysis.

Concepts' discovery through visualization: We use a hierarchical visualization strategy to enrich *human understandability*. From a higher to smaller size of images' substructures (patches), we want to gradually increase attention in the most important parts, from bigger to smaller regions. The idea is similar to a face verification task. The first step is to detect faces in images, then, to compare the faces. The complete face is important, however, for verification, facial characteristics such as eyes, nose, and mouth will be more significant. These characteristics are hierarchically linked to the face (inside the face) and during this task, we assign a gradual level of focus, from the face to its specific regions contained within. Similar to this, we expect to facilitate a gradual human attention process, to understand the importance of the main structure and, subsequently, the specific linked characteristics.

The visualization is intended to represent the concepts we previously decomposed. We want to visualize what parts of the original image impact the most for each concept. Using an example: we want to know which one, a dog's muzzle or a dog's eyes, *causes* the biggest impact in a concept A. In this way, **we can discover which concept is A**.

We try to capture this *causality* through occlusions. We evaluate the impact of each image region by occluding it and verifying what changed according to a concept A. If the response of the concept A changes a lot, the occluded part is important, and a candidate to explain what is the concept A. Different image parts will have different levels of importance. We expect the most important parts to represent the concept.

However, these occlusions can only be made in each image individually.

If we want to account for the *globalism* of a concept (same concept for the majority of images), we need a strategy to include global awareness in the evaluation of the concept after-occlusion impact.

Our solution to be globally aware is to evaluate the occlusion impact on the relation of images containing the concept A. Let us consider a pre-defined set of images. After one image occlusion, we measure how many relationships we have changed in this group. This way, if an image has the concept A, the image occlusion of this concept will change its relation to the others (it will have less A than the others). On the other hand, if there is no concept A in this image, even if occlusions change the image's prediction, it should not change the image's relation to the others.

3.1. Decomposition into concepts: Maximum Activation Groups Extraction

The network's knowledge decomposition is made in five steps (Figure 2).

3.1.1. Decomposition of input features into patches

We decompose the image into non-overlapping patches to evaluate the stimulus of images' subregions to the feature maps. Humans are able to decompose an image into *semantic pieces* to understand it in its entirety.

To be able to interpret what CNN-based classifiers "visualize", we propose then to decompose their feature maps of highest abstraction level into similar dimensions. Note that we prefer to use the last convolutional layer because it is able to represent high-level semantic concepts, but we do not use fully-connected layer activations because they lose the "visual awareness". To proceed, we start from a CNN like a VGG [39] or a ResNet [18], that we

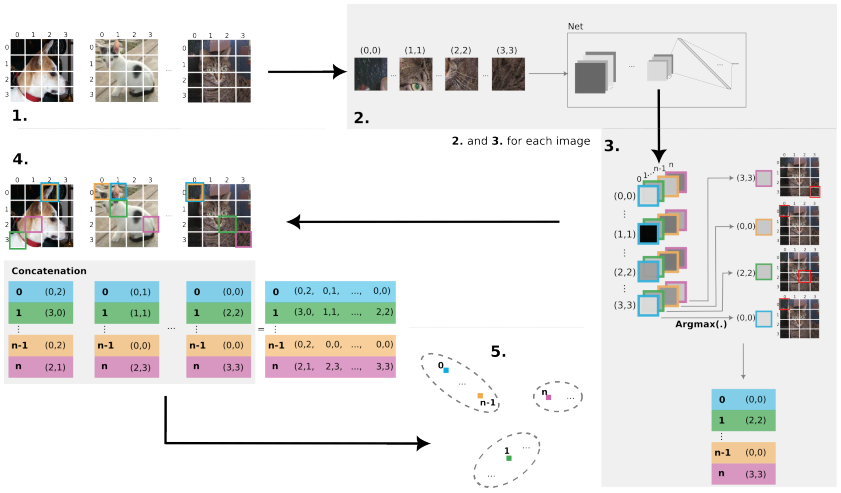


Figure 2: Steps to obtain the *Maximum Activation Groups* (MAGs). We divide dataset images into patches (we perform experiments with different patch sizes to obtain better separation in **5.**) (**1.**). We obtain feature maps (from the last convolutional layer) for each patch (**2.**). We find the corresponding patch with the highest feature map norm by dimension (**3.**). We concatenate the patches' positions of the highest norms for a set of images to represent each feature map's dimension (**4.**). We cluster the dimensions' representations to obtain the MAGs (**5.**). More detail in Appendix A.

model using the formula

$$\Xi = \Xi^{classif} \circ \Xi^{enc} \tag{1}$$

(with $\Xi^{classif}$ the classifier following the encoder denoted by Ξ^{enc}). The encoder includes the network's layers up to the last convolutional layer. Then, for a given image $\mathcal{I}$, we decompose the image into patches.

According to us, this decomposition into patches is one of the keys to be able to understand a little more how the network "reasons"; it is our way to tackle a little more the *black-box effect* of CNNs well-known in deep learning. To be more formal, let us introduce some notations. The dataset

$\mathcal{DS} = (\mathcal{I}_i, GT_i)_{i\in[1,NbIm]}$ used to train Ξ is made up of $NbIm$ images $\mathcal{I}_i$ and their class GT_i (the class number). For a given $i \in [1, NbIm]$, we denote $\mathcal{I}_i$ as the i^{th} image of $\mathcal{DS}$. We denote by $NbIm(c)$ as the number of images of class $c \in [1, NbClasses]$. For a given class c and for a given $i_{local} \in [1, NbIm(c)]$, we will denote $\mathcal{I}^c_{i_{local}}$ as the i^{th}_{local} image of class $c \in [1, NbClasses]$ of $\mathcal{DS}$.

We can then introduce our formalism. To decompose the domain $\mathcal{D}$ of a given image $\mathcal{I}$ into patches of dimension $s_p \times s_p$ (see Figure 2 **(1)**), we proceed this way: $\mathcal{D} = \bigcup_{\ell_x\in[1,\ell_x^{\max}],\ell_y\in[1,\ell_y^{\max}]} \mathcal{P}(\ell_x, \ell_y, s_p)$, with $\ell_x, \ell_y, \ell_x^{\max}, \ell_y^{\max} \in \mathbb{N}$, $\ell_x^{\max}, \ell_y^{\max}$ the number of patches horizontally and vertically (respectively), and (ℓ_x, ℓ_y) the relative coordinates of the patch $\mathcal{P}(\ell_x, \ell_y, s_p)$ described by

$$\mathcal{P}(\ell_x, \ell_y, s_p) = [1 + (\ell_x - 1) \cdot s_p,\ \ell_x \cdot s_p] \times [1 + (\ell_y - 1) \cdot s_p,\ \ell_y \cdot s_p] \quad (2)$$

(we obtain then a partition of $\mathcal{D}$).

3.1.2. Calculus of feature maps' activation per patch

We otain activations of feature maps by giving each individual patch to the network — the response to the stimuli (see Figure 2 **(2)**).We use the model described by Equation 1, in which Ξ^{enc} includes the network's layers up to the last convolutional layer. Since for the image i and for the n_f^{th} feature, we have the 2D mapping $\Xi^{enc}(.,.,n_f) : (x, y) \to \mathbb{R}$, and that we will restrict the input image to the patch $\mathcal{P}(\ell_x, \ell_y, s_p)$, we propose to introduce the term $\mathcal{F}_{i,n_f,(\ell_x,\ell_y),s_p} : (x, y) \to \mathbb{R}$, which maps an image patch $\mathcal{P}(\ell_x, \ell_y, s_p)$ into the n_f^{th} feature map after a forward pass through Ξ^{enc}, representing the 2D feature map activations for the mentioned patch. Now that we have introduced the formalism to represent feature patches, let us show how we decompose the "knowledge" of the encoder into *concepts*.

3.1.3. Identifying important patches for feature map's dimensions

We want to group feature map dimensions according to their activation patterns. Therefore, we characterise these dimensions by their patterns. This characterisation is similar to a brain experiment: if part A of the brain is more activated by emotions than by a task like reading, part A probably knows the concept of emotions. Instead of using emotions and reading, we give the patches to the network. Therefore, we identify the patches that activate a feature map's dimension the most to represent the mentioned dimension. The identification is done by locating the selected patch; in this paper, *it is based on its position in the original image.*

Let us choose some image $\mathcal{I}_i$ in $\mathcal{DS}$. We consider as the *reference patch* in the n_f^{th} feature map corresponding to $\mathcal{I}_i$ the one which maximises the 1-norm. It is then identified by its parameters:

$$(\ell_x^*(i, n_f), \ell_y^*(i, n_f)) = \arg\max_{(\ell_x, \ell_y)} \left\{ ||\mathcal{F}_{i,n_f,(\ell_x,\ell_y),s_p}||_1 \right\}. \tag{3}$$

Intuitively, this position represents the patch where the CNN reacted the most (see Figure 2 **(3)**).

3.1.4. Dimension characterization by the dataset

We have limited (local) information if we use only one image to characterise dimensions. Therefore, we incorporate more images in the process. Instead of having the feature map's dimensions characterised by only one image's patches, we repeat the process with more images to obtain multiple characterisations. We can then define as "concept" the set of features activated at (almost) the same location for each image of $\mathcal{DS}$ (see Figure 2 **(4)**).

In other words, by defining the *representative* of the n_f^{th} feature:

$$rep(n_f) = \left[\ell_x^*(1, n_f), \ell_y^*(1, n_f), \ell_x^*(2, n_f), \ldots, \ell_y^*(NbIm, n_f)\right]^T, \qquad (4)$$

we obtain a vector in a space (of dimension $2NbIm$) which satisfies the property that when two features n_f^1 and n_f^2 are physically near to each other in the images of $\mathcal{DS}$, their representation will be near to each other, and conversely.

3.1.5. Decomposition of feature space into concepts

We use the ensemble of characterisations of a feature map's dimension to create a feature vector representing it. If two dimensions have similar feature vectors, they activate in similar patches for most of the images. We consider them the same concept. Therefore, we cluster the feature vectors for all feature map's dimensions to obtain the groups of concepts. This allows us to find the concepts using any clustering algorithm (in this paper, we used K-means) to obtain the *maximal activation groups* (MAG):

$$\{MAG(k)\}_{k\in[1,K]} = Clustering(K, \{rep(n_f)\}_{n_f}). \qquad (5)$$

Each term $MAG(k)$ is what we formally define as a $\mathcal{DS}$*-relative concept*. They are relative to K and to the clustering algorithm used. Thanks to them, we can understand the *global* behaviour of the CNN.

3.2. Global causality visualization: Multiscale-Interpretable Visualization

After we have the specific *concepts*, we follow the inverse path: we look at what a *concept* represents in each image. We aim to visualize image regions with more impact for the concept $MAG(k)$ in the model's decision, relying on **human understandability**, **causality**, and **globalism**.

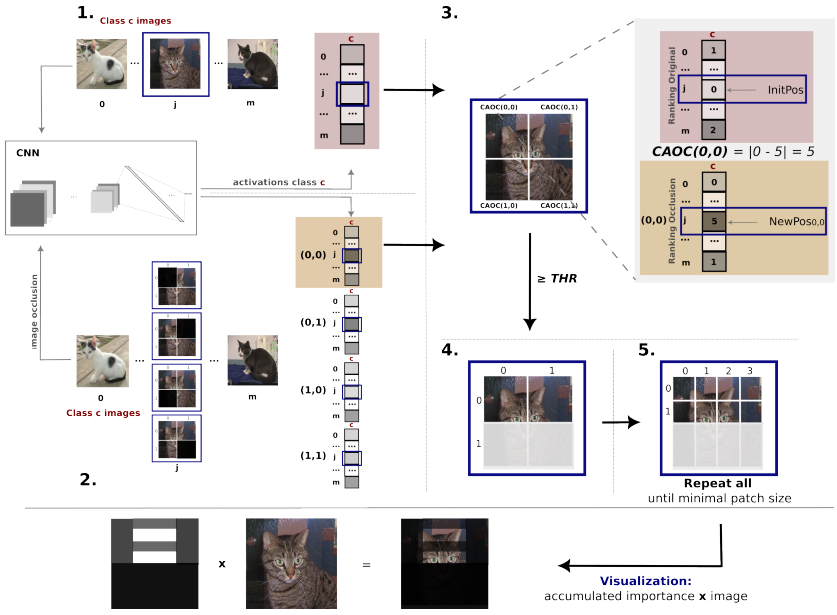


Figure 3: *Multiscale-Interpretable Visualization* (Ms-IV). We obtain the final responses of the network (outputs before Softmax) for a set of images (**1.**). We occlude one patch (out of 4) from the image we want to visualize, and we calculate the new model's response (**2.**). We apply **Argmax** to the image's model responses to obtain orders according to a class **c**. We want two vectors, one ordering all the images, including the original image (to visualize), and; another ordering all images, except the original one (to visualize), but including its occluded version. We obtain the differences in the position of the original and the occluded (**3.**) to account for the modification in the output space. It is considered the importance of that patch (*Class-aware Order Correlation* ($\mathcal{CAOC}$)). We do the same to obtain $\mathcal{CAOC}$ for all patches and filter (based on a threshold) the patches that will continue in the next hierarchical visualization level (**4.**). We reduce the size of patches and repeat the process. We accumulate the importances for all hierarchical levels to, in the end, multiply by the original image and obtain the visualization (**5.**). More detail in Appendix B.

For the sake of simplicity, let us introduce a new term: we define an *occlusion* of the image I of domain $\mathcal{D}$ on a patch $\mathcal{P} \subseteq \mathcal{D}$ as $\blacksquare_{\mathcal{P}}(\mathcal{I}) : \mathcal{D} \to \mathbb{R}$

such that for any $(x, y) \in \mathcal{D}$, $\blacksquare_{\mathcal{P}}(\mathcal{I})(x, y)$ is equal to $\mathcal{I}(x, y)$ when $(x, y) \notin \mathcal{P}$, and 0 otherwise.

We set a visualization threshold ratio $\delta \in]0, 1]$, a minimum patch size $s_p^{min} \in \mathbb{N}^*$ (representing the minimum patch size of a concept), a class c, the image $\mathcal{I}_j^c$ for visualization, and a concept k. The global causality-based visualization is performed in five steps, as shown in Figure 3.

3.2.1. Original concept output space

Let us define the term *concept output space* in order to describe the image's relation to other dataset images, according to a concept. The *concept output space* is the **matrix of the network's outputs**, for a fixed set of images, using only the concepts' dimensions, i.e. zeroing out all the dimensions belonging to other concepts.

We introduce the notation $Activ(\mathcal{I}; \Xi)$ which represents the vector of dimension *NbClasses* used as input of the softmax layer in the network Ξ when we input $\mathcal{I}$.We set at 0 the feature activations not relative to concept k in Ξ, leading to a "new" neural network Ξ_k. The other feature activations are left intact. Then, as illustrated in Figure 3 **(1)**, we compute the following class-aware matrix, with the images' activations of class c, for the $MAG(k)$, representing the original **concept output space**:

$$OS_{Activ}(c, k) = \left(Activ(\mathcal{I}^c_{i_{local}}; \Xi_k)\right)_{i_{local} \in [1, NbIm(c)]} \tag{6}$$

3.2.2. Concept output space under input occlusion

Our causal-based visualization employs patch occlusions to identify influential image regions. Depending on the patch size, we create image-specific **concept output spaces** by individually occluding each patch.

We divide the image $\mathcal{I}_j^c$, where $j \in [1, NbIm(c)]$ that we want to visualize into four patches of the same size $s_p = \frac{s_{image}}{2}$: $\{\mathcal{P}(\ell_x, \ell_y, s_p)\}_{(\ell_x,\ell_y)\in\{0,1\}^2}$ (this partitioning assumes that the image size is a multiple of 2). We perform occlusion on each patch $\mathcal{P}(\ell_x, \ell_y, s_p)$ individually, resulting in a partially occluded image $\blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}_j^c)$. That will replace the original image $\mathcal{I}_j^c$ in the original sequence $\left(\mathcal{I}_{i_{local}}^c\right)_{i_{local}\in[1,NbIm(c)]}$.

Let us define $Occ_{\mathcal{I}_j^c}(\mathcal{I}_{i_{local}}^c) := \blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}_{i_{local}})$ when $i_{local} = j$ and $\mathcal{I}_{i_{local}}^c$ otherwise. Therefore, as presented in Figure 3 **(2)**, the matrix representing the under-occlusion **concept output space**, of image $\mathcal{I}_j^c$ and patch $\mathcal{P}(\ell_x, \ell_y, s_p)$, is: $OS_{\mathcal{P}(\ell_x,\ell_y,s_p),Activ}(c,k) = \Big(Activ(Occ_{\mathcal{I}_j^c}(\mathcal{I}_{i_{local}}^c)); \Xi_k)\Big)_{i_{local}\in[1,NbIm(c)]}$

3.2.3. Measuring patch importance

As we have the original and each occluded-patch **concept output space** for an image, we can verify the changes from the original to the under-occlusion spaces. To create our globally aware visualization, we need to verify this impact on the complete space. We propose to use a **ranking-based** approach to measure the difference between the original and an under-occlusion **concept output space**. We name this approach *Class-aware Order Correlation* ($\mathcal{CAOC}$).

The ranking structure is based on a *target class*, by ordering the points in the **concept output space** from the higher to the lower responses to that class (class-aware). Note that the order of one point depends also on the response of the other points in the space (global awareness). This makes the comparison between the original **concept output space** ranking and the under-occlusion **concept output space** ranking to provide the understanding of how the space changes under a specific occlusion.

We measure the difference in rankings to determine an importance score. To evaluate the effect of the patches' occlusions, we use rankings. We argsort the values for a class c in the data points on $OS_{Activ}(c,k)$ and $OS_{\mathcal{P}(\ell_x,\ell_y,s_p),Activ}(c,k)$ to obtain the sequence of positions of the class scores, sorted from highest to lowest:

$$Seq_c = argsort\left(OS_{Activ}(c,k)_c, decreasing\right) \tag{7}$$

$$Seq'_c = argsort\left(OS_{\mathcal{P}(\ell_x,\ell_y,s_p),Activ}(c,k)_c, decreasing\right). \tag{8}$$

In practice, in the matrices where each row is a data point (a sample's activations), the activation from class c corresponds to column c. Then we compare these sequences. As this sequence of positions are rankings, they can be compared using ranking correlation metrics such as Kendall-tau ($\mathcal{K}$). Calculating this correlation measures how much the patch absence impacts the complete space. This way, the score of patch $\mathcal{P}(\ell_x,\ell_y,s_p)$ is obtained by: $\mathcal{CAOC}(\ell_x,\ell_y) = \mathcal{K}(Seq_c, Seq'_c)$.

However, as only one image was occluded, we propose to use a simpler calculation of importance. Let us define the image $\mathcal{I}_j^c$ position in the original sequence Seq_c as $InitPos$ and the position of $\blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}_j^c)$ in the new sequence Seq'_c as $NewPos^{\blacksquare}_{\ell_x,\ell_y}$. We define the importance of $\mathcal{P}(\ell_x,\ell_y,s_p)$ as: $\mathcal{CAOC}(\ell_x,\ell_y) = \left|InitPos - NewPos^{\blacksquare}_{\ell_x,\ell_y}\right|$ (see Figure 3 **(3)**).

3.2.4. Choosing important patches

The higher the score, the more important is the patch. We use a percentage of the score of the most important patch to define a threshold. Then, we use this threshold to filter the importance of other patches. The visualization consists only of sufficiently important patches. We want to consider

not only the highest score as important for visualization. However, if we visualize all the patches and their respective scores, we obtain a more confusing and noisy visualization. Therefore, we use a threshold thr based on δ: $thr = \max\left(\{\mathcal{CAOC}(\ell_x, \ell_y)\}_{(\ell_x,\ell_y)}\right) \times \delta$. All patches with higher importances will remain in the process (see Figure 3 **(4)**).

3.2.5. Reducing patch size to repeat process hierarchically

We perform new occlusions of smaller patch sizes in the sufficiently important patches by repeating steps from 2. We stop reducing patch sizes when we reach a predefined smallest size. We compose final patch importance by adding up all patch sizes' importances. We continue recursively the procedure in the patches satisfying the inequality $Imp(\ell_x, \ell_y) \geq thr$, returning to step 2. This time, with reduced patch size, while s_p is greater than or equal to s_p^{min}.

During this recursive procedure, each position $(x, y) \in \mathcal{D}$ may have been treated several times. We deduce the *accumulated importance* of a position (x, y) relative to the image $\mathcal{I}_i$ by summing all the computed importance terms where this position was occluded. The final result is called the *accumulated importance matrix*, and we denote it $\mathcal{M}_{Imp}$ (see Figure 3 **(5)**). We finally multiply the initial image by $\mathcal{M}_{Imp}$ and we plot it. In doing so, we have highlighted important regions.

4. Experiments and results

Here, we present visualizations of the clusters' dispersion obtained with MAGE, and qualitative experiments to visually compare Ms-IV with other

methods. We reinforce the results with quantitative evaluation (*Robustness, Faithfulness* and *Localization*). Extra experiments in Appendix C.

To complete the experimentation, we present a concept and bias discovery evaluation with humans. Experiments were performed using two CNN architectures, ResNet-18 [18] and VGG16 [39], and datasets: cats vs. dogs [1], and CUB-200-2011 [19] dataset for *Localization* evaluation. Models trained with initial weights from Imagenet, learning rate $1e-7$, cross-entropy loss, the Adam optimizer and early stop in 20 epochs of non-improving validation loss. Code available at `https://github.com/CarolMazini/unsupervised-IVC`.

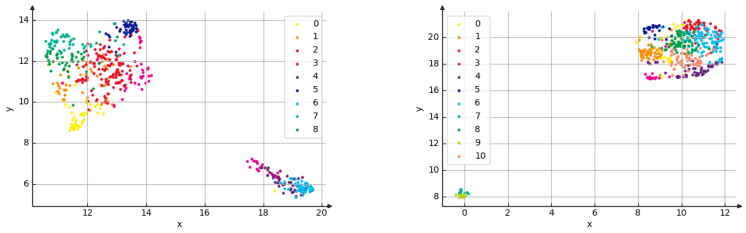
The cat vs. dog dataset has two classes, dog and cat, with 19,891 images (9,936 dogs and 9,955 cats) in the training set, 5,109 images (2,564 dogs and 2,545 cats) in the validation set, and 12,499 in the test set (without labels). The CUB-200 dataset has 200 classes featuring different bird species. Besides the class annotation (200 labels), it has the bird bounding box and parts annotation (15 different parts including back, beak, and belly). To more controlled evaluation, we merged 20 different species of warblers and 20 species of sparrows to separate sparrows from warblers. We used the training/validation split provided in the dataset. The two classes each one have 600 images for training and 600 images for validation, totaling 1,200 images in the training set and 1,200 in the validation set.

4.1. Scatter plot of MAG

We performed experiments with different patch sizes to generate feature map representations (complete experiments in Appendix C). We chose the

[1] https://www.kaggle.com/competitions/dogs-vs-cats-redux-kernels-edition/data

one with good final separation of clusters, without being too small (to reduce computation).



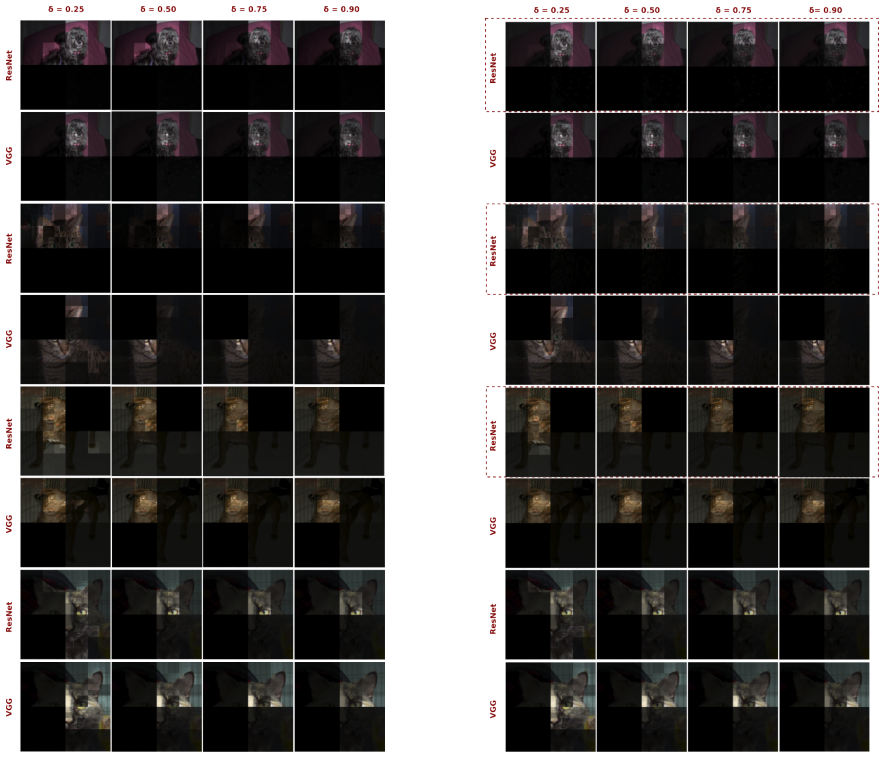
(a) VGG 9 clusters (b) ResNet 11 clusters

Figure 4: Projection of 5120-dimensional representations, from cat vs. dog trained model, to 2D with UMAP. Figures (a) and (b) represent the plots of feature map dimensions from VGG16 and ResNet-18 respectively. Colors represent clusters obtained by K-means. We use $k = 9$ for VGG and 11 for ResNet, chosen by the Elbow curve method using Inertia.

We show in Figure 4 the dispersion of obtained clusters (high dimensionality reduced to 2D using the Uniform Manifold Approximation and Projection (UMAP) algorithm [27]). To generate the representation, we use a subset of 512 images (half from each class), $n = 4$ (patches' size in representation) and $t = 5$ (number of patches per image). To group the concepts, we use K-means, with $k \in [2, 25[$ selected by the Elbow curve method and Inertia. The scatter plots, even with the 5120-dimensional representations reduced to 2-dimensional, show the separations of "clusters of concepts".

4.2. Multiscale visualization

We present three visualization experiments: i) visualizations based on *thr* at $\{0.25, 0.50, 0.75, 0.90\}$ for changing the acceptance of important patches (calculated by the percentage *thr* of the maximum patch importance); ii)



(a) $\mathcal{CAOC}$ (b) *PD*s

Figure 5: Ms-IV visualizations using $\mathcal{CAOC}$ metric to measure patch importance for image samples (two dogs and two cats) using VGG16/ResNet-18 (explanations in the text).

visualizations comparing $\mathcal{CAOC}$ and *PD*s; and, iii) visualizations comparing Ms-IV, Integrated Gradients (IG) and Occlusion (OC).

In Fig. 5(a) we show results for two values of *thr* (0.25 and 0.9) in two image samples and two architectures, using $\mathcal{CAOC}$ to obtain patches contributions. The final value of *thr* (last column) shows more focused attention (less bright regions). ResNet18 network focuses more on the animal's eyes than VGG. By switching the metric from $\mathcal{CAOC}$ to *PD*s (using the same visualization multiscale process), we obtain Fig. 5(b). For most visualizations,

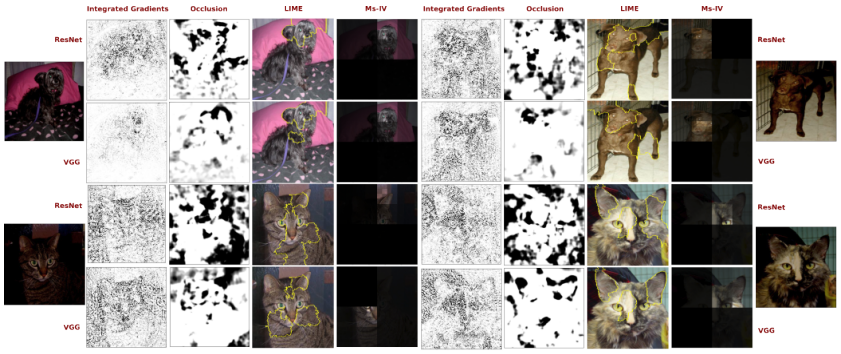


Figure 6: Attribution maps by IG and OC methods versus Ms-IV: IG and OC do not allow us to recognize the shapes of the dog or the cat, whereas Ms-IV, illuminating the initial image, enables us to easily recognize which part of the image is important in the model's decision. Three interpretable components were used for LIME visualizations. Occlusion method uses 7x7 patches, while for Ms-IV we used $thr = 0.75$.

we obtained the same light regions. However, for the examples highlighted by a red dotted rectangle, we see differences: for the dog image ($thr = 0.25$) the dog's paw is highlighted only by $\mathcal{CAOC}$.

These metrics serve for different purposes, $\mathcal{CAOC}$ is sparsity-aware: $\mathcal{CAOC}$ metric will differ from *PD*s when the model's output space changes density. If the new patch-disturbed image falls in a sparse space region, the patch importance should be smaller, as the region was "less modified" according to the model. We present in Fig. 6 the comparison between Ms-IV and two xAI methods: IG and OC.

4.3. Quantitative evaluation

Papers such as Bommer et al. [7], analyze the use of metrics such as *Complexity*, *Robustness*, *Faithfulness* and *Localization*, directed towards specific xAI applications. We will discuss their use in our context.

The *complexity* is evaluated based on the number of presented important features. A less complex visualization has fewer very important features. Ms-IV uses a δ parameter to regulate this criterion. Higher δ highlights fewer patches, facilitating interpretation.

Robustness evaluates the impact of adversarial attacks (changing or not the classification) on the explanations. For attacks that change the sample's class, we can expect a different explanation (**Misclassification**), however, for other attacks that do not change the sample's class, we expect the explanations to remain the same (**Preserved Class**).

To this evaluation, we use the *Worst Case Evaluation* proposed by Huang et al. [20]. The method applies a genetic algorithm to find the worst perturbation (adversarial example) for the interpretability of an image explanation. We generate two types of perturbation: one to change the classification but not the explanation; and another to change the explanation but not the classification. We perturb 30 images (15 per class) by applying a genetic algorithm with 100 iterations, population size of 100 particles and selection of 20 particles for next iterations (reduced numbers due to computational resource limitations). We use Pearson's correlation to compare the original and perturbed image explanations. We compare Ms-IV to IG and LIME.

The results in Table 1a show high **Preserved class** *robustness* for Ms-IV, closer to the method IG (the best for **Preserved class** according to Huang et al. [20]). Methods such as IG are high resolution (pixel-level importance), and have high robustness when modifications do not change classification results (Table 1a **Preserved Class**). However, they lose interpretability (example in Figure 6) with noisy visualizations and in robust-

Table 1: *Robustness* and *Faithfulness* analysis in the cat vs. dog dataset. (a) *Robustness* values are calculated using Worst Case Evaluation for **Preserved Class** and **Misclassification** for three visualization methods: IG, LIME and Ms-IV. Results are derived using Pearson's correlation between the original image and Worst Case visualizations. Higher values are expected for **Preserved Class** and lower values for **Misclassification**. Ms-IV presents a good trade-off between high **Preserved Class** and low **Misclassification**. (b) *Faithfulness* analysis based on the percentage of class changes after occlusion (cl.change), decrease of class output value (Decrease), increase of class output value (Increase) using LIME and Ms-IV directed occlusions of 512 images. Additionally, we present the methods' comparison of biggest variation (absolute output class difference) under occlusion (>). Ms-IV presents bigger output variations.

VGG		
	Preserved Class	**Misclassification**
IG	**0.41**	0.52
LIME	-0.02	**0.08**
Ms-IV	0.34	0.14
ResNet		
	Preserved Class	**Misclassification**
IG	**0.43**	0.50
LIME	0.078	**0.01**
Ms-IV	0.26	0.29

(a)

VGG				
	cl. change	**Decrease.**	**Increase.**	**>**
LIME	0.03	0.83	0.12	0.27
Ms-IV	**0.08**	0.80	0.10	**0.72**
ResNet				
	cl. change	**Pos.**	**Neg.**	**>**
LIME	0.06	0.70	0.23	0.30
Ms-IV	**0.08**	0.65	0.25	**0.69**

(b)

ness when modifications alter sample's class (Table 1a **Misclassification**). LIME significantly improves interpretability, but despite its high robustness for misclassification (Table 1a **Misclassification**), it does not have as much robustness as IG for preserved classes (Table 1a **Preserved Class**).

Faithfulness refers to how much a change in an important feature changes the model's response. For this metric, it is expected to have different outputs (and even different classification) after perturbations. As we constructed an occlusion-based visualization, we already account for perturbations' impact in the explanations. However, we want to verify if the use of occlusion to construct our visualization induced a higher *faithfulness* to the visualization

method. To provide a fairer comparison, we compare Ms-IV to LIME, as it also visualizes image regions instead of pixels.

Results in Table 1b were obtained using 512 images (256 from each class). We extract the important region of each image according to both methods: LIME and Ms-IV. We use these regions as masks to occlude the most important image parts. The results in the table represent the percentage of class changes after occlusion (cl.change), decrease of class output value (Decrease) and increase of class output value (Increase).

Additionally, we evaluate which method disturbs the model's output more in important regions checking, for each image, whether LIME or Ms-IV had the greatest variation (in terms of absolute output class difference) when occluded (>). Ms-IV shows a greater number of output variations with 72% of images for the VGG model, and 69% of images for the ResNet model, indicating higher *faithfulness* of the selected important regions to the model.

Ms-IV presents a trade-off between the two methods, IG and LIME (Figure 1a), and provides interpretable components that, when occluded, have a bigger impact than LIME occlusion components (Table 1b).

The criterion *Localization* refers to the ability of a well-trained model to locate the object of interest in the image (of the correct class). For example, in a cat/dog classification problem, if we have a cat for which the model provides the correct answer, the expectation is that the explanation shows the important region inside the cat region (considering an unbiased model).

As in this paper, we aim to decompose and visualize concepts, so the idea of *localization* needs to be adapted. We want to evaluate if a MAG (concept cluster) can show the same concept in different images, relying on Ms-IV.

To evaluate this, we produce MAGs visualizations using different images and the methods Ms-IV and LIME. Then, we use the human eye to label the displayed areas as different animal parts (a total of 14 different labels).

As our focus is to be globally-aware, we consider that a good-*Locatization* method should show, for different images and the same MAG, the same highlighted animal parts. We also evaluate the original idea of *localization* by calculating the percentage of background highlighted by the visualizations (the lower, the better). We use 12 individuals to label the 200 image visualizations (reduced amount of visualizations is due to limited human resources).

Table 2a presents the results for 10 images of 5 MAGs from both models, VGG and ResNet (total of 200 visualizations). For each MAG, we show the most frequently labeled animal part within its percentage of appearance in the analyzed images (the higher, the better). Subsequently, we show the background percentage in each MAG. MS-IV had the best conventional *localization* rates (highlighting fewer background regions) and presented higher percentages of the same concept for each MAG, especially for VGG.

To reinforce the comparison, we also perform a localization experiment with an already parts-labeled dataset, CUB-200-2011 [19]. The dataset has 200 bird classes with 60 images each. We use 20 classes of Warblers together, and 20 classes of Sparrows together, to compose a binary classification problem. We train a VGG16 and a ResNet-18 model for this problem, obtaining a validation accuracy higher than 95%. There are 16 coordinates of annotated parts per image: *back, beak, belly, breast, crown, forehead, left eye, left leg, left wing, nape, right eye, right leg, right wing, tail, throat.*

First, we find the MAGs for both models. For these bird classification

models: VGG has 12 concepts and ResNet has 9 concepts. Second, we find the 100 most activated images for each MAG (each concept). Third, we generate the visualization, LIME and Ms-IV, for each isolated MAG for its 100 most activated images. The idea is, using the most activated images, to explicitly visualize the concepts. Finally, we calculate the centroid of the highlighted regions in the visualizations and compare it with the parts coordinates. We consider the one with the closest coordinate as the visualized part. If the visualization centroid is outside the bird bounding-box, we consider it a background highlight. Tables 2b and 2c present the results for both models using LIME and Ms-IV.

These results evince better localization using Ms-IV with less background highlights. We also present the top 2 most frequent concepts in each cluster. The ideal is to have a high percent of images highlighting few parts. Ms-IV presents an improvement of LIME results, specially for ResNet model.

4.4. Knowledge Discovery

In these experiments, we apply the ensemble of methods to *find concepts* and *detect bias*. To measure the provided interpretability, the produced visualizations were analyzed by the 24 individuals selected from computer and non-computer experts from two countries: Brazil and France. They were a total of 11 computer experts, including people from industry and academia. There were a total of 13 non-computer experts including people from non-academic domains and from the three main domains (in similar quantities): humanities (sociology, geography, architecture), biological sciences (medicine, physical education) and exact sciences (mathematics) in different educational stages (undergraduate, graduate and post-graduate).

Table 2: *Localization* for cat vs. dog and CUB datasets. (a) Cat vs. dog dataset considering 5 MAG's concepts (10 images each) for VGG and ResNet evaluated in two ways: Conventional *localization* – the percentage of background considered important (the lower the better); and concept *localization* – the quantity of the same concept in each cluster (the higher, the better). Ms-IV presents better *localization* according to 200 visualizations labeled by 12 individuals. For CUB dataset trained models (Warblers vs. Sparrows task): (b) percent of background *localization* and the top2 most-found concepts per MAG, and; (c) concepts *localization* for each MAG. In average, Ms-IV provides better results.

VGG	0	1	2	3	4
LIME	0.3 faces / 0.2 bellow eyes	0.5 eyes' region	0.5 eyes' region	0.3 chest / 0.4 muzzle	0.3 mouth
Ms-IV	**0.4 eyes**	**0.6 eyes' region**	**0.6 eyes' region**	**0.7 muzzle**	**0.4 eyes / 0.2 mouth**
% background					
LIME	0.2	0.3	0.3	0.0	0.0
Ms-IV	**0.1**	**0.2**	**0.2**	0.0	0.0

ResNet	0	1	2	3	4
LIME	0.3 eyes and muzzle	0.4 eyes' region	0.4 eyes' region	0.2 muzzle / 0.2 fur	0.3 eyes' region
Ms-IV	0.3 eyes and forehead	**0.3 eyes / 0.4 muzzle**	**0.6 eyes' region**	**0.7 eyes**	**0.3 eyes' region / 0.2 muzzle**
% background					
LIME	0.4	0.5	0.2	0.4	0.4
Ms-IV	**0.1**	**0.1**	0.2	**0.0**	**0.1**

(a)

Mean VGG	LIME	Ms-IV
Background	0.33	0.25
Top 2 concepts	0.20	**0.21**
Mean ResNet	**LIME**	**Ms-IV**
Background	0.39	0.26
Top 2 concepts	0.15	**0.19**

(b)

Cluster	Method	VGG		ResNet		Cluster	Method	VGG		ResNet	
0	LIME	Background	0.26	Background	0.33	6	LIME	Background	**0.24**	Background	0.40
		Nape	0.11	Nape	0.11			Crown	**0.17**	Left wing	0.08
		Back	0.10	Left wing	0.07			Back	**0.09**	Crown	0.07
	Ms-IV	Background	**0.09**	Background	**0.21**		Ms-IV	Background	0.28	Background	**0.26**
		Back	**0.13**	Right wing	**0.11**			Breast	0.10	Right wing	**0.11**
		Crown	**0.12**	Tail	**0.09**			Nape	0.09	Beak	**0.08**
1	LIME	Background	0.21	Background	0.55	7	LIME	Background	0.67	Background	0.31
		Nape	0.11	Tail	0.07			Crown	0.08	Crown	0.11
		Back	0.09	Left wing	0.06			Forehead	0.05	Right wing	0.08
	Ms-IV	Background	**0.18**	Background	**0.24**		Ms-IV	Background	**0.43**	Background	**0.29**
		Beak	**0.13**	Crown	**0.13**			Right wing	**0.07**	Nape	**0.12**
		Tail	**0.10**	Tail	**0.10**			Tail	**0.07**	Back	**0.09**
2	LIME	Background	0.36	Background	0.39	8	LIME	Background	**0.28**	Background	**0.33**
		Left wing	0.09	Back	0.08			Nape	**0.12**	Tail	**0.08**
		Nape	0.08	Crown	0.07			Back	**0.10**	Back	**0.07**
	Ms-IV	Background	**0.14**	Background	**0.19**		Ms-IV	Background	0.37	Background	0.41
		Nape	**0.12**	Right wing	**0.11**			Nape	0.09	Tail	0.08
		Left wing	**0.11**	Beak	**0.11**			Crown	0.08	Crown	0.07
3	LIME	Background	**0.15**	Background	0.38	9	LIME	Background	**0.28**		
		Nape	**0.14**	Beak	0.08			Tail	**0.10**	-	-
		Crown	**0.14**	Back	0.08			Back	**0.10**		
	Ms-IV	Background	0.23	Background	**0.26**		Ms-IV	Background	0.36		
		Right wing	0.10	Right wing	**0.09**			Back	0.09	-	-
		Back	0.10	Breast	**0.09**			Tail	0.08		
4	LIME	Background	0.32	Background	0.47	10	LIME	Background	0.64		
		Back	0.12	Left wing	0.06			Back	0.09	-	-
		Crown	0.11	Back	0.05			Right wing	0.05		
	Ms-IV	Background	**0.25**	Background	**0.25**		Ms-IV	Background	**0.30**		
		Right wing	**0.12**	Beak	**0.12**			Right wing	**0.13**	-	-
		Beak	**0.09**	Nape	**0.10**			Tail	**0.09**		
5	LIME	Background	0.42	Background	0.40	11	LIME	Background	0.21		
		Back	0.10	Nape	0.08			Left wing	0.11	-	-
		Crown	0.09	Beak	0.07			Tail	0.10		
	Ms-IV	Background	**0.25**	Background	**0.25**		Ms-IV	Background	**0.22**		
		Right wing	**0.13**	Right wing	**0.09**			Right wing	**0.17**	-	-
		Nape	**0.12**	Back	**0.09**			Tail	**0.11**		

(c)

(a) Visualizations 2 clusters ResNet-18

(b) Visualizations 3 clusters VGG16

Figure 7: Some visualizations obtained for clusters 0 and 1 of ResNet-18, and clusters 2, 5 and 6 of VGG16 (other visualizations in Appendix C.4). We present the selected concepts for these clusters, by 24 participants, to describe the two classes. According to the answers, for ResNet-18: cluster 0 presents the **eye** and **muzzle** of cats, highlighting **eye** and **ear** of dogs. Cluster 1 presents **eye** for both classes and **muzzle** for dogs. For VGG16: cluster 2 presents **eye** for both classes. Cluster 5 detects the **eye** for cats and the **muzzle** and **paws** for dogs. Cluster 6 presents *ears* of cats and, the *muzzle* and *eyes* for dogs.

Finding concepts: We selected six MAG-generated clusters from ResNet-18 and VGG16. We visualized each cluster through Ms-IV applied to 16 images (8 cats and 8 dogs) from the top-middle-ranked positions. From a ranking of 512 images, we started at position 100 to avoid sparsity in higher and lower positions (possible outliers). We presented the Ms-IV visualizations of these image subsets to the research participants and asked which animal part corresponded to the lighter regions in dogs and cats. There were a total of 12 image subsets (limited analysis to six clusters per network).

From the 13 concepts, fewer than three of them received most of the participants' votes for cluster. There was an agreement about concepts for both

computer and non-computer experts. Concepts such as **eyes** and **muzzle** were the most frequently observed. We highlight Fig. 7 as an example of high agreement and variability of concepts: **eye**, **muzzle**, **paws** and **ear**.

Table 3: From a total of 24 participants and 8 different bias/non-bias comparison, 77% of the responses showed the non-bias group choice as a better separation (results using the cat vs. dog model). We display the values for computer (**Comp.**) and non-computer (**Non-comp.**) experts to make a selection between **Bias** and **Non-bias**. Even non-computer experts present a high percentage of the non-bias choice.

	Comp.	Non-Comp.	Total		Comp.	Non-Comp.	Total
Bias 0	3	4	7	**Bias 4**	0	1	1
Non-bias 0	8	9	17	**Non-bias 4**	11	12	23
Bias 1	3	3	6	**Bias 5**	2	4	6
Non-bias 1	8	10	18	**Non-bias 5**	9	9	18
Bias 2	2	4	6	**Bias 6**	2	5	7
Non-bias 2	9	9	18	**Non-bias 6**	9	8	17
Bias 3	0	0	0	**Bias 7**	3	7	10
Non-bias 3	11	13	24	**Non-bias 7**	8	6	14
				Bias Total	15	28	43
				Non-bias Total	73	76	149
				% Non-bias Total	82%	73%	77%

Bias Detection: We compare a biased and a less biased model (more accurate). The analyzed ResNet-18 model is the less biased, we call it the normal one. We train an extra ResNet-18 model, initialized with ImageNet weights, and trained with 100 images, 50 dark cats and 50 beige dogs.

We generated the biased and unbiased ResNet-18 image subsets to each *concept* (as in the **finding concepts** part). We paired one biased ResNet-18 group and one normal ResNet-18 group. We asked the participants which of the models seemed to highlight only the important parts to differentiate cats and dogs, without explaining Neural Network bias.

Results in Table 3 show, both computer and non-computer experts found, with high accuracy, the unbiased model (73% of correct for the non-computer experts). Our visualization facilitates model analysis, even for non-specialists.

5. Conclusion

We propose a new way to make CNNs interpretable thanks to the combination of *MAGE*, to group feature maps into "concepts", and *Ms-IV*, to provide a simple (multiscale) understanding of the model's knowledge. *CaOC* metric is used to consider the structure and organization of the model's final decision space and provides global awareness of sample perturbation. In the future, we plan to improve the "clusters' quality" through hierarchical/spectral clustering techniques, and to introduce more subtle segmentation techniques using mathematical morphology in our multiscale visualization.

References

[1] S. H. P. Abeyagunasekera, Y. Perera, K. Chamara, U. Kaushalya, P. Sumathipala, O. Senaweera, Lisa : Enhance the explainability of medical images unifying current xai techniques, in: 7th International conference for Convergence in Technology (I2CT), 2022, pp. 1–9.

[2] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xAI), IEEE Access 6 (2018) 1–23.

[3] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, F. Herrera, Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence, Information Fusion 99 (2023) 101805.

[4] N. Aslam, I. U. Khan, S. Mirza, A. AlOwayed, F. M. Anis, R. M. Aljuaid, R. Baageel, Interpretable machine learning models for mali-

cious domains detection using explainable artificial intelligence (xai), Sustainability 14 (12) (2022) 7375.

[5] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Muller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (7) (2015) 1–46.

[6] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3319–3327.

[7] P. Bommer, M. Kretschmer, A. Hedström, D. Bareeva, M. M. Höhne, Finding the right XAI method - A guide for the evaluation and ranking of explainable AI methods in climate science, CoRR 2303.00652 (2023).

[8] J. Born, N. Wiedemann, M. Cossio, C. Buhre, G. Brändle, K. Leidermann, J. Goulet, A. Aujayeb, M. Moor, B. Rieck, K. Borgwardt, Accelerating detection of lung pathologies with explainable ultrasound image analysis, Applied Sciences 11 (2) (2021).

[9] K. Borys, Y. A. Schmitt, M. Nauta, C. Seifert, N. Krämer, C. M. Friedrich, F. Nensa, Explainable ai in medical imaging: An overview for clinical practitioners - beyond saliency-based xai approaches, European Journal of Radiology (2023).

[10] A. Chaddad, J. Peng, J. Xu, A. Bouridane, Survey of explainable ai techniques in healthcare, Sensors 23 (2) (2023).

[11] T. Chen, Applications of xai for forecasting in the manufacturing domain, in: SpringerBriefs in Applied Sciences and Technology, SpringerBriefs in Applied Sciences and Technology, Springer Science and Business Media Deutschland GmbH, 2023, pp. 13–50.

[12] B. Crook, M. Schlüter, T. Speith, Revisiting the performance-explainability trade-off in explainable artificial intelligence (xai), arXiv (2023).

[13] A. Ghorbani, J. Wexler, J. Z. Y, B. Kim, Towards automatic concept-based explanations, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 1–10.

[14] R. Gu, G. Wang, T. Song, R. Huang, M. Aertsen, J. Deprest, S. Ourselin, T. Vercauteren, S. Zhang, Ca-net: Comprehensive attention convolutional neural networks for explainable medical image segmentation, IEEE Transactions on Medical Imaging 40 (2021) 699–711.

[15] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Computing Surveys 51 (5) (2018) 1–42.

[16] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (2002) 389–422.

[17] A. Haghanifar, M. M. Majdabadi, Y. Choi, S. Deivalakshmi, S. Ko, Covid-cxnet: Detecting covid-19 in frontal chest x-ray images using deep learning, Multimedia Tools and Applications 81 (2022) 30615–30645.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[19] X. He, Y. Peng, Fine-grained visual-textual representation learning, IEEE Transactions on Circuits and Systems for Video Technology PP (2019) 1–12.

[20] W. Huang, X. Zhao, G. Jin, X. Huang, Safari: Versatile and efficient evaluations for robustness of interpretability, in: International Conference on Computer Vision (ICCV), 2022, pp. 1–10.

[21] T. Hulsen, Explainable artificial intelligence (xai): Concepts and challenges in healthcare, AI 4 (3) (2023) 652–666.

[22] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, R. Sayres, Interpretability beyond feature attribution: Quantitative testing with Concept Activation Vectors (TCAV), in: 35th International Conference on Machine Learning (ICML), 2018, pp. 2668–2677.

[23] H. Li, J. G. Ellis, L. Zhang, S.-F. Chang, Patternnet: Visual pattern mining with deep neural network, in: Proceedings of the 2018 ACM on international conference on multimedia retrieval, 2018, pp. 291–299.

[24] X. Li, H. Xiong, X. Li, X. Zhang, J. Liu, H. Jiang, Z. Chen, D. Dou, G-lime: Statistical learning for local interpretations of deep neural networks using global priors, Artificial Intelligence 314 (C) (2023).

[25] S. Lu, Z. Zhu, J. M. Gorriz, S.-H. Wang, Y.-D. Zhang, Nagnn: Classification of covid-19 based on neighboring aware representation from deep

graph neural network, International Journal of Intelligent Systems 37 (2022) 1572–159.

[26] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: 31st International Conference on Neural Information Processing Systems (NeurIPS), 2017, pp. 4768–4777.

[27] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv (2018).

[28] D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, M. Rohrbach, Multimodal explanations: Justifying decisions and pointing to the evidence, in: 31st International Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8779–8788.

[29] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.

[30] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the predictions of any classifier, in: 22nd International Conference on Knowledge Discovery and Data Mining (KDD), 2016, pp. 1135–1144.

[31] P. D. S, R. K. K, V. S, N. K, A. K, An overview of interpretability techniques for explainable artificial intelligence (xai) in deep learning-based medical image analysis, in: 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Vol. 1, 2023, pp. 175–182.

[32] G. Schwalbe, B. Finzel, A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts, Data Mining and Knowledge Discovery (2023).

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: 16th International Conference on Computer Vision (ICCV), 2017, pp. 618–626.

[34] S. M. Shankaranarayana, D. Runje, Alime: Autoencoder based approach for local interpretability, in: 19th Intelligent Data Engineering and Automated Learning (IDEAL), Springer International Publishing, 2019, pp. 454–463.

[35] C. E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (1948) 379–423.

[36] Y. Shen, N. Wu, J. Phang, J. Park, K. Liu, S. Tyagi, L. Heacock, S. G. Kim, L. Moy, K. Cho, K. J. Geras, An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization, Medical Image Analysis 68 (2021) 101908.

[37] R.-K. Sheu, M. Pardeshi, K.-C. Pai, L.-C. Chen, C.-L. Wu, W.-C. Chen, Interpretable classification of pneumonia infection using explainable ai (xai-icp), IEEE Access PP (2023) 1–1.

[38] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: 34th International Conference on Machine Learning (ICML), 2017, pp. 3145–3153.

[39] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1–14.

[40] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, in: 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1–14.

[41] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: 34th International Conference on Machine Learning (ICML), 2017, pp. 3319–3328.

[42] R. T. Sutton, O. R. Zaiane, R. Goebel, D. C. Baumgart, Artificial intelligence enabled automated diagnosis and grading of ulcerative colitis endoscopy images, Scientific Reports 12 (2748) (2022).

[43] R. Tan, L. Gao, N. Khan, L. Guan, Interpretable artificial intelligence through locality guided neural networks, Neural Networks 155 (2022) 58–73.

[44] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, K. N. Ramamurthy, Treeview: Peeking into deep neural networks via feature-space partitioning (2016).

[45] G. Visani, E. Bagli, F. Chesani, Optilime: Optimized lime explanations for diagnostic computer algorithms, arXiv (2020).

[46] M. R. Zafar, N. Khan, Deterministic local interpretable model-agnostic explanations for stable explainability, Machine Learning and Knowledge Extraction 3 (3) (2021) 525–541.

[47] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: 13th European Conference on Computer Vision (ECCV), 2014, pp. 818–833.

[48] Q. Zhang, R. Cao, F. Shi, Y. N. Wu, S.-C. Zhu, Interpreting cnn knowledge via an explanatory graph, in: 32nd AAAI Conference on Artificial Intelligence (AAAI), 2018, pp. 4454–4463.

[49] R. Zhang, P. Madumal, T. Miller, K. A. Ehinger, B. I. P. Rubinstein, Invertible concept-based explanations for CNN models with non-negative concept activation vectors, in: 35th Conference on Artificial Intelligence (AAAI), 2021, pp. 11682 – 11690.

[50] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: 29th Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2921–2929.

[51] Z. Zhou, G. Hooker, F. Wang, S-lime: Stabilized-lime for model explanation, in: 27th Conference on Knowledge Discovery & Data Mining (SIGKDD), 2021, pp. 1–10.

Appendix A. Decomposition of the feature space into concepts: Maximum Activation Groups Extraction (MAGE)

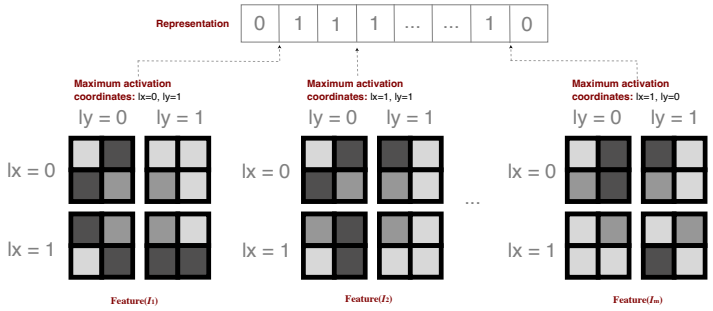


Figure A.8: Finding the "position" of a feature f_p (in each image of) the data set (illustrative example). We start from a sequence $(\mathcal{I}_i)_i$ of images whose domain is of size 4×4. We decompose their common domain into 4 patches of the same size 2×2. On the first image, we observe that among the 4 four subdivisions in the f_p^{th} feature map, the one that maximizes its 1-norm corresponds to $(\ell_x = 1, \ell_y = 1)$, so we write in the vector (depicted above) these values: 1 and then 1. We continue this procedure for the next images until we reach the end of the dataset. This vector represents then where the f_p^{th} feature is located in the images of the data set; we call it the *representative* of the feature number f_p.

We present in Figure A.8 a schematic example of the proposed representation in the MAGE process.

Appendix B. Global causality-based visualization: Multiscale- Interpretable Visualization (Ms-IV)

In Figure B.9 we visually exemplify the impact of $\mathcal{CAOC}$ in a sparse decision space.

In the sequel, we explain in more detail the algorithm and pseudocode of Ms-IV.

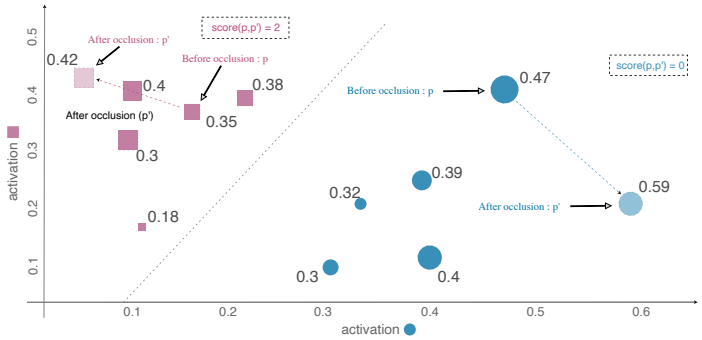


Figure B.9: Explanation of the computation of $\mathcal{CAOC}$ (on a fictitious scenario). Here, our dataset is made up of images of one disk or one square, the output is a class ("disk" or "square"). We have set the concept value to some random k. We plot the activation distributions in a 2D space: the horizontal coordinate represents how much the sample is predicted as being a circle, and vertically, how much it is predicted as being a square. We can see that, by occlusion, one square moves in this space by a distance of 0.07 and one disk by 0.12. However, we need some **quantification** of the impact of a concept on the two classes. Thus, we propose using ranking correlation between before/after the occlusion of the most important patch images relative to the concept k. We find that the disk did not move in the disk's ranking when we did the occlusion (it remains the "strongest" disk), so the correlation $\mathcal{CAOC}(k, \mathit{disks})$ is maximal, meaning that concept k does not have much influence on disks. Conversely, in the squares case, the square's position changes from 3^{rd} to 5^{th} position, leading to a ranking change of 2, thus $\mathcal{CAOC}(k, \mathit{squares})$ is lower, which means that concept k is important for squares.

We propose here an algorithm (see Algorithm 1 depicted in Figure B.10) that uses a multiscale hierarchical approach (such as a quad-tree) capable of highlighting the areas of a given image $\mathcal{I}$ (belonging to $\mathcal{DS}$) that are important to the network's decision regarding the concept k. This approach is *global* in the sense that computations on the entire dataset will have been completed beforehand. Note that our procedure is different from LIME: even

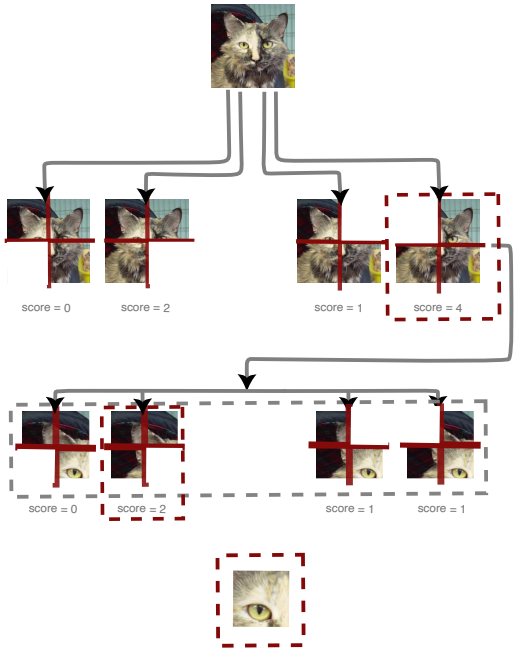


Figure B.10: How our visualization algorithm works. We fix some concept valued k and some image number i. We sort the activations of each image and we call *InitPos* the position of the activation of the current image $\mathcal{I}_i$ in the computed sequence. The goal is to enlighten the areas of the image as much as the concept k is important in each region of this same image. To this aim, we decompose the initial domain into four patches; it is the first step of our recursive subdivision. By occluding separately each of these four patches and computing how much their new position $NewPos^{\blacksquare}_{\ell_x,\ell_y}$ (in the sequence of activations of the occluded images) differs from *InitPos*, we obtain four scores $\left|InitPos - NewPos^{\blacksquare}_{\ell_x,\ell_y}\right|$ (called the importance). Choosing the maximal score, allows us to deduce in which of these four patches the concept is represented most. By continuing this recursive subdivision in the most important patches until we reach the minimal size of a patch, we will know how much we have to illuminate each coordinate in the image (by adding up the importances we have computed for each pixel).

though in both cases, we use parts of images to show the model's knowledge, in our case we illuminate the image relative to one unique concept at a time.

For the sake of simplicity, let us introduce a new term: we define an *occlusion* of the image I of domain $\mathcal{D}$ on a patch $\mathcal{P} \subseteq \mathcal{D}$ as $\blacksquare_{\mathcal{P}}(\mathcal{I}) : \mathcal{D} \to \mathbb{R}$ such that for any $(x, y) \in \mathcal{D}$, $\blacksquare_{\mathcal{P}}(\mathcal{I})(x, y)$ is equal to $\mathcal{I}(x, y)$ when $(x, y) \notin \mathcal{P}$, and 0 otherwise. Now let us formally explain the main steps of our algorithm (we deal with the recursive part of the algorithm using a list that we will not detail here for the sake of simplicity):

1. We fix a visualization threshold ratio $\delta \in]0, 1]$, a minimal patch size $s_p^{min} \in \mathbb{N}^*$ (representing the minimal patch size of a concept), a class c, the image $\mathcal{I}^c_{i_{local}}$, and a concept k.
2. We compute the sequence $Seq = sort\left(\left(Activ_c(\mathcal{I}^c_j, \Xi_k)\right)_{j\in[1,NbIm(c)]}\right)$ and then the position $InitPos$ of $Activ_c(\mathcal{I}^c_{i_{local}}, \Xi_k)$ within it. This position represents "how much" $\mathcal{I}^c_{i_{local}}$ truly belongs to class c.
3. We divide the image into four patches of the same size $s_p = \frac{s_{image}}{2}$ resulting in this partition $\{\mathcal{P}(0, 0, s_p), \mathcal{P}(0, 1, s_p), \mathcal{P}(1, 0, s_p), \mathcal{P}(1, 1, s_p)\}$ (we assume that the image size is a multiple of 2).
4. For each $(\ell_x, \ell_y) \in \{0, 1\}^2$:
   (a) We occlude $\mathcal{I}^c_{i_{local}}$ on the patch $\mathcal{P}(\ell_x, \ell_y, s_p)$ resulting in $\blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}^c_{i_{local}})$.
   (b) We calculate:

$$Seq' := sort\left(\left(Activ_c(\blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}^c_j), \Xi_k)\right)_{j\in[1,NbIm(c)]}\right)$$

   and the position $NewPos^{\blacksquare}_{\ell_x,\ell_y}$ of $Activ_c(\blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}^c_{i_{local}}), \Xi_k)$ in it. It will then represent how much $\mathcal{I}^c_{i_{local}}$ is still of class c after occluding the image in the patch domain. If the initial activation is almost preserved despite the occlusion, we will have $NewPos^{\blacksquare}_{\ell_x,\ell_y} \approx InitPos$.

(c) For this reason, we propose to calculate what we call the *importance* of the patch $\mathcal{P}(\ell_x, \ell_y, s_p)$:

$$Imp(\ell_x, \ell_y) = \left| InitPos - NewPos_{\ell_x, \ell_y}^{\blacksquare} \right|$$

5. We compute a threshold *thr* based on δ: $thr = \max\left(\{Imp(\ell_x, \ell_y)\}_{(\ell_x, \ell_y)}\right) \times \delta$
6. We continue the procedure recursively in the patches that satisfy the inequality $Imp(\ell_x, \ell_y) \geq thr$ while s_p is greater than or equal to s_p^{min}.
7. During this recursive procedure, each position $(x, y) \in \mathcal{D}$ may have been treated several times. We deduce the *accumulated importance* of a position (x, y) relative to the image $\mathcal{I}_i$ by summing all the computed importance terms where this position was occluded. The final result is called the *accumulated importance matrix* and we denote it $\mathcal{M}_{Imp}$.
8. We finally multiply the initial image by $\mathcal{M}_{Imp}$ and we plot it. We have highlighted important regions.

Algorithm 1: Ms-IV algorithm

Input: s_p^{min}; s_{image}; δ; I of class c; concept k;
Output: $\mathcal{M}_{Imp}$: matrix of importances
Data: dataset of squared images $\mathcal{DS}$

$Seq := sort\left(\left(Activ_c(\mathcal{I}_j^c, \Xi_k)\right)_{j\in[1,NbIm(c)]]}\right)$
$InitPos = Position(Activ_c(\mathcal{I}_{i_{local}}^c, \Xi_k), Seq)$
$\dim\_l_n := \frac{s_{image}}{s_p^{min}}$ // dimension of final matrix (smallest patches)
$\mathcal{M}_{Imp} := CreateMatrixOfZeros(\dim\_l_n, \dim\_l_n)$ // final matrix initialization
$level^{\max} := int(\sqrt{\dim\_l_n})$ // final level
$ListOfCoords := \{(0,0)\}$ // level 0 has only patch (0,0)
$s_p = s_{image}$
/* Quadtree-like propagation */
for $level \in [1, level^{\max}]$ **do**
  $s_p := \frac{s_p}{2}$ // new patch size
  $\dim_{level} := 2^{level}$ // side dimension
  $\mathcal{M}_{Imp}^{Aux} := CreateMatrixOfZeros(\dim_{level}, \dim_{level})$
  $\mathcal{M}_{Imp}^{Aux,2} := CreateMatrixOfZeros(\dim\_l_n, \dim\_l_n)$
  /* analysis of selected patches */
  for $(\ell_x, \ell_y) \in ListOfCoords$ **do**
    /* division into 4 patches */
    for $(\ell_a, \ell_b) \in [2\ell_x, 2\ell_x+1] \times [2\ell_y, 2\ell_y+1]$ **do**
      $u := \frac{\dim\_l_n}{\dim_{level}}$ // number of smallest patches
      $Seq' := sort\left((Activ_c(\blacksquare_{\mathcal{P}(\ell_a,\ell_b,s_p)}(\mathcal{I}_j^c), \Xi_k))_{j\in[1,NbIm(c)]}\right)$
      $NewPos^{\blacksquare}_{\ell_a,\ell_b} := Position(Activ_c(\blacksquare_{\mathcal{P}(\ell_a,\ell_b,s_p)}(\mathcal{I}_{i_{local}}^c), \Xi_k), Seq')$
      $Imp = |InitPos - NewPos^{\blacksquare}_{\ell_a,\ell_b}|$ // patch importance
      $\mathcal{M}_{Imp}^{Aux}(\ell_a;\ \ell_b)$ += Imp
      $\mathcal{M}_{Imp}^{Aux,2}(\ell_a u, \ldots, (\ell_a+1)u-1;\ \ell_b u, \ldots, (\ell_b+1)u-1)$ += Imp
  $\mathcal{M}_{Imp}^{Aux} := \frac{\mathcal{M}_{Imp}^{Aux} - \min \mathcal{M}_{Imp}^{Aux}}{\max \mathcal{M}_{Imp}^{Aux} - \min \mathcal{M}_{Imp}^{Aux}}$ // normalization
  $\mathcal{M}_{Imp}$ += $\mathcal{M}_{Imp}^{Aux}$
  /* choice of patches for next level */
  $ListOfAuxiliaryCoords := \{\}$
  $thr = \max(\mathcal{M}_{Imp}^{Aux}) \times \delta$ // finding threshold value for selection
  for $(\ell_a, \ell_b) \in [1, \dim_{level}] \times [1, \dim_{level}]$ **do**
    if $\mathcal{M}_{Imp}^{Aux}(\ell_a;\ \ell_b) \geq thr$ **then**
      $ListOfAuxiliaryCoords := ListOfAuxiliaryCoords \cup \{(\ell_a, \ell_b)\}$
  $ListOfCoords := ListOfAuxiliaryCoords$
  if $thr = 0$ **then**
    break // if no changes in ranking, we early stop

Appendix C. Methods evaluation

We present the experiments to test MAGE, $\mathcal{CAOC}$ and Ms-IV, using two CNN architectures, ResNet-18 [18] and VGG16 [39], trained on a classification dataset of cats vs. dogs [2].

Appendix C.1. Dataset and training

Table C.4: Accuracy values in training and validation sets for ResNet-18 and VGG16. We present the results for each class separately and together (total accuracy). VGG16 presented the best accuracy.

	Train			**Val**		
	Cat	**Dog**	**Total**	**Cat**	**Dog**	**Total**
ResNet	98.60	97.82	98.21	97.93	97.79	97.86
VGG	99.09	99.00	99.04	98.47	98.74	98.61

We used 19,891 (9,936 dogs and 9,955 cats) images in the training set, 5,109 (2,564 dogs and 2,545 cats) images in the validation set, and 12,499 in the test set. For the training, we used the pre-trained networks in the ImageNet dataset. We excluded the networks' original classification layer and included a 2 neuron layer followed by softmax activation. The networks were trained using Cross Entropy loss, Adam optimizer, and a learning rate of $1e-7$. We saved only the model that minimizes validation loss. We present in Table C.4 the accuracy values for each model.

[2] https://www.kaggle.com/competitions/dogs-vs-cats-redux-kernels-edition/data

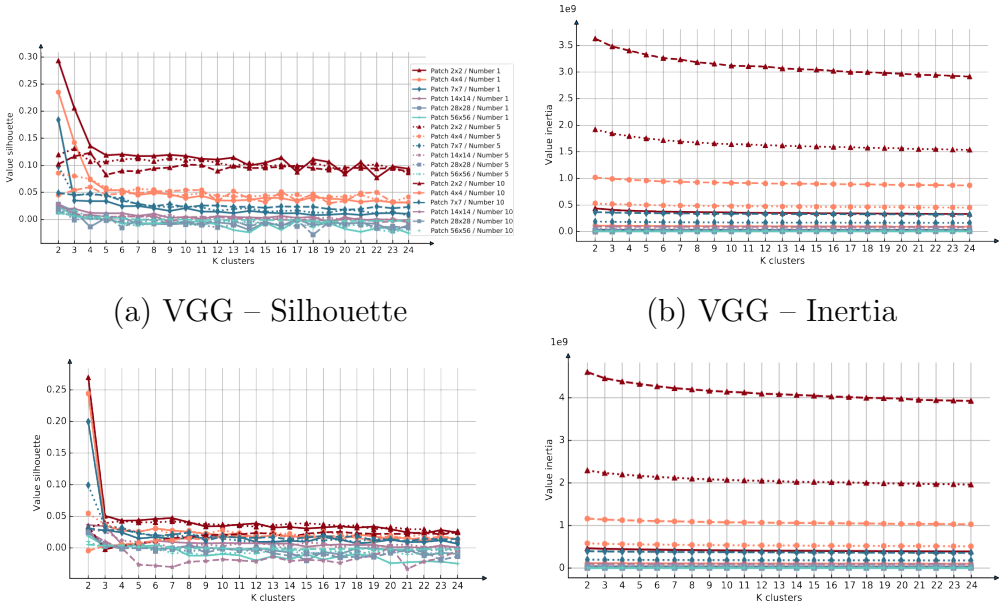
Appendix C.2. Evaluating the quality of MAGE

To test MAGE, we vary the representation patch size $s_p \times s_p$ with the following values: $2, 4, 7, 14, 28$ and 56. The number t of patches from each image to compose the representation is equal to $1, 5$,and 10. We use a subset of 512 images, half from each class. For each configuration, we apply the k-means, and we vary the number of clusters $k \in [2, 25[$. We use the metrics Silhouette and Inertia (distance of each sample to its cluster centroid). Inertia is used to choose the number k of clusters.

Smaller sizes of patches present better quality (higher Silhouette), however, they fail to capture interpretable structures. To analyze fewer configurations, we visualize the dispersion (scatter plots) and central feature maps to each cluster using $n \in \{4, 7, 14\}$, $t = 5$ (intermediate value of t with smaller inertia than $t = 10$). That choice removes the smallest n value (representing less interpretable components), and the bigger n values with less interesting Silhouette values. We selected k, for each configuration by using the Elbow curve method [3] and Inertia.

We show the scatter plots that compare the spatial position of feature maps for patch sizes $n \in \{4, 7, 14\}$ in Fig. C.12 for VGG16. The scatter plot visualizations show greater intra-clusters sparsity for larger patch sizes. It should be noted that for visualization purposes, we reduced the representation dimensionality using the UMAP technique [27], with the parameters $n\_neighbors = 50$, $min\_dist = 0.0$ and Euclidean distance. Colors designate the different assigned clusters for each of them.

[3] https://kneed.readthedocs.io



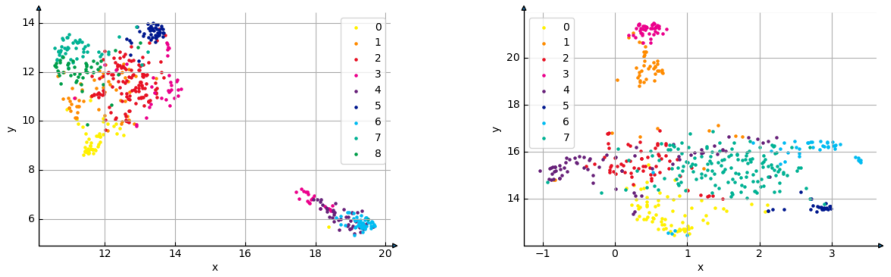
(a) VGG – Silhouette

(b) VGG – Inertia

(c) ResNet – Silhouette

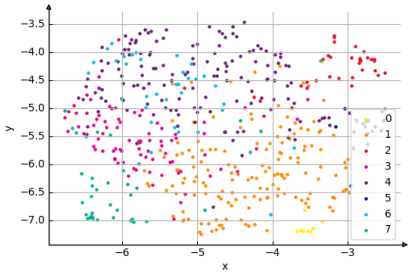
(d) ResNet – Inertia

Figure C.11: Silhouette and Inertia results from K-means with $k \in [2, 25[$ with different sizes and numbers of patches used in the feature map representation, $n \in \{2, 4, 7, 14, 28, 56\}$ and $t \in \{1, 5, 10\}$. Figures (a) and (c) present the Silhouette for VGG16 and ResNet respectively. Figures (b) and (d) present the Inertia for VGG16 and ResNet respectively. The best clusters should maximize the Silhouette, which is between -1 and 1. Inertia is not directly comparable, as we changed representation, but will be used for finding the best number of clusters k. The representations using small patch size seem to improve this mentioned quality.

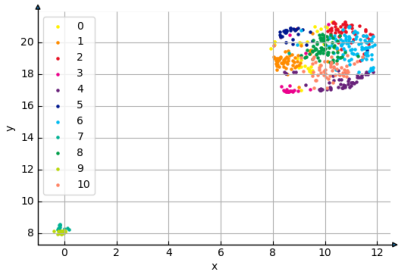


(a) VGG $n = 4$, 9 clusters

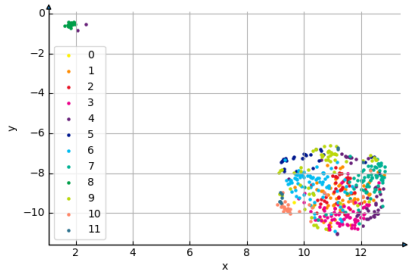
(b) VGG $n = 7$, 8 clusters



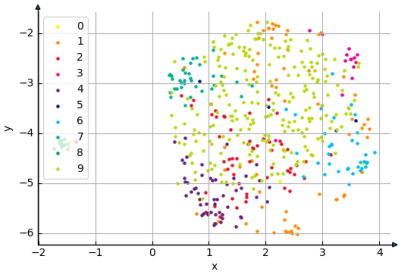
(c) VGG $n = 14$, 8 clusters



(d) ResNet $n = 4$, 11 clusters



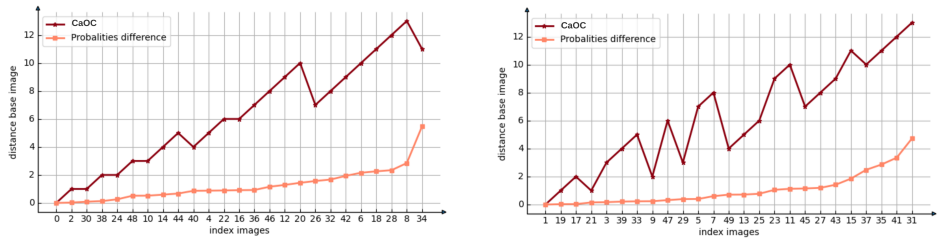
(e) ResNet $n = 7$, 12 clusters



(f) ResNet $n = 14$, 10 clusters

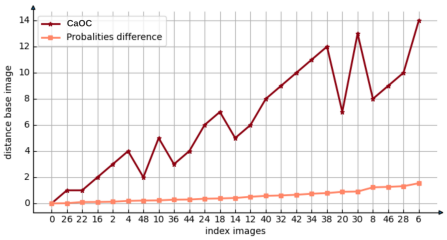
Figure C.12: Smallest values of n seem to present denser clusters. Figures (a), (b) and (c) represent the scatter plots of feature maps of VGG16 while Figures (d), (e) and (f) represent those of ResNet according to the representation using $n \in \{4, 7, 14\}$ and $t = 5$, respectively. The colors represent the clusters obtained by K-means. We used the value k equal to $9, 8$ and 8 for VGG and, $11, 12$ and 10 for ResNet (for each patch size) chosen by the Elbow curve method from the Inertia presented in Figure C.11(b).

These results highlight that the size of the patches is a crucial parameter. Smaller patches can help to provide better concept clusters; however, they will probably capture fewer interpretable structures (same xAI pixel-level technique's problem). Moreover, when using these smaller patches, the number of analyzed regions increases together with the number of computations. On the other hand, big patches will not cluster feature maps well enough, with poorer evaluation results and bigger sparsity (Fig. C.12). As the feature map cluster centers were reasonably similar, we continue the experiments with the $n = 4$ clusters (small but not the smallest).

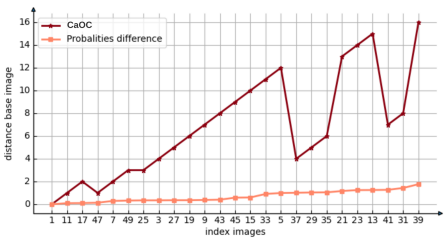


(a) VGG – dog

(b) VGG – cat



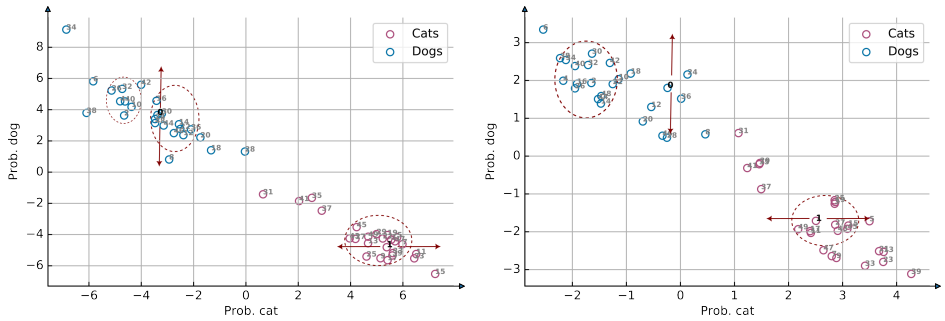
(c) ResNet – dog



(d) ResNet – cat

Figure C.13: *CaOC* and *Probabilities difference* behave differently. Based on a dog (index 0) and cat (index 1) image, we calculated the difference to 24 other images of each class, using *CaOC* and *Probabilities difference*. We ordered the images according to the distances obtained by *Probabilities difference*. *CaOC* presents discontinuities in the graph in relation to this order. Even indexes are dogs (figures (a) and (c)), and odd indexes are cats (figures (b) and (d)).

We show the difference between the $\mathcal{CAOC}$ metric and the *probabilities difference* (*PD*) used as a metric. We selected 50 images from the dataset (to make the visualization easier), 25 from each class (dogs as even numbers and cats as odd numbers), and we calculated the difference from image 0 (dog) and image 1 (cat) to all the others (from their respective classes) using $\mathcal{CAOC}$ and *PD*. We present the results for the dog class in Fig. C.13. In this figure, we ordered the images with respect to the distance obtained by *PD*s to



(a) VGG16 50 images

(b) ResNet18 50 images

Figure C.14: Scatter plots of 50 images using VGG16/ResNet-18 final non-normalized probabilities. $\mathcal{CAOC}$ and *PD* are based on the dog's probability axis for dogs, and the cat's probability axis for cats. Black numbers represent the base samples for the differences in Fig. C.13. ResNet-18 is sparser.

observe the behavior of $\mathcal{CAOC}$ as a function of *PD*. As *PD* increases, $\mathcal{CAOC}$ does not follow a continuous behavior. So we look closer to discontinuities such as sample 34 (Fig. C.13(a)) in Fig C.14. We project 50 samples using their classes' (non-normalized) probabilities as coordinates (that is, the activations before the softmax). Samples from VGG C.14(a) present a denser region close to sample 0 than to samples from ResNet18 C.14(b), which is reflected in Fig. C.13, presenting more ResNet18 discontinuities. This density-awareness is expected from $\mathcal{CAOC}$. The sparsity represents fewer model-informative regions (based on the dataset), which thus count less for deciding on the model's globally important patterns.

Appendix C.4. Knowledge Discovering

Finding concepts: We selected six MAG generated clusters from ResNet-18 and VGG16. We visualized each cluster through Ms-IV applied to 16 images (8 cats and 8 dogs) from the top-middle ranking positions. From a ranking of 512 images, we started at position 100 to avoid sparsity in higher and lower positions (possible outliers). We presented the Ms-IV visualizations of these image subsets to the research participants and asked which animal part corresponded to the lighter regions in dogs and cats. As we limited the analysis to six clusters per network, there were a total of 12 image subsets.

The 12 obtained subsets and answers are presented in Figures C.15, C.16 and C.17 for ResNet-18 and in Figures C.18, C.19 and C.20 for VGG16.

In general, out of the 13 proposed concepts, fewer than three of them received most of the participants' votes for each cluster. There was agreement about concepts for both computer and non-computer experts. Concepts such as **eyes** and **muzzle** were the most observed.

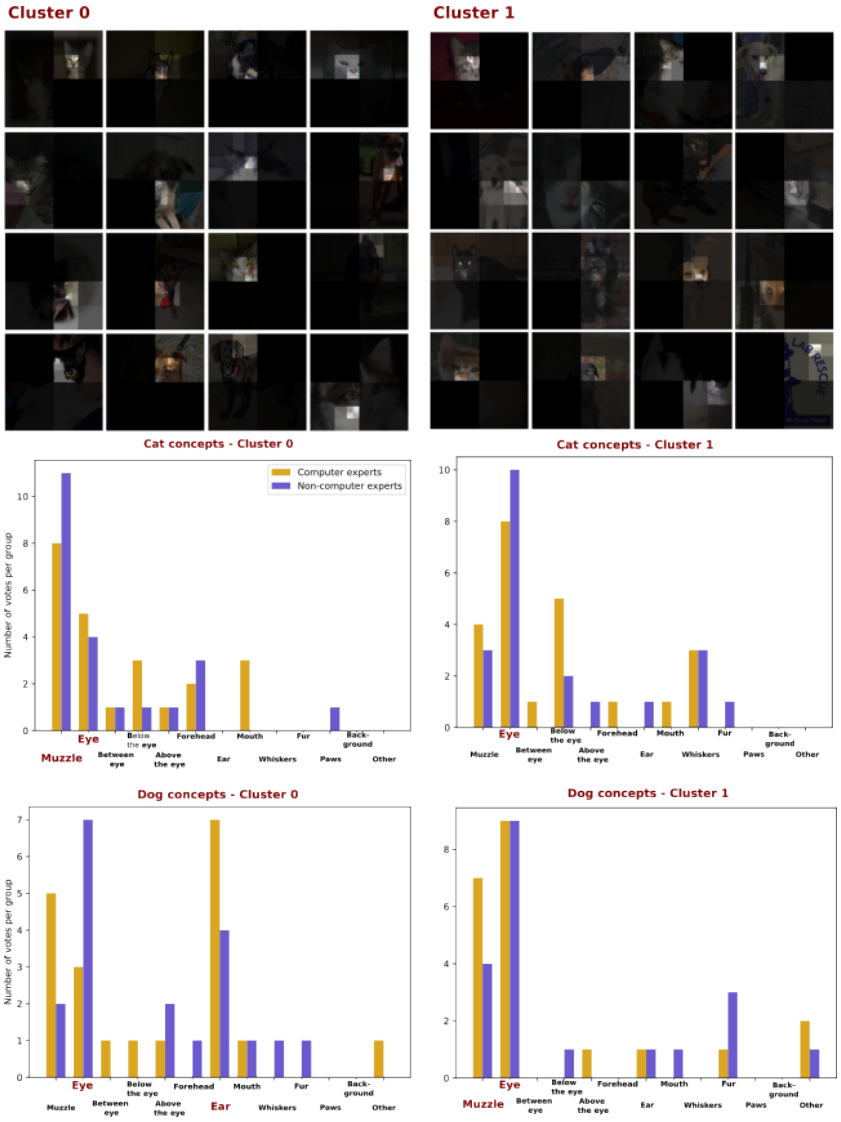


Figure C.15: Visualizations obtained for clusters 0 and 1 of ResNet-18 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 0 presents the **eye** and **muzzle** of cats, while highlighting the **eye** and **ear** of dogs. Cluster 1 presents the **eye** for both classes.

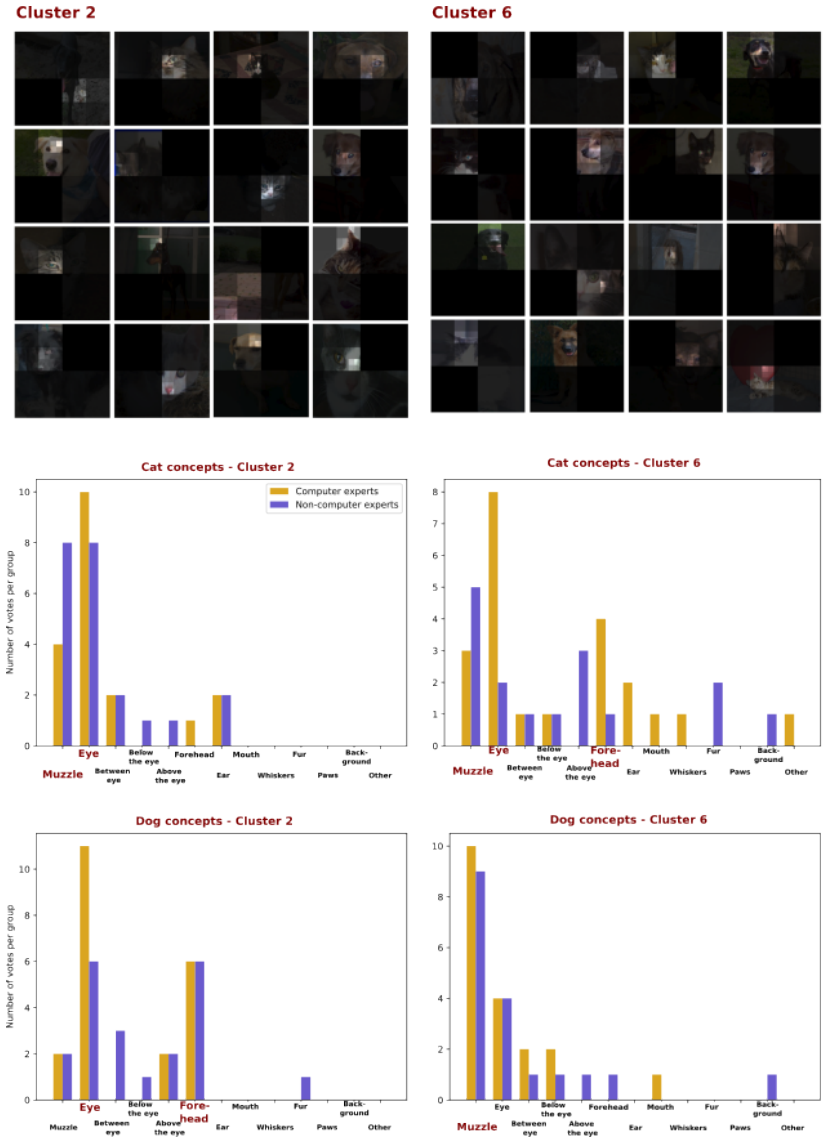


Figure C.16: Visualizations obtained for clusters 2 and 6 of ResNet-18 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 2 presents the **eye** and **muzzle** of cats, while highlighting the **eye** and **forehead** of dogs. Cluster 6 presents the **muzzle** for dogs and a mix of concepts, **eye**, **muzzle** and **forehead**, for cats.

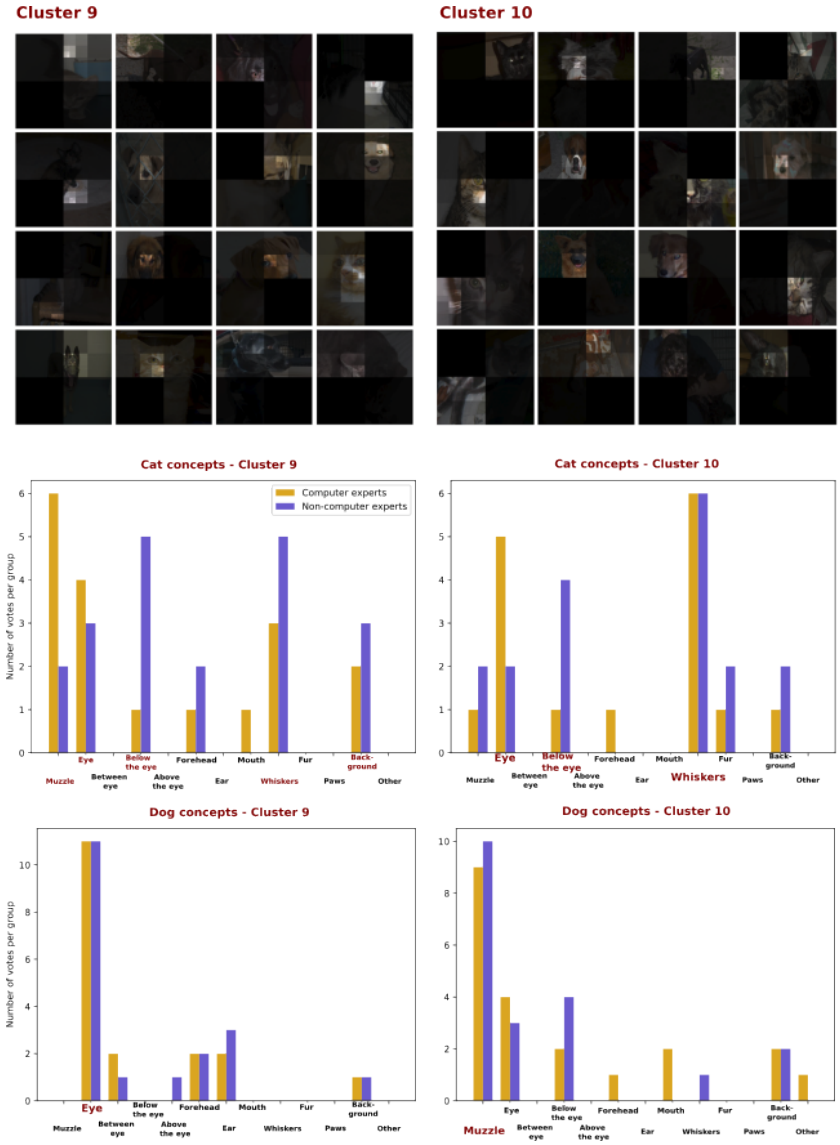


Figure C.17: Visualizations obtained for clusters 9 and 10 of ResNet-18 and the results of selected concepts were described separately by 24 participants. According to the answers, cluster 9 seems not to be well-formed for the cat, but highlights the dog's **eye**. Cluster 10 presents the **muzzle** for dogs and the **eyes**, **below the eyes** and **whiskers** for cats.

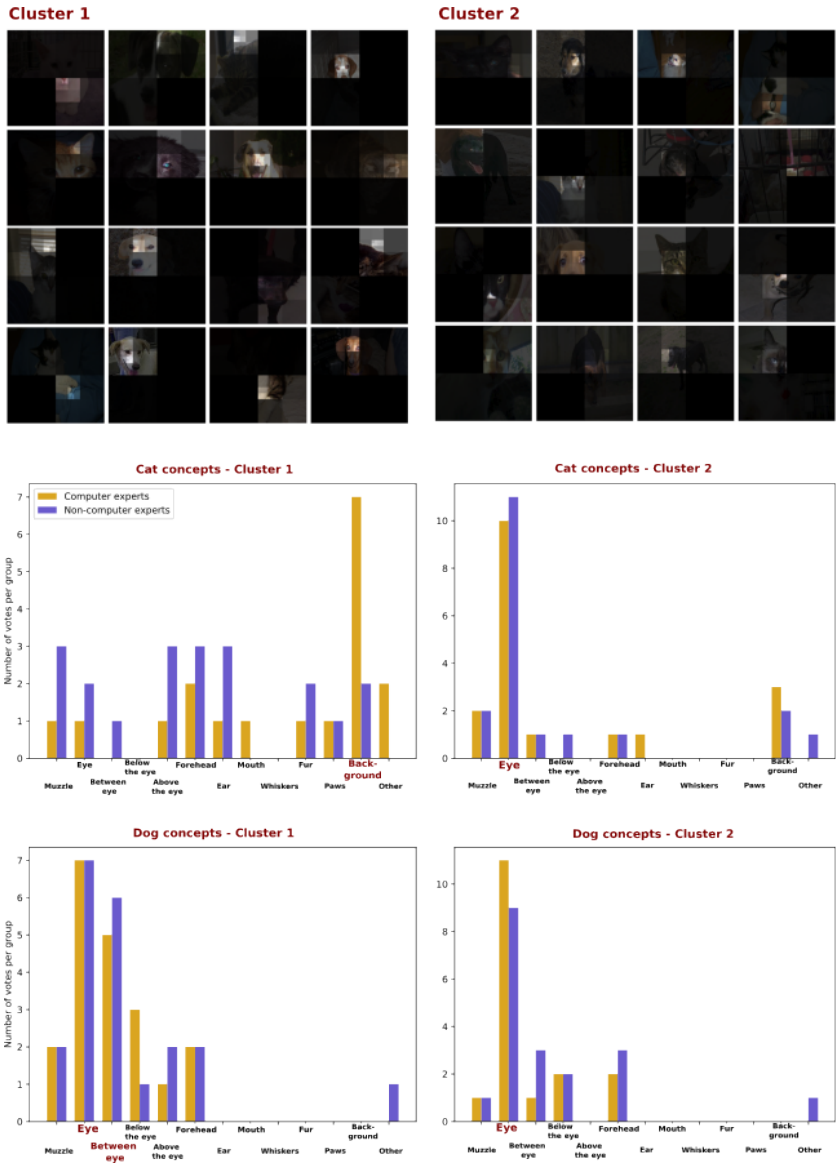


Figure C.18: Visualizations obtained for clusters 1 and 2 of VGG16 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 1 seems not to detect cats well, highlighting the **background**, but highlights the dogs' **eyes** and the area **between eyes**. Cluster 2 presents the **eyes** for both animals.

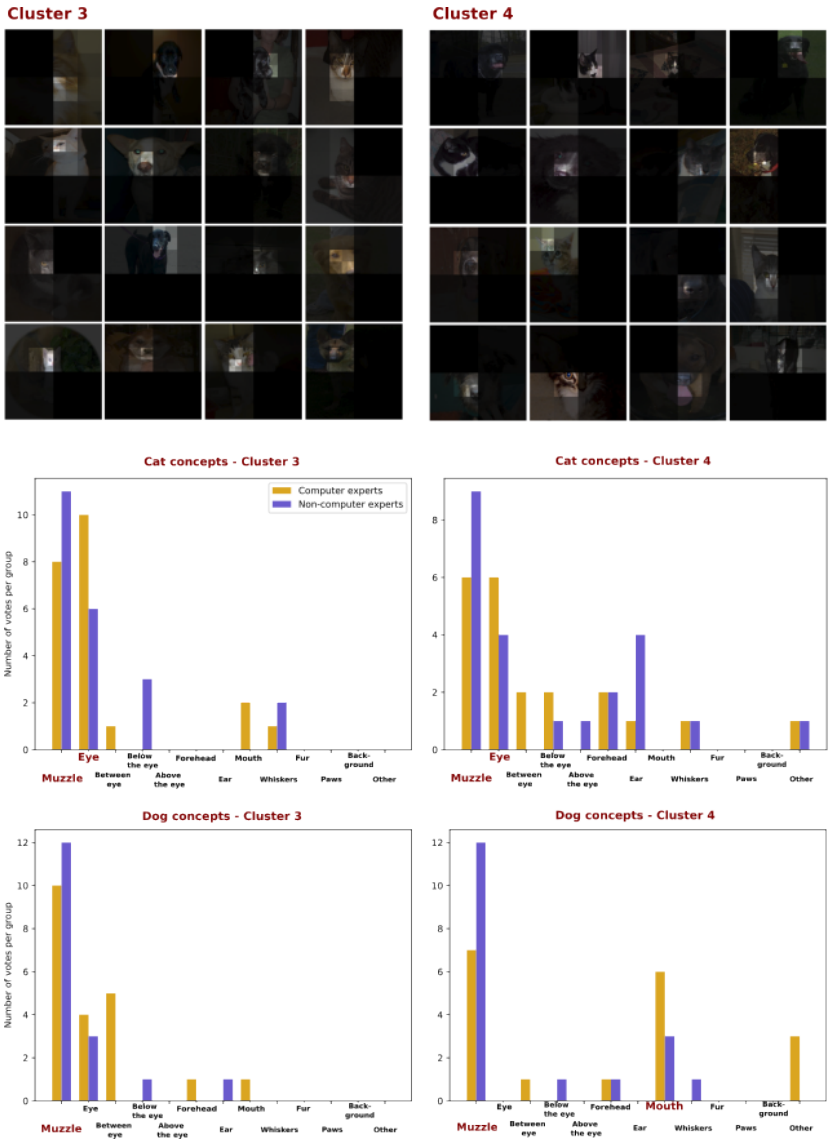


Figure C.19: Visualizations obtained for clusters 3 and 4 of VGG16 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 3 seems not to detect the **muzzle** for both animals and the **eye** for cats. Cluster 4 presents also the **muzzle** and **eye** for cats, but the **muzzle** and *mouth* for dogs.

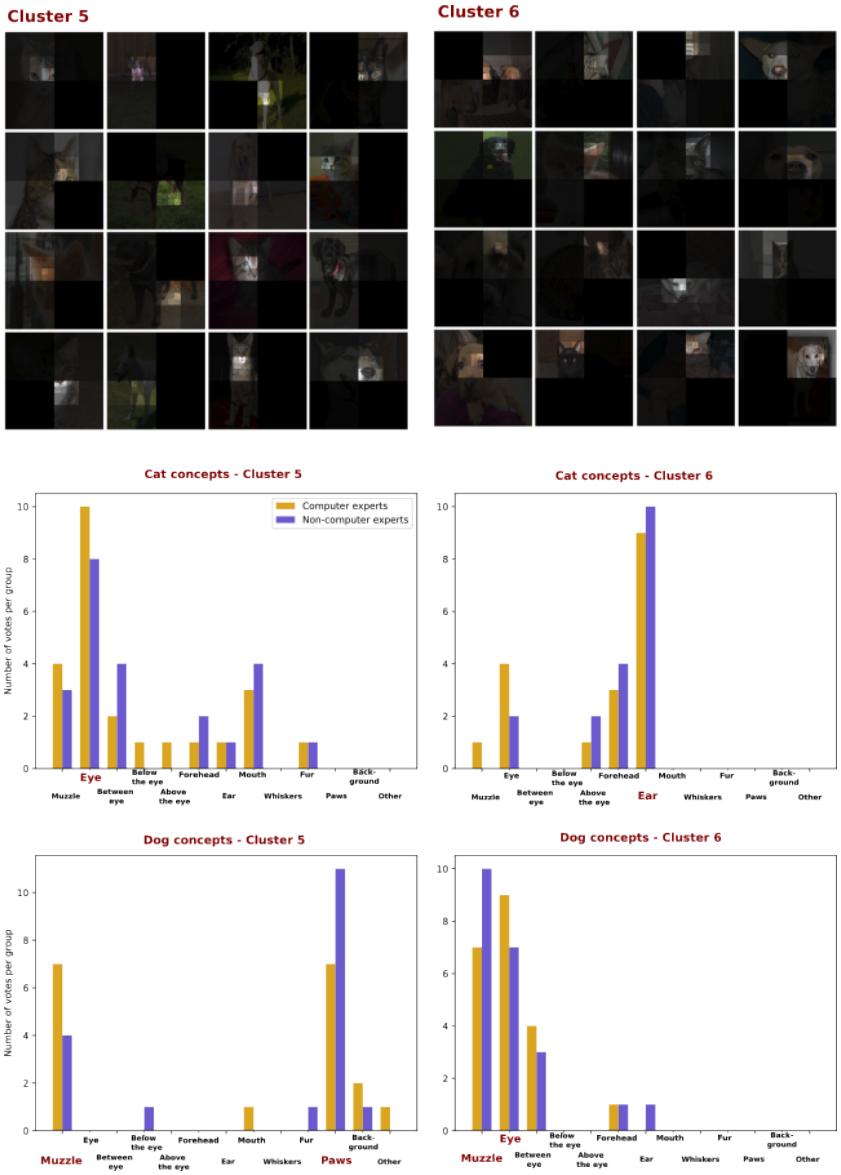


Figure C.20: Visualizations obtained for clusters 5 and 6 of VGG16 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 5 seems not to detect the **eye** for cats and the **muzzle** and **paws** for dogs. Cluster 6 presents the *ear* of cats and the *muzzle* and *eye* for dogs.