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 demonstrate 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:
concepts decomposition, coordinate-based representation, ranking-based global view, causal visualizations

1. Introduction

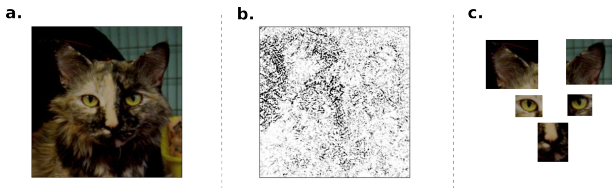


Figure 1: Pixel-level importance is generally more difficult to interpret than components one. From left to right: an image, its Integrated gradients [44] and an easier-to-interpret visualization.

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). It is pixel-level (high-resolution) importance annotated. Nevertheless, we do not directly understand the model's knowledge. It is not easily *interpretable*. *Interpretability* relates to how easy it is, for Humans, to understand something. Often, explanation methods translate the models' outputs into input (or modules) contributions, as a mapping process. The interpretation of ML models involves how the explanations, or the machine knowledge's representation, will impact the humans' comprehension and understanding [4; 12]. Besides some interchangeability between

the terms *interpretable* and *explainable*, *interpretability* is more subjective, as it involves semantics and the idea of how humans understand signals [1; 14]. We can consider the process of interpretation as a *translation of knowledge* that depends not only on the knowledge (or the information semantics), but also on how it is transmitted and received [40]. Consider the metaphor proposed by Saussure [9], one famous researcher from the Semiotics field, about the value of a coin. There are two sides of it. One is the *information* (or sign) that could be a number. The other is dependent on the comparison to the same value in other currencies and, to what it can buy. In other words, it depends on how the information is *interpreted* according to social constructions. In ML, this could mean that interpretation not only depends on the models' knowledge, but also depends on the comparison of this knowledge to the humans' knowledge. That rises the need to approximate explanations to the human reasoning. Same as other research domains, as neuroscience, that inspired the beginning of Artificial Neural Networks through the proposition of the Perceptron [36], we use the human interpretation of knowledge to inspire our models' interpretation method.

Let us establish a human/machine's knowledge analogy. If we give a classification task to a human being, such as a "cat versus dog" image decision, this human probably remembers what makes a cat to be a cat, and a dog, a dog; (s)he will compare what (s)he knows, and what is in the image. The comparison is not pixel-based, it is more at a conceptual level. In other words, the human searches for the muzzle, whiskers, eyes, ears,... and matches those *concepts* with what (s)he knows. When we pass this decision to the network, even with the mechanisms of neurons, synapses and

more recently, attention ones, that try to imitate parts of human reasoning, the network needs to understand the image content, starting from its pixels. Some researches [5; 27] argue that high-level semantics are learned in deeper layers. Moreover, there is a general belief that, in fact, the network learns a certain level of abstraction, that, perhaps, can be translated, and understood by humans as *concepts*. As the network reasoning is complex, one idea is to try to understand the network's *concepts* by decomposition. However, this may not be straightforward, as there are numerous internal units to be analysed. Based on the already mentioned works about high-level semantics in deeper layers, we can limit the analysis to one of the later layers; however, it is still difficult to analyse, considering the number of units. Based on some pruning works [19; 30], we know that there is a certain level of redundancy among the network's units and/or the units' response (such as feature maps and activations). While those responses are different, some present a high similarity, which could indicate the same learned *concept*.

To enable interpretability, we firstly propose to create *concept*'s groups, with similar reasoning units, in order to analyse them individually. Subsequently, we propose a method to visualize each group's *concept*. In the first task, grouping units, we rely on the correlation learned by the network. We want to retrieve, in the same group, similar patterns, visual in the case of CNNs. We represent each unit as its most activated patterns, and, at the end, we group similar activated pattern units.Nevertheless, these most activated group patterns do not necessarily include causal discovery. Causality provides more interpretability [17; 35]. Let us use the example of Yu *et al.* [48]: we consider a model predicting *lung cancer* and *not lung can-*

cer, based on some individuals' attributes. One of these attributes is *yellow fingers*, generally found on smokers. A feature analysis based on correlation could indicate *yellow fingers* as important for deciding *lung cancer*. However, the attribute *smoking* is possibly a better cause for *lung cancer* than *yellow fingers*. We want to discover important *concepts* from our groups based on causal relations.Inspired by techniques such as Occlusion and Sensitivity analysis [50], we use the idea of verifying the impact of occluded structures in the final prediction. We aim to find causality, but with more spatial awareness: we evaluate the changes in the models' output space organization when we disturb the data. For this, we propose a ranking-based metric to analyse the models' output space changes under samples' perturbations.Finally, after causality analysis based on data perturbations and spatial awareness, we are able to visualize what kind of patterns are more important for a *concept*. This is the final tool to discover what is the meaning of each found model's *concept*.

In this paper, we present our three main contributions: **1.** A method, Maximum Activation Groups Extraction (MAGE), that groups model "concepts" represented by feature-map activation patterns; **2.** The Class-aware Order Correlation ($\mathcal{CAOC}$) metric, to determine the correlation between two models *spatial organizations*, and; **3** A Multiscale Interpretable Visualization (Ms-IV), to provide more interpretable visualizations.

We provide: a summary of state-of-the-art methods and interpretability in Section 2; the intuition of our proposed methods in Section 3; the first method MAGE used to mine model's *concepts* in Section 4; the second method Ms-IV to hierarchically visualize concepts using the metric $\mathcal{CAOC}$ in

Section 5; qualitative and human-based experiments to knowledge and bias discovery in Section 6, and; conclusion in Section 7.

2. State-of-the-Art

xAI methods can be broadly categorized as *intrinsic*, *model-specific*, or *post-hoc* and *model-agnostic*. Examples of *intrinsic* methods are decision trees [34], some attention networks [13] and joint training with text explanation [32]. 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. Model *specific-methods* can also cover *intrinsic* models. However, it refers to explanations specifically applied to some determined architectures. For example, Deconvolution [50], CAM [53] and Grad-CAM [38] 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. *Post-hoc* methods can be divided into the following categories: influence analysis, example-based methods and surrogate models [1].

Influence methods: They explain the learning method through influences of inputs or internal components to the output [1]. Examples are Sensitivity Analysis (SA) with salience maps [41] or occlusion techniques [29], [50]; Layer-wise Relevance Propagation [3]; and feature importance methods including Shapley values [26] and gradient-based methods, such as Guided-Backpropagation [43], Integrated Gradients [44] and Grad-CAM [38]. The negative point of these approaches is the difficulty to interpret. Except for Grad-CAM [38], the visualizations are pixel-wise, which can be complex to

understand because of factors such as noisy influences.

Example-based models and Concept Activation Methods: An example-based model searches for examples that well describe the learning method. It can be through an activation maximization [10] by synthesizing a sample which provides the method maximized response; prototypes and criticisms [6; 20] showing examples that are representative and non-representative for the analyzed data; counterfactual examples [47] by searching for minimal changes that would highly modify the learning method response; Concept Activation Vectors (CAVs) [21] presenting human-friendly concepts to explain decisions (detailed below). These methods are helpful to improve interpretability, however, except for CAVs, they propose a limited view (based on examples) of what would be important for the model, that will depend on the human's interpretation of these examples.

Surrogate models: These models are generally linear and easily interpretable, and are used to explain a more complex model decision. The idea is to train these surrogate models based on the different responses to the original model. It can be based on the model decision as LIME [35] or based on the internal activations as TreeView [45]. These methods are better described in the sequel.

2.1. Model-agnostic methods and interpretability

There is also another category named *model-agnostic* methods, that are generally *post-hoc* methods, and can be used to explain multiple types of architectures. Some examples of *post-hoc* and *model-agnostic* methods are Layer-wise Relevance Propagation [3] and Integrated Gradients [44], but also LIME [35] and its numerous derivatives [35; 49; 54; 46; 23; 39], TreeView [45],

and Explanatory graphs [51]. We describe some methods with higher level of interpretability and their differences.

LIME [35]: is a *model-agnostic* methods 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 expressing 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, specially in high-dimensional data. However, even presenting high interpretability, these explanations are generally local, *i.e.*, they are sample-based or rely on local explanations to explain the model behaviour.

Derivatives of LIME: SP-LIME [35] helps to explain the model behaviour by explaining individual representative samples. However, LIME presents instabilities, with multiple different possible explanations for the same sample (due to the random perturbations). Some LIME adaptations try to solve the instability problem: D-LIME [49] presenting a deterministic choice of perturbations; S-LIME [54] determines the sufficient number of perturbation samples to provide stability; OptiLIME [46], leverages the trade-off between stability and adherence (how well the linear plane corresponds to the ML surface) to choose the linear model kernel for the approximation tackling the problem of instability and inconsistency, and; G-LIME [23] using a Bayesian estimator based on sparse and informative global priors. In addition, ALIME [39], improving the fidelity from the approximated model's output space to the original one through a denoising autoencoder to predict

the proximity between the original and the perturbed samples.

TreeView [45]: the method focus on how the model responds to some important attributes of dataset such as cement, sky, grass for a brick-face class decision. For this, the method cluster internal activations according to these attributes to form factors, generate different clusters of data according to each factor and, train a decision tree using these clusters per factor. The TreeView method is the composition of all those factors.

LIME vs. TreeView: LIME and TreeView both belong to surrogate models categories: approximating the learning method by another learning method, simpler and explicable. However, while LIME explains the decision based on *input components*, TreeView explains it based on *internal activations divided into factors*. These factors, however, are pre-defined by a human dataset analysis. Both need supervision.

Explanatory Graphs: proposed by Zhang et al. [51], represents a CNN knowledge hierarchy through convolutional layers. Each node in the graph represents candidate patterns of the object's parts, summarizing 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 optimize the graph of hierarchically connected patterns that best fit network feature maps. There are some other methods using explanatory graphs by mining association rules as [31] or using attention mechanisms [11]. They also improve interpretability for bias discovering and medical applications [22; 8].

Testing with Concept Activation Vectors (TCAV): proposed by Kim et al. [21], aims to, from a set of low-level features, to provide human-friendly interpretable concepts. More in details, CAVs' method is part of TCAV, which analyses how sensitive a model's prediction is to a user 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. At the end, TCAV shows most similar images to a concept. Similar works quantify concepts sensitivity [33], increase performance and model fidelity [52], and apply TCAVs in multimodal emotion recognition [2] and skin lesion classification [25].

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.

3. Intuition

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.

3.1. Concepts' decomposition

We define *concepts* as combinations of features which form a *semantic meaning*. This generally implies a *spatial proximity*. In the context of 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, we can consider these features as pixels. For the human visual system, the process of grouping pixels into concepts seems automatic, but it is not the same for the machines. In the case of artificial neural networks, the patterns to 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 an easier analysis.

3.2. 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 the verification, facial characteristics such as eyes, nose, and mouth will be more important. These characteristics are hierarchically linked to the face (inside the face) and during this task, we give a gradual level of focus, from the face to its contained specific important regions. 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, *cause* 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 relations we have changed in this group. In this way, if an image has the concept A, the image occlusion of this concept will change the relation of it 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.

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

The network's knowledge decomposition is decomposed in five steps, presented in Figure 2.

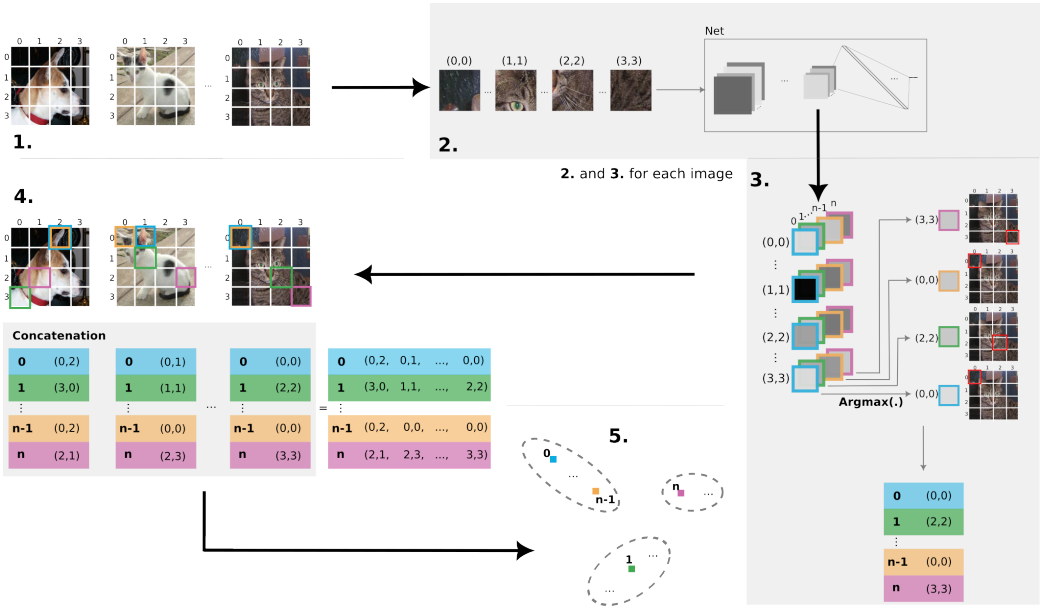


Figure 2: Steps to obtain the *Maximum Activation Groups* (MAGs). We divide dataset images into patches (**1.**). We obtain feature maps (from the last convolutional layer) for each patch (**2.**). We find the corresponding patch with the highest feature map norm by dimension (**3.**). We concatenate 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.**).

4.1. 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 [42] or a ResNet [15], that we model using the formula $\Xi = \Xi^{classif} \circ \Xi^{enc}$ (with $\Xi^{classif}$ the classifier following the encoder denoted by Ξ^{enc}). Then, for a given image $\mathcal{I}$, we decompose the image into patches. According to us, this decomposition into patches is one of the key to be able to understand a little more how the network "reason"; it is our way to tackle a little bit more the *black-box effect* of CNN's 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 of $NbIm$ images $\mathcal{I}_i$ and their class GT_i (the class number). For a given $i \in [1, NbIm]$, we denote by $\mathcal{I}_i$ the i^{th} image of $\mathcal{DS}$. We denote by $NbIm(c)$ 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 by $\mathcal{I}^c_{i_{local}}$ 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

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

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

4.2. Calculus of feature maps' activation per patch

We get feature maps' activations by given each individual patch to the network — the response to the stimuli (see Figure 2 **(2)**). 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 it to the patch $\mathcal{P}(\ell_x, \ell_y, s_p)$, we propose to introduce the term $\mathcal{F}_{i,n_f,(\ell_x,\ell_y),s_p} : (x,y) \to \mathbb{R}$ representing the restriction to the citepd patch of this 2D feature map. Now that we have introduced the formalism to represent features patches, let us show how we decompose the "knowledge" of the encoder into *concepts*.

4.3. Identifying important patches for feature map's dimensions

We want to group feature map's dimensions according to their activation patterns. Therefore, we characterize these dimensions by their patterns. This characterization 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 the emotions and the reading, we give the patches to the network. Therefore, we find the patches that most activate a feature map's dimension to represent this dimension. The characterization is made by the identification of the chosen patch, in this paper, *it is 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 maximizes the 1-norm. It is then identified by its parameters:

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

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

4.4. Dimension characterization by the dataset

We have limited (local) information if we use only one image to characterize dimensions. Therefore, we incorporate more images in the process. Instead of having the feature map's dimensions characterized by only one image's patches, we repeat the process with more images to obtain multiple characterizations. 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:

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

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 representatives will be located near to each other, and conversely.

4.5. Decomposition of feature space into concepts

We use the ensemble of characterizations 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):

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

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

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

After we have the specific *concepts*, we follow the inverse path: we look to what a *concept* represents in each image. We want to visualize image regions with more impact for the concept $MAG(k)$ in the model decision, relying on **human understandability**, **causality**, and **globalism**. For sake of simplicity, let us introduce a new term: we define as *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 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 $\mathcal{I}_j^c$ to visualize, and a concept k. The global causality-based visualization is made in five steps, presented in Figure 3.

5.1. 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 features activations

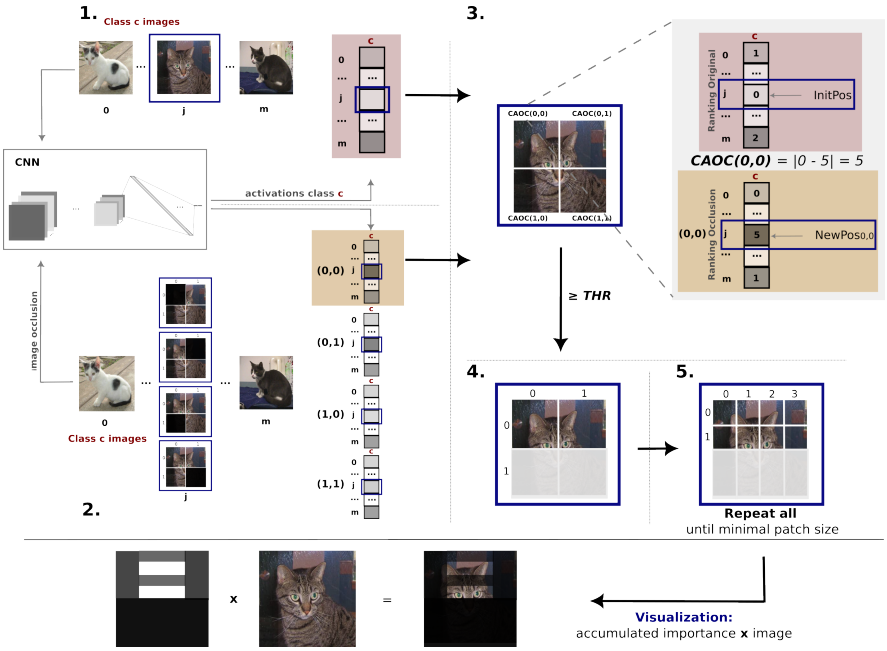


Figure 3: *Multiscale-Interpretable Visualization* (Ms-IV). We obtain the final responses of the network (outputs before Softmax) for a set of image (**1.**). We occlude one patch (out of 4) from the image we want to visualize, and we calculate the new model's response (**2.**). We apply **Argmax** to the images' 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 we 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, at the end, multiply by the original image and obtain the visualization (**5.**).

not relative to concept k in Ξ, leading to a "new" neural network Ξ_k. The other features 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**:

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

5.2. Concept output space under input occlusion

Our causal-based visualization uses patch occlusions to find impactful image regions. Therefore, based on a patch size, we also construct the image corresponding **concept output spaces** with each individually occluded patch. We divide the image $\mathcal{I}^c_j, j \in [1, NbIm(c)]$ 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 the occlusion of each patch $\mathcal{P}(\ell_x, \ell_y, s_p)$ individually, leading to a partially occluded image $\blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}^c_j)$, that will replace the original image $\mathcal{I}^c_j$ in the original sequence $\left(\mathcal{I}^c_{i_{local}}\right)_{i_{local}\in[1,NbIm(c)]}$. Let us define $Occ_{\mathcal{I}^c_j}(\mathcal{I}^c_{i_{local}}) := \blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}_{i_{local}})$ when $i_{local} = j$ and $\mathcal{I}^c_{i_{local}}$ otherwise. Therefore, as presented in Figure 3 **(2)**, the matrix representing the under-occlusion **concept output space**, of image $\mathcal{I}^c_j$ and patch $\mathcal{P}(\ell_x, \ell_y, s_p)$, is: $OS_{\mathcal{P}(\ell_x,\ell_y,s_p),Activ}(c,k) = \left(Activ(Occ_{\mathcal{I}^c_j}(\mathcal{I}^c_{i_{local}})); \Xi_k)\right)_{i_{local}\in[1,NbIm(c)]}$

5.3. Measuring patch importance

As we have the original and each occluded-patch **concept output spaces** 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 construction 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 rankings difference to have an importance score. To measure the impact 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 class scores' positions, from highest to lowest: $Seq_c = argsort\,(OS_{Activ}(c,k)_c, decreasing)$ and $Seq'_c = argsort\left(OS_{\mathcal{P}(\ell_x,\ell_y,s_p),Activ}(c,k)_c, decreasing\right)$. In practice, in the matrices where each row is a data point (a sample's activations), the activation from class c corresponds to the column c. Then we compare the two sequences. As these sequence of positions are rankings, they can be compared by using ranking correlation metrics such as Kendall-tau ($\mathcal{K}$). Calculating this correlation measures how much the patch absence impacts in 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 calculus 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)**).

5.4. Choosing important patches

The higher the score, the more important is the patch. We use a percentage of the most important patch's score to define a threshold. Then, we use this threshold to filter other patches' importance. The visualization is composed only by 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)**).

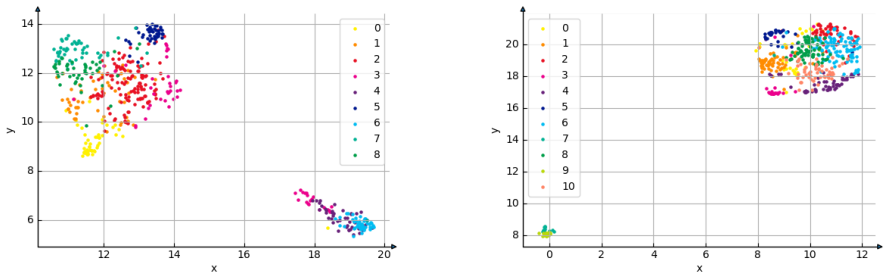
5.5. Reducing patch size to repeat process hierarchically

We perform new occlusions of smaller patch sizes in the sufficiently important patches by repeating steps from 2. We stop reducing patches sizes when we reach a pre-defined smallest size. We compose final patches importances by adding up all patch's 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 patches 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) relatively 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.

6. 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 visual results by showing quantitative evaluation (*Robustness, Faithfulness* and *Localization*). To complete the experimentation, we present a concept's and bias discovery evaluation with humans. The experiments are performed using two CNN architectures, ResNet-18 [15] and VGG16 [42], trained in a classification dataset of cats vs. dogs [1], and CUB-200-2011 [16] dataset for *Localization* evaluation.

6.1. MAGs' scatter plot



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

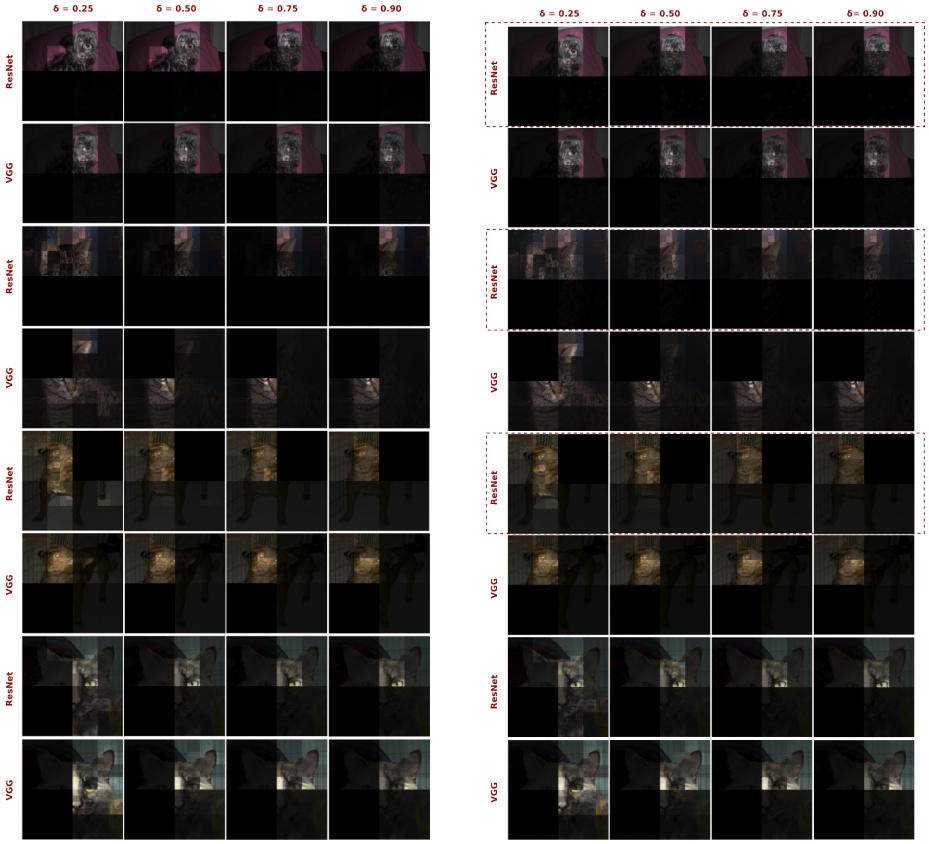
Figure 4: Projection of 5120-dimensional representations to 2D with UMAP. Figures (a) and (b) represent the plots of feature maps' dimensions from VGG16 and ResNet-18 respectively. Colors represent clusters obtained by K-means. We use k equal to 9 for VGG and, 11 for ResNet chosen by the Elbow curve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

We show in Figure 4 the dispersion of obtained clusters (high dimensionality reduced to 2D using UMAP). To generate the representation, we use

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

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".

6.2. Multiscale visualization



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

Figure 5: Ms-IV visualizations using $\mathcal{CAOC}$ metric to measure patches importance for two image samples (a dog and a cat) using VGG16/ResNet-18 models (see explanations in the text).

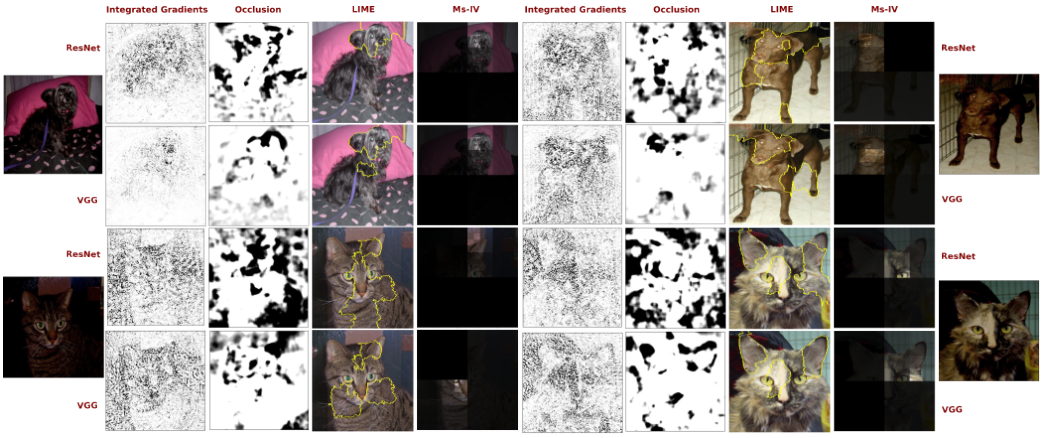


Figure 6: Attribution maps by IG and OC methods versus Ms-IV: IG and OC do not let us recognize the shapes of the dog or the cat, when Ms-IV, enlightening the initial image, lets us easily recognize which part of the image is important in the model's decision.

We present three visualization experiments: i) visualizations based on *thr* in $\{0.25, 0.50, 0.75, 0.90\}$ for changing the acceptance of important patches (calculated by the percentage *thr* of the maximum patch importance); ii) visualizations comparison between $\mathcal{CAOC}$ and *PD*s; and, iii) visualizations comparison of Ms-IV, Integrated Gradients (IG) and Occlusion (OC). In Fig. 5(a) we show results for two values *thr* (0.25 and 0.9) in two image samples and the two architectures, using $\mathcal{CAOC}$ to obtain patches contributions. The final value *thr* (last column) has more concentrated attention (less light regions). ResNet18 network focus more on the animal's eyes than VGG. By changing the metric from $\mathcal{CAOC}$ to *PD*s (using the same visualization multiscale process), we obtain Fig. 5(b). For most visualizations, 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.

6.3. Quantitative evaluation

Papers such as Bommer et al. [7], analyse the use of metrics as *Complexity*, *Robustness*, *Faithfulness* and *Localization*, directed to 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) to 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 be the same (**Preserved Class**).

To this evaluation, we use the *Worst Case Evaluation* proposed by Huang et al. [18]. 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's size of 100 particles and selection of 20 particles for next iterations (reduced numbers due to computational

resources' limitation). We use Pearson's correlation to compare the original and perturbed image explanation. We compare Ms-IV to IG and LIME.

Table 1: *Robustness* and *Faithfulness* analysis. (a) *Robustness* values calculated using Worst Case Evaluation for **Preserved Class** and **Misclassification** for three visualization methods: IG, LIME and Ms-IV. Results using Pearson's correlation between original image and Worst Case visualizations. We expect higher values 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)

The results in Table 1a show high **Preserved class** *robustness* for Ms-IV, closer to the gradient-based method IG (the best for **Preserved class** according to Huang et al. [18]). However, LIME, which is not good for **Preserved class** *robustness*, wins in **Misclassification** *robustness*. As LIME is the closest to our method in terms of easier visual interpretability (Figure 6), we follow with comparisons between LIME and Ms-IV. *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 differ-

ent 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 important region disturbs more the model's output by verifying, for each image, if it was LIME or Ms-IV that presented the biggest variation (absolute output class difference) under occlusion (>). Ms-IV presents a bigger number of output variations with 72% of images for VGG model, and 69% of images for ResNet model, which demonstrates higher *faithfulness* of the chosen important regions to the model. The criterium *Localization* refers to the method capability of, based on a well-trained model, localize the object of interest in the image (of the correct class). For example, in a cat/dog classification problem, if we have a cat image for which the model gives the correct answer, it is expected 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, this idea of *localization* needs to be adapted. We want to evaluate if a MAG (concept cluster) is able to show the

same concept in different images, relying on our Ms-IV method. 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 shown areas as different animal parts (total of 14 different labels). As our focus is to be global-aware, we consider that a good-*Locatization* method, in our context, 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 all 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 labelled animal part within its percentage of apparition in the analysed images (the higher, the better). Subsequently, we show the background percentage in each MAG. MS-IV had the best conventional *localization* rates (highlighting less background regions) and presented higher percentages of same concept for each MAG, specially for the VGG model. To reinforce the comparison, we also perform a localization experiment with an already parts-labeled dataset, CUB-200-2011 [16]. 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 to this problem, obtaining 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 MAG's 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 with the parts coordinates. We consider, as the visualized part, the one with the closest coordinate. If the visualization centroid is outside the bird bounding-box, we consider 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.

6.4. 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 analysed by the 24 individuals selected between computer and non-computer experts from two countries: Brazil and France. They were a total of 11 computer experts, including people from industry and academy. There were a total of 13 non-computer experts including people from non-academical domains, and from the three main domains (in similar quantities): humanities (sociology, geography, architecture), biologics (medicine, physical education) and exact sciences (mathematics) in different educational stages (under graduation, graduation and post-graduation). **Finding concepts:**

Table 2: *Localization* analysis for cat vs. dog and CUB datasets. (a) Evaluation of cat vs. dog dataset considering 5 MAG's concepts (10 images each) and both models VGG and ResNet. Two types of evaluation: Conventional *localization* showing the percent of background considered important (the lower the best), and; concept *localization* considering the quantity of same concept in each cluster (the higher, the better). In both cases, Ms-IV presents better *localization* according to 200 visualizations labelled by 12 individuals. Using CUB dataset and, VGG and ResNet models trained with Warblers vs. Sparrows: (b) percent of background *localization* and the top2 most-found concepts per MAG, and; (c) concepts *localization* for each MAG individually. In average, Ms-IV provide 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. We present the selected concepts for these cluster, 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 *ear* of cats and, the *muzzle* and *eye* for dogs.

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 in position 100 to avoid sparsity in higher and lower positions (possible outliers). We presented the Ms-IV visualizations of these images' subsets to the research participants, and asked which animal part corresponded to the lighter regions in dogs and in cats. There was a total of 12 image subsets (limited analysis to six clusters per network). From the 13 concepts, less than three of them received most of the participants' votes

for each cluster. There was an agreement about concepts, for computer and non-computer experts. Concepts such as **eyes** and **muzzle** were the most observed. We highlight the Fig. 7 as an example of high concordance and more 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 presented the non-bias group choice as a better separation. We show the values for computer (**Comp.**) and non-computer (**Non-comp.**) experts to select 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 images' subsets to represent each *concept* group (as in **finding concepts** part). Then, 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 the concept of Neural Network bias. We present the results in Table 3. Both computer and non-computer experts found, with high accuracy, the non-biased model (73% of

correct guesses for the non-computer experts). Our proposed visualization facilitates model analysis, even for non-specialists.

7. 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 models' knowledge. *CaOC* metric is used to consider the structure and organization of the model's final decision space and provides globally-awareness of samples' 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] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xAI), IEEE Access 6 (2018) 1–23.

[2] A. R. Asokan, N. Kumar, A. V. Ragam, S. Shylaja, Interpretability for multimodal emotion recognition using concept activation vectors, in: 31th International Joint Conference on Neural Networks (IJCNN), IEEE, 2022, pp. 01–08.

[3] 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.

[4] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (xAI): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[5] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1798–1828.

[6] J. Bien, R. Tibshirani, Prototype selection for interpretable classification, Annals of Applied Statistics 5 (4) (2012) 1–23.

[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 abs/2303.00652 (2023).

[8] L. Cui, H. Seo, M. Tabar, F. Ma, S. Wang, D. Lee, Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation, in: 26th International Conference on Knowledge Discovery & Data Mining (SIGKDD), KDD '20, 2020, p. 492–502.

[9] F. de Saussure, Cours de linguistique générale, Language Arts & Disciplines, 1989.

[10] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing higher-layer features of a deep network, Technical Report, Univeristé de Montréal (2009) 1–13.

[11] Y. Geng, J. Chen, Z. Ye, Z. Yuan, W. Zhang, H. Chen, Explainable zero-shot learning via attentive graph convolutional network and knowledge graphs, Semantic Web 12 (5) (2021) 741–765.

[12] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning, in: 5th International Conference on data science and advanced analytics (DSAA), IEEE, 2018, pp. 80–89.

[13] 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.

[14] 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.

[15] 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.

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

[17] J. M. Hofman, A. Sharma, D. J. Watts, Prediction and explanation in social systems, Science 355 (2017) 486 – 488.

[18] 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.

[19] K. Kahatapitiya, R. Rodrigo, Exploiting the redundancy in convolutional filters for parameter reduction, in: 21st IEEE Winter Conference on Applications of Computer Vision (WACV), 2021, pp. 1409–1419.

[20] B. Kim, R. Khanna, O. Koyejo, Examples are not enough, learn to criticize! Criticism for interpretability, in: 30th International Conference on Neural Information Processing Systems (NeurIPS), 2016, pp. 2288–2296.

[21] 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.

[22] C. Y. Li, X. Liang, Z. Hu, E. P. Xing, Knowledge-driven encode, retrieve, paraphrase for medical image report generation, in: 33rd Conference on Artificial Intelligence (AAAI), 2019, pp. 1–10.

[23] 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).

[24] S. P. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (2) (1982) 129–136.

[25] A. Lucieri, M. N. Bajwa, S. A. Braun, M. I. Malik, A. Dengel, SherazAhmed, On interpretability of deep learning based skin lesion classifiers using concept activation vectors, in: 29th International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–10.

[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] A. Mahendran, A. Vedaldi, Visualizing deep convolutional neural networks using natural pre-images, Biometrika 120 (2016) 233–255.

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

[29] L. Merrick, Randomized ablation feature importance (2019).

[30] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning convolutional neural networks for resource efficient inference, in: 5th International Conference on Learning Representations (ICLR), 2017, pp. 1–17.

[31] A. Nikolov, M. d'Aquin, Uncovering semantic bias in neural network models using a knowledge graph, in: 29th International Conference on Information and Knowledge Management (CIKM), 2020, pp. 1175–1184.

[32] 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.

[33] J. Pfau, A. T. Young, J. Wei, M. L. Wei, M. J. Keiser, Robust semantic interpretability: Revisiting concept activation vectors, arXiv (2021).

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

[35] 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.

[36] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain., Psychological Review 65 (6) (1958) 386–408.

[37] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[38] 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.

[39] 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.

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

[41] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: Workshop at 2nd International Conference on Learning Representations (ICLR), 2014, pp. 1–8.

[42] 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.

[43] 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.

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

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

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

[47] S. Wachter, B. D. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, Harvard Journal of Law & Technology 31 (2) (2018) 841–887.

[48] K. Yu, X. Guo, L. Liu, J. Li, H. Wang, Z. Ling, X. Wu, Causality-based feature selection: Methods and evaluations, ACM Comput. Surv. 53 (5) (2020).

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

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

[51] 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.

[52] 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.

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

[54] 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.

Supplementary material

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

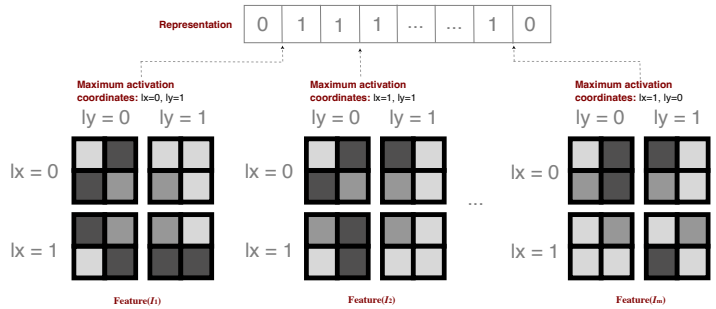


Figure 8: Finding the "position" of a feature f_p (in each image of) the data set (illustrative example). We start from a sequence $(\mathcal{I}_i)_i$ of images whose domain is of size 4×4. We decompose their common domain into 4 patches of same size 2×2. On the first image, we observe that among the 4 four subdivisions in the f_p^{th} feature map, the one which 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 data set. This vector represents then where the f_p^{th} feature is located in the images of the data set; we call is the *representative* of the feature number f_p.

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

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

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

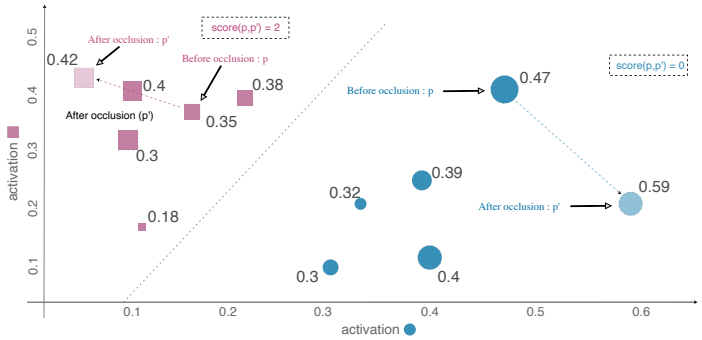


Figure 9: Explanation of the computation of $\mathcal{CAOC}$ (on a fictive scenario). Here, our data set is made of images of one disk or of one square, the output is a class ("disk" or "square"). We have fixed the concept value to some random k. We plot the activations distributions in a 2D space: the 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 move in this space by a distance 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 to use rankings correlation between before/after the occlusion of the most important patch images relatively to the concept k. We obtain that the disk did not move in the disks ranking when we did the occlusion (it remains the "strongest" disk), so the correlation $\mathcal{CAOC}(k, disks)$ is maximal, meaning that concept k does not have much influence on disks. Conversely, in the squares case, the square position changes from 3^{rd} to 5^{th} position, leading to a ranking change of 2, thus $\mathcal{CAOC}(k, squares)$ is lower, which means that concept k is important for squares.

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

We propose here an algorithm (see Algorithm 1 which is depicted in Figure 10) using a multiscale hierarchical approach (like a quad-tree) able to enlighten the areas of a given image $\mathcal{I}$ (belonging to $\mathcal{DS}$) as much as the concerned area is important in the decision of the network relatively to the

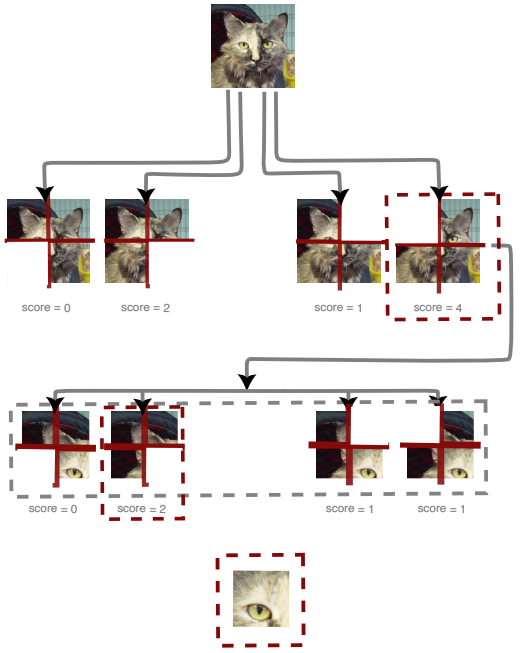


Figure 10: How our visualization algorithm works. We fix some concept valued k and some image number i. We sort the activations of each image and we call *InitPos* the position of the activation of the current image $\mathcal{I}_i$ in the computed sequence. The goal is to enlighten the areas of the image 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 by computing how much their new position $NewPos^{\blacksquare}_{\ell_x,\ell_y}$ (in the sequence of activations of the occluded images) is different from *InitPos*, we obtain 4 scores $\left|InitPos - NewPos^{\blacksquare}_{\ell_x,\ell_y}\right|$ (called the importance) which, by choosing the maximal one, allows us to deduce in which of these four patches the concept is the most represented. 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 enlighten each coordinate in the image (by summing the importances we have computed for each pixel).

concept k. This approach is *global* in the sense that computations on the whole data set will have been done before. Note that our procedure is differ-

ent from LIME: even in both cases, we use parts of images to show model's knowledge, in our case we enlighten the image relatively to one unique concept at a time.

For sake of simplicity, let us introduce a new term: we define as *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 expose formally the main steps of our algorithm (we handle the recursive part of the algorithm thanks list that we do not detail here for 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 $Activ_c(\mathcal{I}^c_{i_{local}}, \Xi_k)$ in it. This position represents "how much" $\mathcal{I}^c_{i_{local}}$ is really of class c.
3. We divide the image into four patches of the same size $s_p = \frac{s_{image}}{2}$ resulting into 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 $\mathcal{P}(\ell_x, \ell_y, s_p)$ leading to $\blacksquare_{\mathcal{P}(\ell_x,\ell_y,s_p)}(\mathcal{I}^c_{i_{local}})$.
   (b) We compute:

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

   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

having occluded the image on the patch domain. If the initial activation is almost preserved despite of the occlusion, we will have $NewPos^{\blacksquare}_{\ell_x,\ell_y} \approx InitPos$.

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

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

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 recursively the procedure in the patches satisfying 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) relatively 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

Methods evaluation

We present the experiments to test MAGE, $\mathcal{CAOC}$ and Ms-IV, using two CNN architectures, ResNet-18 [15] and VGG16 [42], trained in a classification dataset of cats vs. dogs [2].

Dataset and training

Table 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 training set, 5,109 (2,564 dogs and 2,545 cats) images in validation set and, 12,499 in test set. For the training, we used the pre-trained networks in ImageNet dataset, we excluded the networks' original classification layer and we included a 2 neuron layer followed by softmax activation. The networks were trained during 1000 epochs using Cross Entropy loss, Adam optimizer and learning rate equal to $1e-7$. We saved only the model that minimizes validation loss. We present in Table 4 the accuracy values for each model.

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

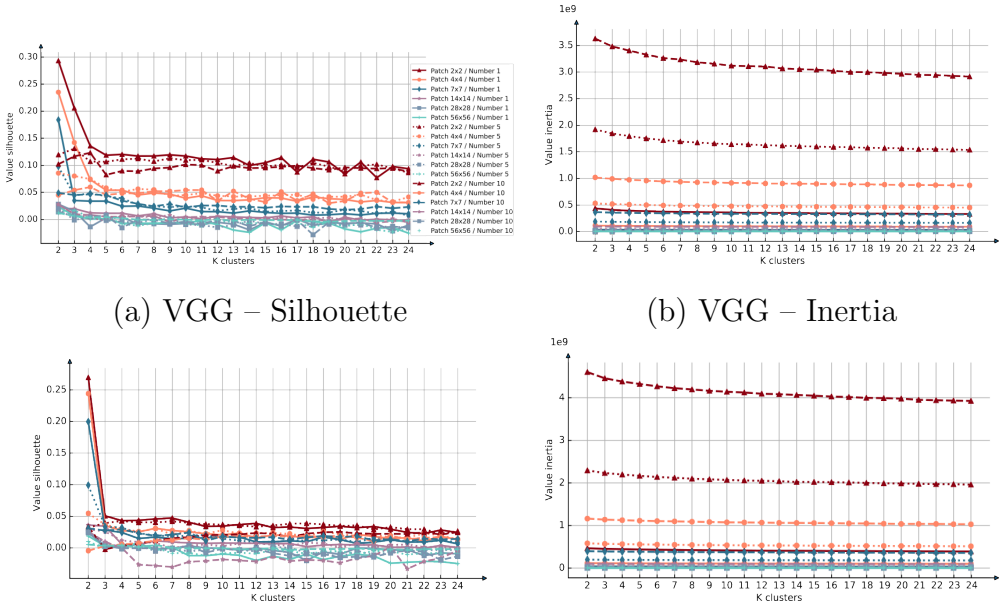
Evaluating the quality of MAGE

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 $1, 5$,ad 10. We use a subset of 512 images, half from each class. For each configuration, we apply the k-means [24] and, we vary the number of clusters $k \in [2, 25[$. We use the metrics Silhouette [37] 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 comparing the feature maps' spatial position for patches size $n \in \{4, 7, 14\}$ in Fig. 12 for VGG16. The scatter plot visualizations show more sparsity intra-clusters for bigger patches' sizes. Note that for visualization purposes, we reduced the representation dimensionality using UMAP technique [28], with the parameter $n\_neighbors = 50$, $min\_dist = 0.0$ and Euclidean distance. Colors determine the different assigned clusters for each of them.

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



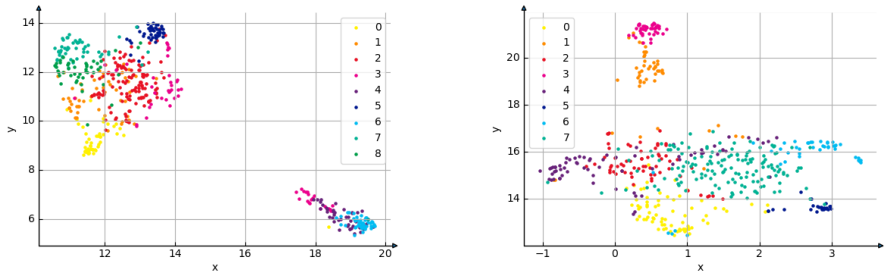
(a) VGG – Silhouette

(b) VGG – Inertia

(c) ResNet – Silhouette

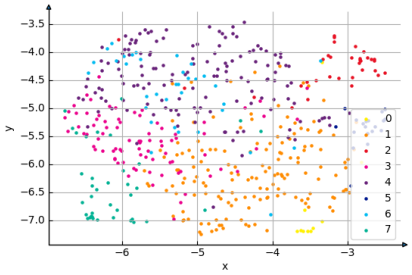
(d) ResNet – Inertia

Figure 11: Silhouette and Inertia results from K-means with $k \in [2, 25[$ with different size and number of patches used in the feature maps 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 patches' size seem to improve this mentioned quality.

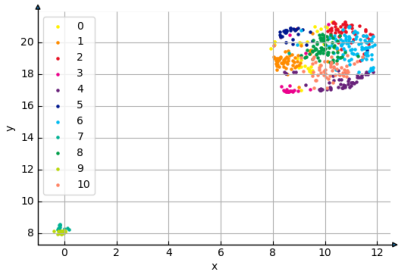


(a) VGG $n = 4$, 9 clusters

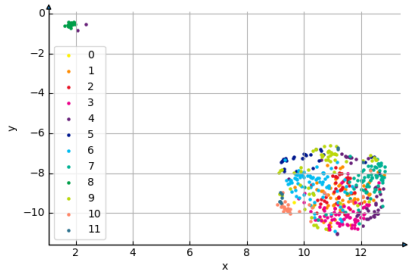
(b) VGG $n = 7$, 8 clusters



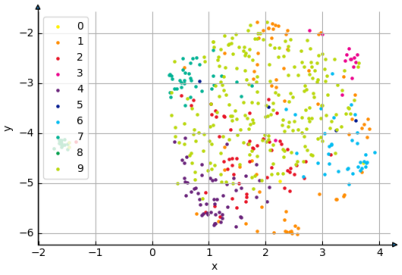
(c) VGG $n = 14$, 8 clusters



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



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

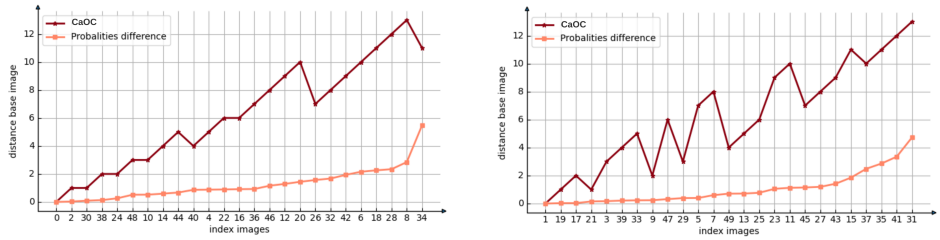


(f) ResNet $n = 14$, 10 clusters

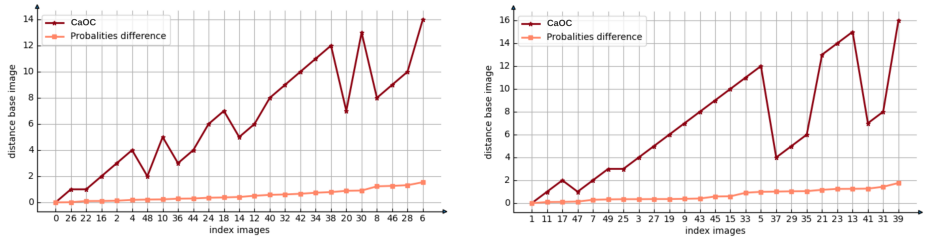
Figure 12: Smallest values of n seem to present denser clusters. Figures (a), (b) and (c) represent the scatter plots of feature maps of VGG16 and Figures (d), (e) and (f) of ResNet according to the representation using $n \in \{4, 7, 14\}$ and $t = 5$, respectively. The colors represent the clusters 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 size of patch) chosen by the Elbow curve method from the Inertia presented in Figure 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 less 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. 12). As the feature map cluster centers were reasonably similar, we continue the experiments with the $n = 4$ clusters (small but not the smallest).

Relation between CaOC and Probabilities



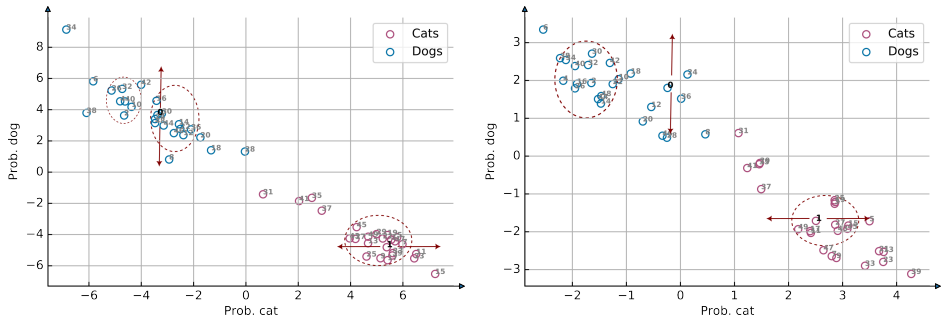
(a) VGG – dog (b) VGG – cat



(c) ResNet – dog (d) ResNet – cat

Figure 13: *CaOC* and *Probabilities difference* behave differently. Based on a dog (index 0) and cat (index 1) image, we calculated the difference to other 24 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 with respect to this order. Even indexes are dogs (figures (a) and (c)) and Odd indexes 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 calculate the difference from images 0 (dog) and 1 (cat) to all the others (from their corresponding classes) using $\mathcal{CAOC}$ and *PD*. We present the results for dog's class in Fig. 13. In this figure, we ordered the images with respect to the distance obtained by *PD*s to ob-



(a) VGG16 50 images

(b) ResNet18 50 images

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

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

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 in position 100 to avoid sparsity in higher and lower positions (possible outliers). We presented the Ms-IV visualizations of these images' subsets to the research participants, and asked which animal part corresponded to the lighter regions in dogs and in cats. As we limited the analysis to six clusters per network, there was a total of 12 image subsets.

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

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

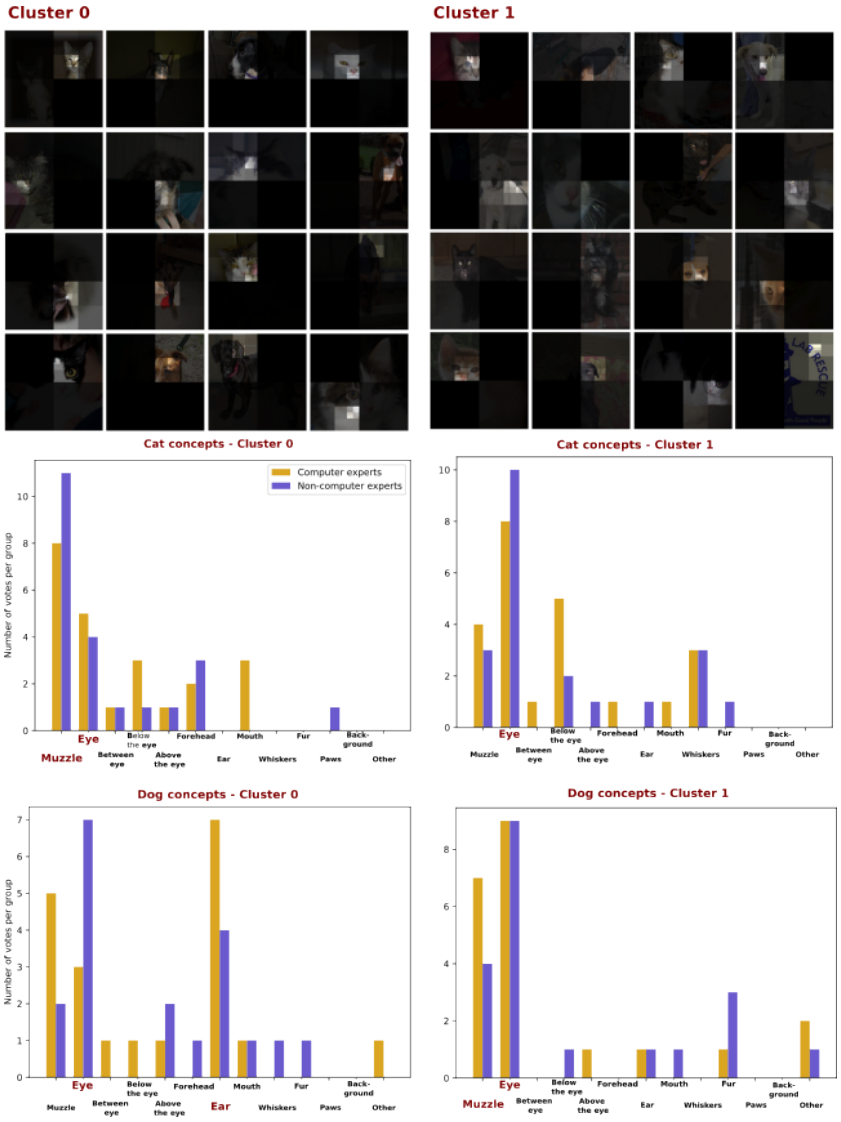


Figure 15: Visualizations obtained for clusters 0 and 1 of ResNet-18 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 0 presents the **eye** and **muzzle** of cats, while highlights the **eye** and **ear** of dogs. Cluster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

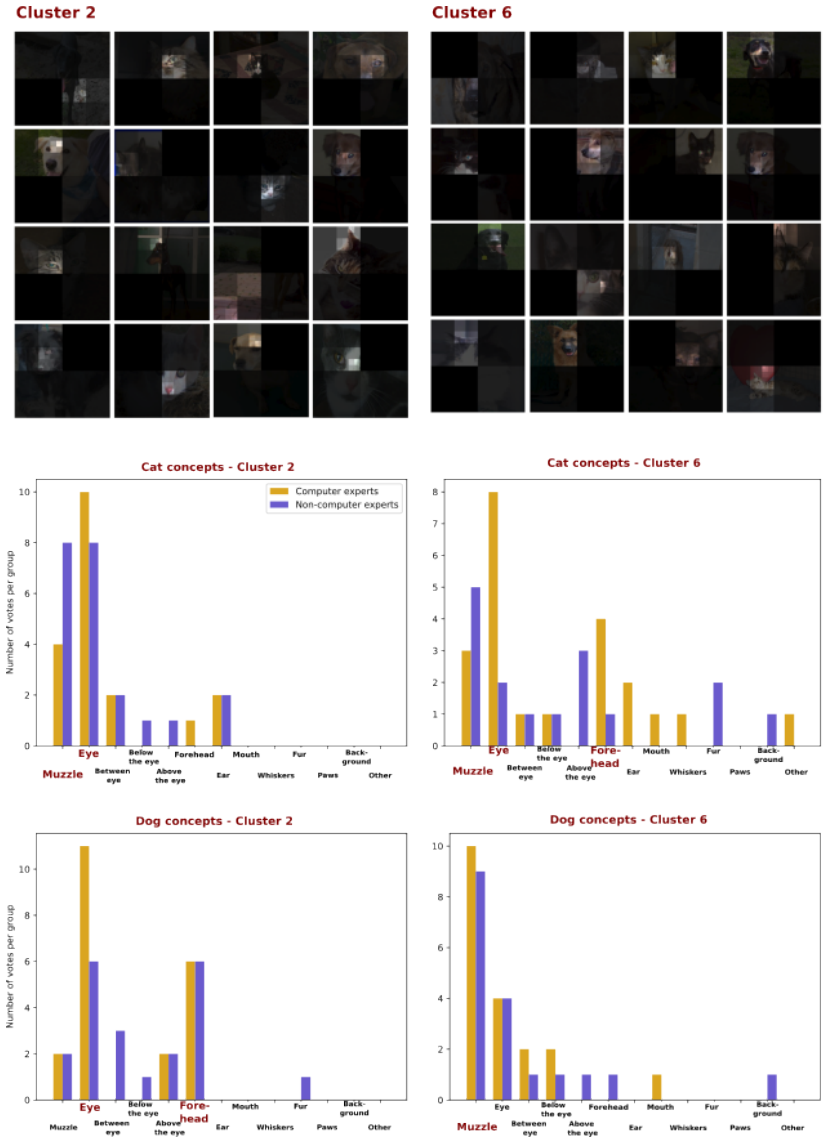


Figure 16: Visualizations obtained for clusters 2 and 6 of ResNet-18 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 2 presents the **eye** and **muzzle** of cats, while highlights the **eye** and **forehead** of dogs. Cluster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

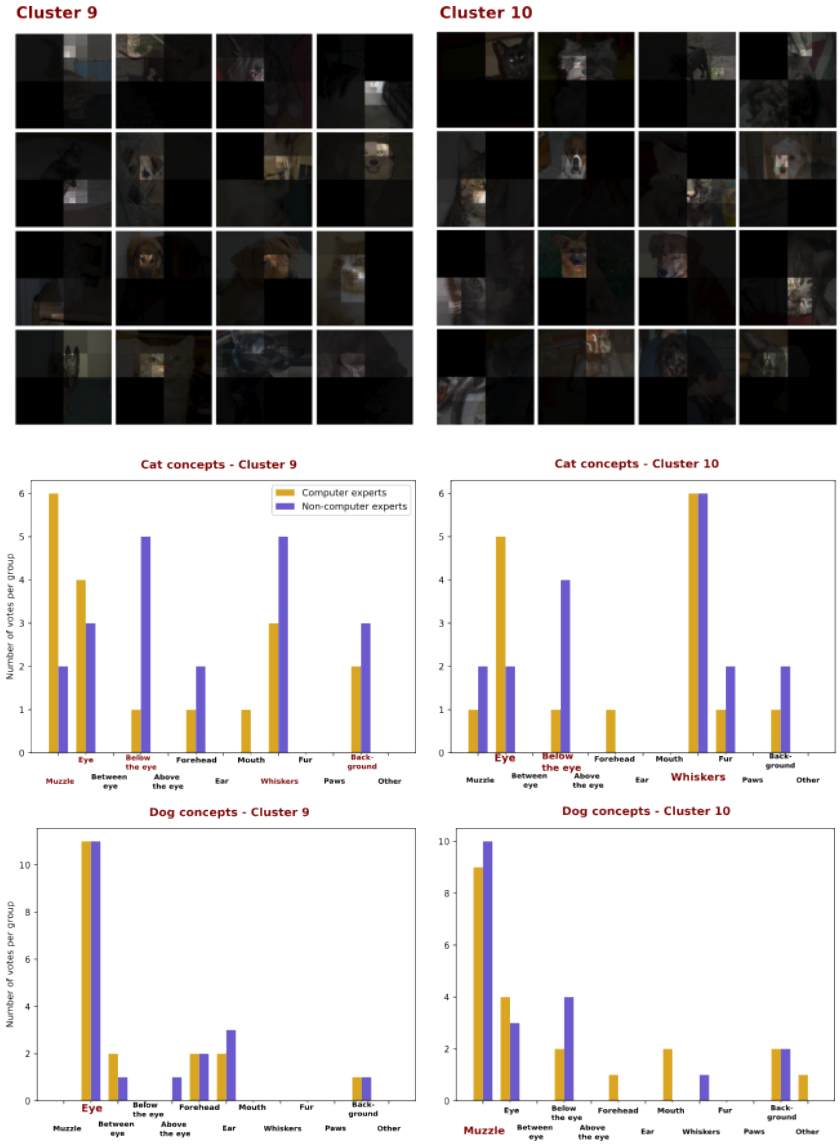


Figure 17: Visualizations obtained for clusters 9 and 10 of ResNet-18 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 9 seems not to be well-formed for the cat, but highlights the dogs' **eye**. Cluster 10 presents the **muzzle**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

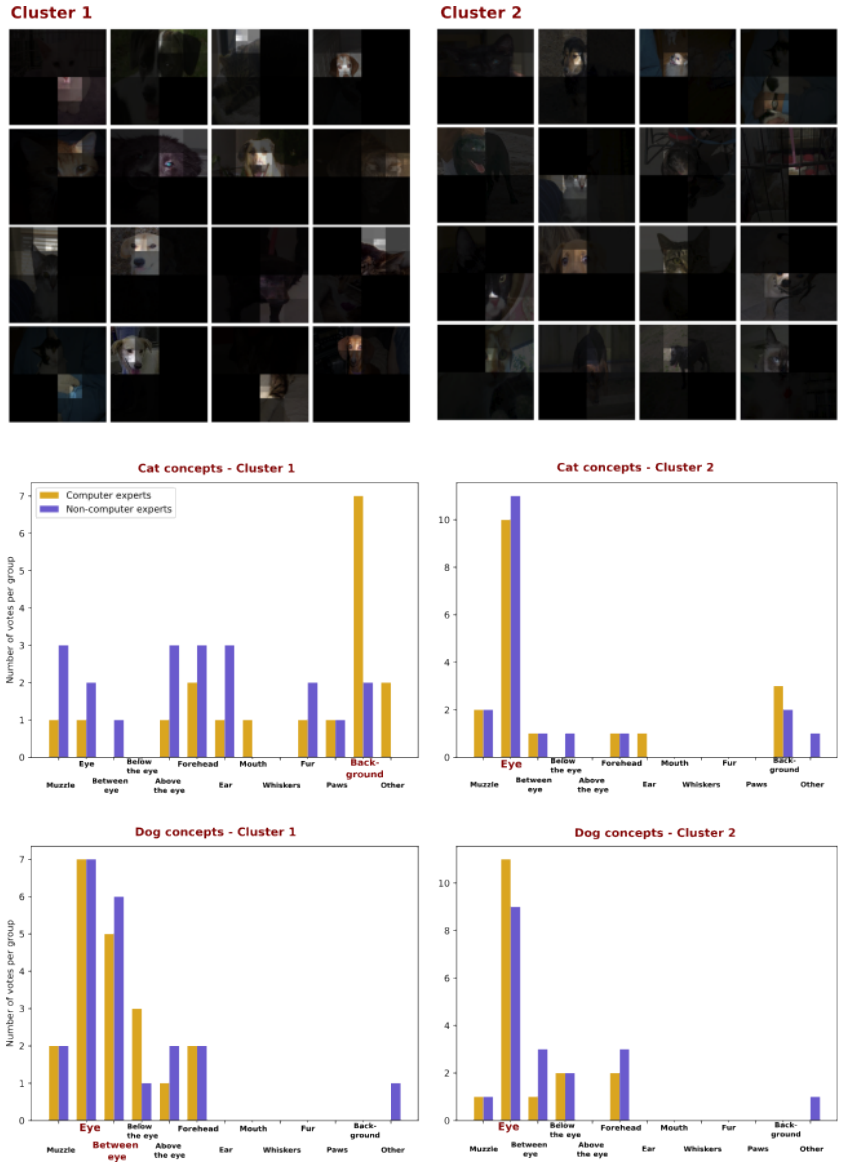


Figure 18: Visualizations obtained for clusters 1 and 2 of VGG16 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 1 seems not to detect cats well, highlighting the **background**, but highlights the dogs' **eye** and **between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

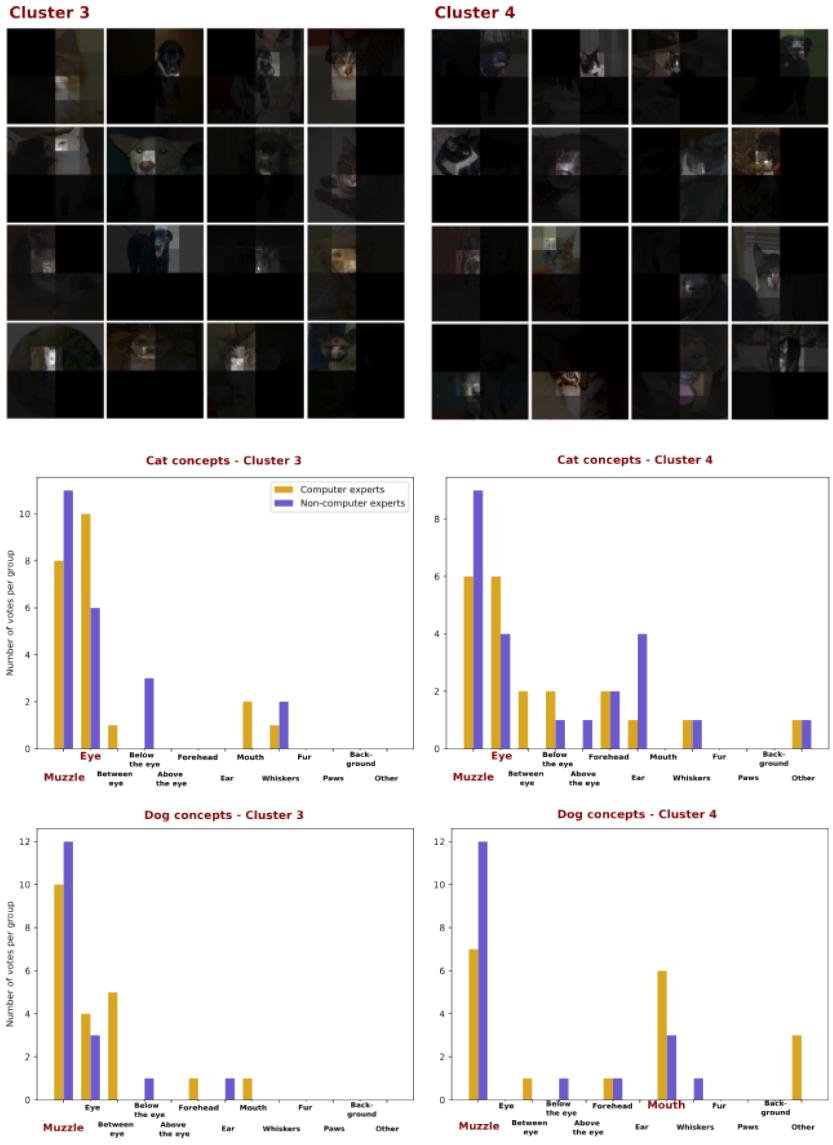


Figure 19: Visualizations obtained for clusters 3 and 4 of VGG16 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 3 seems not to detect the **muzzle** for both animals and the **eye** for cats. Cluster 4 presents also the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

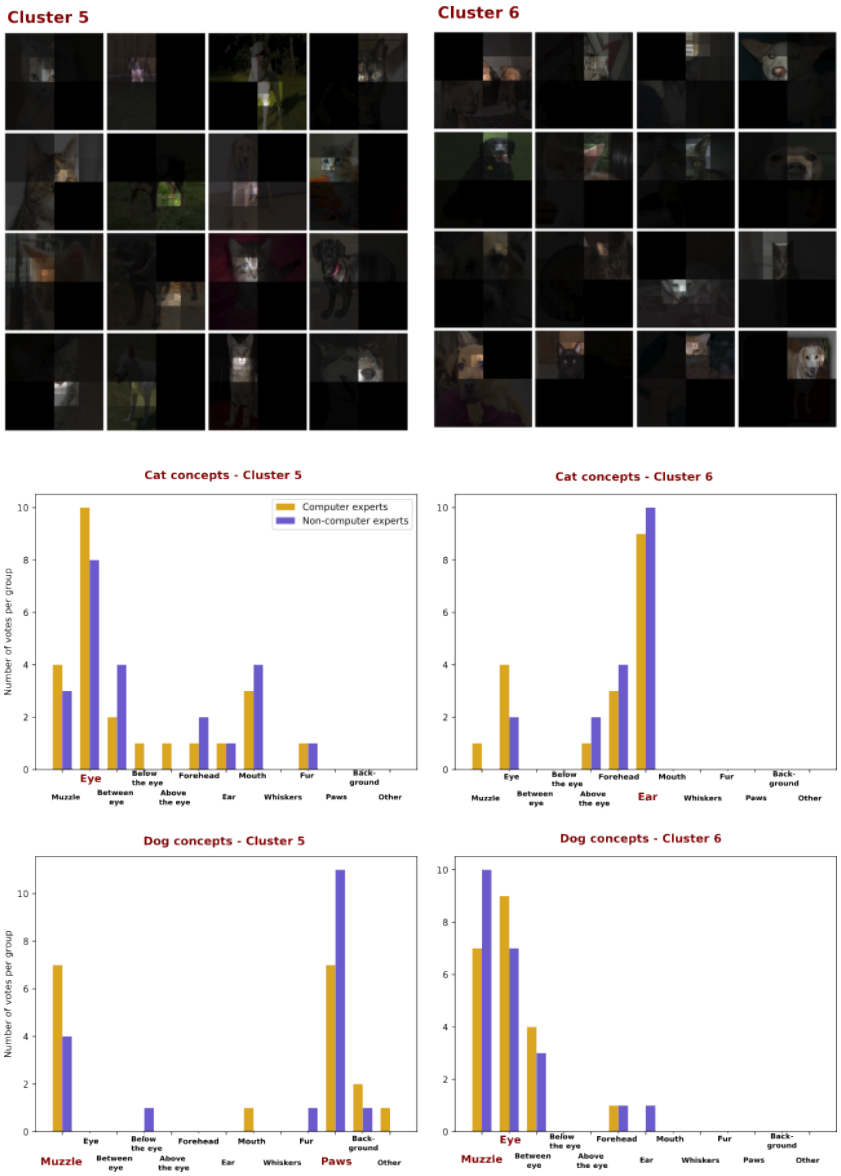


Figure 20: Visualizations obtained for clusters 5 and 6 of VGG16 and results of selected concepts, by 24 participants, to describe the two classes separately. According to the answers, cluster 5 seems not to detect the **eye** for cats and, the **muzzle** and **paws** for dogs. Cluster 6 presents *ear*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