Unsupervised Interpretable Basis Extraction for Concept – Based Visual Explanations

Alexandros Doumanoglou, Stylianos Asteriadis, and Dimitrios Zarpalas

***Abstract*—An important line of research attempts to explain CNN image classifier predictions and intermediate layer representations in terms of human understandable concepts. In this work, we expand on previous works in the literature that use annotated concept datasets to extract interpretable feature space directions and propose an unsupervised post-hoc method to extract a disentangling interpretable basis by looking for the rotation of the feature space that explains sparse one-hot thresholded transformed representations of pixel activations. We do experimentation with existing popular CNNs and demonstrate the effectiveness of our method in extracting an interpretable basis across network architectures and training datasets. We make extensions to the existing basis interpretability metrics found in the literature and show that, intermediate layer representations become more interpretable when transformed to the bases extracted with our method. Finally, using the basis interpretability metrics, we compare the bases extracted with our method with the bases derived with a supervised approach and find that, in one aspect, the proposed unsupervised approach has a strength that constitutes a limitation of the supervised one and give potential directions for future research.**

***Impact Statement*—CNN image classifiers have demonstrated outstanding performance in real-world tasks. They can be used in robotics, visual understanding, automatic risk assessment and more. However, to a human expert, CNNs are often black-boxes and the reasoning behind their predictions can be unclear. Recent advances in explainable and interpretable artificial intelligence (XAI and IAI), attempt to shed light on this process. In an attempt to understand intermediate layer representations, one can project them onto a feature space basis that quantifies the presence of different concepts in the representation. This basis is called interpretable because it can make representations more understandable. In the typical approach, constructing an interpretable basis requires access to annotations. This work proposes a novel unsupervised method to learn such a basis, without the need for explicit labels. This can ease the process of obtaining explanations, eliminate annotation costs, save time, and eventually help humans debug and trust deep models.**

***Index Terms*—Explainable Artificial Intelligence (XAI), Interpretable Artificial Intelligence (IAI), Interpretable Basis, Unsupervised Learning.**

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, "The publication of the article in OA mode was financially supported by HEAL-Link"

A. Doumanoglou is with the Information Technologies Institute (ITI), Centre for Research and Technology HELLAS (CERTH), Thessaloniki, Greece and the Department of Advanced Computing Sciences, University of Maastricht, Maastricht, Netherlands (e-mail: aldoum@iti.gr).

S. Asteriadis is with the Department of Advanced Computing Sciences, University of Maastricht, Maastricht, Netherlands (e-mail: stelios.asteriadis@maastrichtuniversity.nl)..

D. Zarpalas is with the Information Technologies Institute (ITI), Centre for Research and Technology HELLAS (CERTH), Thessaloniki, Greece (e-mail: zarpalas@iti.gr).

This paragraph will include the Associate Editor who handled your paper.

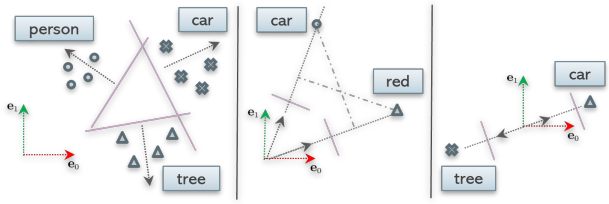

Fig. 1: The natural basis of the feature space is given by $\mathbf{e_0}, \mathbf{e_1}$. **Left:** An interpretable direction is the direction of the feature space along which, the feature representations of a concept lie. **Middle:** A case where the hyperplane normals of two concept detectors (*car*, *red*) are not orthogonal. In this case, the feature representation of a *car* is also classified as *red* and vice versa. Consequently, *car* and *red* are positively correlated and not (linearly) disentangled. **Right:** For a pair of mutually-exclusive concepts, the hyperplane normals of the two concept detectors may form an angle greater than $90°$. However, in a large dimensional feature space with several detectors of mutually-exclusive concepts, the maximum angle between all pairs of hyperplane normals, is approximately $90°$.

I. Introduction

DESPITE the impressive performance of convolutional neural networks (CNNs) in computer vision image classification tasks [1], [2], [3], [4], the understanding of their inner workings still remains a challenge. In an attempt to shed light into the CNN "black-box", the scientific community tries to understand the properties of the intermediate layers' feature space. Early research [5] showed that any possible direction in this feature space may have a semantic meaning, i.e: feature vectors that maximally activate a direction, correspond to image patches that share some sort of semantic concept. For instance, image patches of *car doors*, *cat heads* or *people's faces*, maximally activate different directions of this feature space. Beyond this early result, more recently, rigorous experimentation showed that linear separability of features corresponding to different semantic concepts, increases towards the top layer [6]. The latter has been attributed to the top layer's linearity and the fact that intermediate layers are enforced to produce representations that are helpful to solve the task at hand.

The fact that linear separation of concept representations is possible (especially for layers near the top) [5], has motivated attempts in finding feature space directions for specific concepts [7] and constructing an interpretable feature space basis [8]. In an interpretable basis, each basis vector points

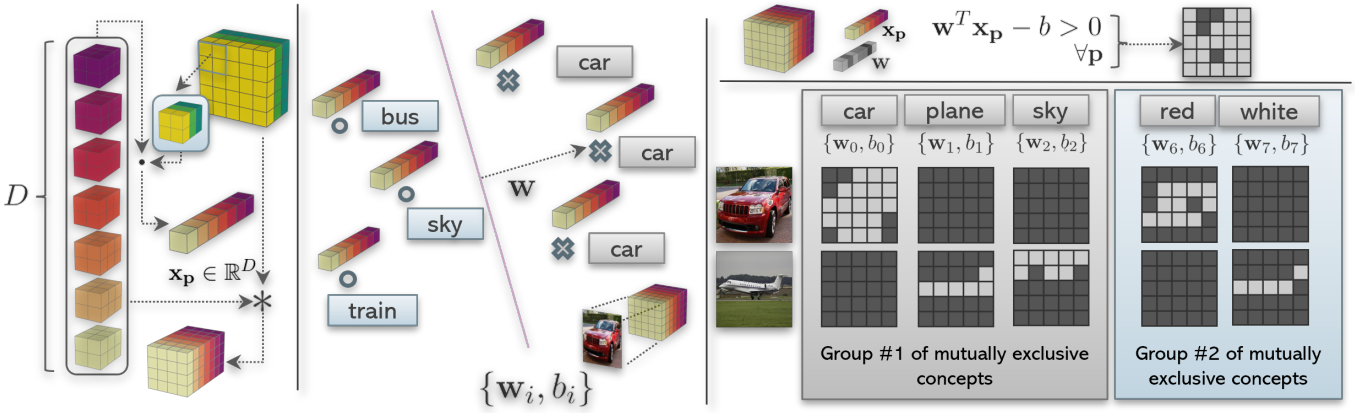

Fig. 2: **Left:** In a standard convolution layer with D filters, all the filters work together to transform each input patch to a feature vector of spatial dimensionality 1×1. Each spatial element $\mathbf{p}$ of the transformed representation, is assigned a feature vector $\mathbf{x_p} \in \mathbb{R}^D$ which lies in the co-domain of the transformation function. Thus the dimensionality of the feature space equals the number of filters in the layer, and each spatial element of the transformed representation, constitutes a sample in this feature space. **Middle:** To find an interpretable basis in the aforementioned feature space in a supervised way, it means to train a set of linear classifiers (concept detectors), one for each interpretable concept, by using feature vectors corresponding to image patches containing the concept. **Right:** In case classifier training succeeds, the application of the classifier rule ($\mathbf{w}^T\mathbf{x_p} - b > 0$) at each spatial element of the representation $\mathbf{x_p}$, produces a binary mask which is active for pixels corresponding to image patches containing the classifier's concept. We observe, that in a successfully learned interpretable basis, a single pixel is classified positively by at most one classifier, among a group of classifiers that are trained to detect mutually-exclusive concepts.

towards the direction of a concept's representations. Projecting a representation onto a basis vector, quantifies the presence of the respective concept in the representation. An interpretable basis can help to obtain possible explanations regarding the CNN and its predictions. When considering the basis vectors as concept embeddings, an interpretable basis can be used to explain the relationship between concepts and filters, similar to what was proposed in [9]. Moreover, it can also be used to interpret predictions of individual examples [8], or used to quantify the class sensitivity of the CNN with respect to a concept [7], [10].

In the typical approach for computing an interpretable basis, the set of interpretable concepts need to be defined in the form of an annotated concept dataset. Using this dataset, one may have access to labels for intermediate CNN representations. These can be subsequently used to find the orientation of the hyperplane that separates the representations of a concept with respect to representations of other concepts [7], [8]. The interpretable basis is constructed by directly using the hyperplane normals as basis vectors. As with any other supervised approach, using an annotated concept dataset to construct an interpretable basis, may increase the fidelity of explanations obtained via that basis. However, this comes at the cost of obtaining the annotations, which is even more prominent when annotations need to be dense (per-pixel) [8]. Additionally, annotated concept datasets are domain-specific, and thus, explaining CNN classifiers for different domains can become even more costly.

The motivation of this work in based on an observation of how an interpretable basis transforms representations. We explain by examples, that the one-hot, thresholded projection of a representation onto an interpretable basis, results into a new, transformed sparse representation. Thus, we propose a method that is able to suggest a feature space basis which satisfies this property found in interpretable bases. In contrast to the typical approach, the proposed method learns the basis directly from the structure of feature space representations, without requiring access to semantic annotations. In that sense, our method can be considered to be unsupervised. However, without annotations, the final suggested basis vectors are not assigned an explicit concept name. In a real-world setting, the concept name associated with a basis vector could be identified by inspecting samples of image patches whose projected representations onto the vector are maximum. For evaluation purposes though, a procedure to label the basis is required, by assigning a concept label to each basis vector, as in [11] or [12].

Our work's contributions can be summarized to the following: (i) We present a **post-hoc**, unsupervised method that suggests an interpretable basis for the feature space of a CNN's intermediate layer. Since post-hoc, the proposed method, applies to pre-trained CNN architectures and does not require any form of retraining them. (ii) Inspired by related work, we propose simple extensions for two basis interpretability metrics. (iii) We provide quantitative evaluation of our method on extracting an interpretable basis for the last layer of popular CNNs, demonstrating applicability to standard architectures. We show that our method is able to improve on the interpretability metrics compared to the interpretability of the natural basis [11], and also compare against a supervised approach [8] to set a baseline for future works and discuss interesting findings that may help future research.

II. Background & Related Work

In this section we discuss background and related-work in five areas related to this paper. First, we briefly describe prior work on supervised interpretable basis extraction and establish the terminology that is used in this paper. Second, we discuss on supervised and unsupervised discovering of interpretable feature space directions. Third, we highlight how the proposed method differs from other works proposing sparsity as a measure to build inherent-interpretable models. Fourth, we explain the basis labeling problem and potential solutions, and finally, discuss interpretability metrics for assessing the quality of a basis.

A. Supervised interpretable basis extraction and terminology

As already mentioned in the introduction, each basis vector of an interpretable basis points towards the direction of a concept's representations. To construct an interpretable basis, Zhou et al. [8] trained a set of binary linear classifiers that separate the CNN's intermediate feature representations based on their semantic meaning. This is accomplished with an densely (per-pixel) annotated concept dataset and implicit use of CNN receptive fields, to assign labels to spatial representation elements of images. Each binary classifier can be considered as a concept detector, since it can separate representations of one concept from representations of other concepts. As already mentioned in Section I, i) the hyperplane normal directions of the linear classifiers can form a (not necessarily orthogonal, and potentially over/under-complete) basis of the feature space and ii) projecting a representation onto a basis vector, quantifies the presence of the respective concept in the representation. When constructing the basis, each basis vector retains the concept label of it's respective concept detector. In a strict sense, the concept detector's bias, which is related to the position of the hyperplane in the feature space, is not part of the basis. For simplicity though, we will retain the association between biases and basis vectors, in such a manner that biases together with the basis vectors form the original concept detectors. For brevity, in this paper, we will use the terms *basis*, *concept detectors*, and *classifiers* almost interchangeably.

B. Discovering concept directions in the CNN feature space

Discovering interpretable directions [5] in the feature space of a CNN image classifier has been previously studied in the literature. In most cases though [6], [8], [7], [10], those directions are directly computed by solving a logistic regression problem that linearly separates CNN's representations based on their concept label. Thus, these methods rely on the existence of an annotated image dataset. To alleviate the need for concept annotations, Ghorbani et al, [13] proposed a method to automatically group semantically similar image patches out of an unlabeled image dataset. The image patches of each group could then be treated as samples coming from the same concept. Subsequently, the concept samples may be assigned pseudo-labels and can be used as label-representation pairs to reveal each concept's direction in the feature space of the CNN. The latter may be accomplished via solving the respective logistic regression problem. In contrast to [13], our approach is fundamentally different. In the proposed work, the thresholded projections of CNN's representations on the learned directions, are sparse. Thus, our work directly tries to exploit existing structure in the CNN's feature representations, instead of using pseudo-labels to convert the problem to a supervised one.

C. Sparsity in Inherent Interpretable Models

The proposed work shares conceptual similarities with [14], [15] and [16]. All previous works are proposing CNN architectures that are inherently interpretable. During training, they enforce intermediate layer representations to be comprised of pixels with sparse activations across feature maps. While we share the same principal idea that sparse pixel activations can lead to more interpretable representations, the proposed method is post-hoc, and has the potential to be applied (possibly) in any pre-trained CNN. In other words, our method suggests a view of the feature space described by the derived basis, that shares similar sparsity properties that other methods enforce during network training. Essentially, and in a more abstract and less strict way, our method reveals the degree that this property is already present in CNNs that were trained without explicitly enforcing this objective.

D. Labeling a feature space basis

We define *basis labeling* as the procedure of assigning a concept label name to each one of its vectors. When the basis vectors have been learned in a supervised way, the concept label to attribute to the each vector is actually known before learning the vector's direction. However, when the basis is learned without annotations (such as the current work) or if the natural feature space basis is considered (as in [11] or [12]), attributing meaning to each basis vector requires to put the vector under test. In the testing procedure, each basis vector is accompanied with a (possibly learned) bias (threshold) to form a linear classifier. Then, for all possible concepts, the suitability of the classifier to separate the representations of one concept (positive samples) with respect to the representations of other concepts (negative samples) is evaluated. Finally, each basis vector is assigned a concept label name based on the evaluation metrics of the aforementioned procedure. It is evident, that labeling a basis requires access to a dataset containing concept annotations, such as [11], [17], or [18]. Bau et al. [11] assigned one concept label to each vector of the natural feature space basis based on the Broden dataset (which was also introduced in the same work). Later, Mu et al. [12] used the same dataset to label the natural basis with logical compositions of concepts (e.g. the concept of "blue AND (NOT water)"). In this work, we use [11] to label the bases extracted with our method, while [12] or other potential future works could also be considered.

E. Metrics to evaluate the interpretability of a feature space basis

In basis evaluation literature [8], [11], [12], measuring the interpretability of a basis slightly varies, depending on whether the basis was learned in a supervised way or not. On one

hand, in case the basis was learned with supervision, Zhou et al. [8] used mean average precision (mAP) considering all the classifiers associated with the basis. On the other hand, to assess the interpretability of the natural basis, Bau et al. [11] considered the number of *unique* concept labels that have been assigned to the basis vectors, provided that the performance of the respective classifiers exceeds a threshold. Those labels come from the basis labeling procedure. In [16], Losch et al. considered Area Under inspectability Curve (AUiC) in order to propose a metric agnostic to a specific threshold. In this paper, we combine ideas from [11] and [16] to propose two metrics that can be used to evaluate the interpretability of a basis.

F. Other Visual Concept Learning Related Work

The proposed approach follows the previous line of research in directional concept discovery [8], [7]. However, recently, and in parallel to our work, other approaches have been developed. In [19], a local explanation method [20] has been extended in order to provide global concept-based explanations. Furthermore, [21] uses subspace clustering for concept discovery, while [22] extends [20] in order to find disentangled concept subspaces that are relevant to the classifier's prediction outcome. Similar to our work, the aforementioned approaches do not require annotations for concept discovery. However, to the best of our knowledge, the proposed approach is the first one to drop the annotation requirement in the directional concept discovery line of research, as the rest of the works deviate from this line, either partially or in total.

III. Motivation

To describe the motivation of our approach, let's assume that we have access to an interpretable basis of a CNN. Let's also assume that the basis was successfully learned, i.e: the CNN representations can be linearly separated based on their semantic label. The latter, implies a) the accuracy of the concept detectors is high and b) the CNN has learned to linearly separate (i.e. disentangle [23]) the aforementioned concepts. In b), disentangled representations can be obtained by projecting representations onto the basis. Inversely, in case the CNN representations could not be linearly separated based on their semantic label, it would mean that the accuracy of concept detectors is low and thus concept disentanglement via a linear transformation is not possible.

For example, let's consider a basis with five concept detectors, one for each element in the set {*car, plane, sky, red, white*}. Consider the images of the red car and white plane of Fig 2 - right. If we apply the concept detectors to the intermediate representation of an image patch, we observe, that among a group of classifiers detecting mutually-exclusive concepts, only one concept detector classifies the patch positively (i.e. as a patch containing the respective concept). For instance, a patch belonging to the concept *car* (such as the one located at second row - second column) is not a *plane* or *sky*, while it is also *red* and not *white*. In that case, {*car, plane, sky*} is a group of mutually-exclusive concepts, and {*red, white*} another one. This simple fact, can be summarized to the following observation: *projecting a representation to an interpretable basis and thresholding, results into sparse one-hot representations*.

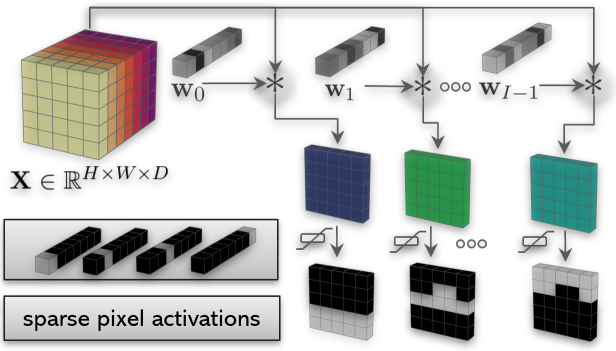

Fig. 3: Overview of the proposed method. Without any form of annotation, our method solves for hyperplane normal directions and thresholds of potential binary and linear concept classifiers, driven by the objective that for a single pixel of an image's representation, only a fraction of the classifiers make positive predictions. The application of the linear classification rules to each pixel in the intermediate representation is accomplished by 1×1 convolution between the image representation $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$ and the classifiers' hyperplane normal directions $\mathbf{w}_i \in \mathbb{R}^D$, followed by bias subtraction and application of the sigmoid activation function. While solving for that objective, the name of the concept that each classifier detects is unknown. In case annotations exist, labeling the basis can be achieved, in a post-processing step, by using methods in related work. In absence of annotations, the concepts can be identified by inspecting samples that the classifiers classify positively.

In this work, we take a non-standard approach to extract an interpretable basis for the feature space of a CNN. Let us consider a set of linear classifiers. In this case, the classification rule dictates projection on the classifier's hyperplane normal and hard-thresholding against the classifier's bias. Based on our previous observation, in case this set of classifiers forms an interpretable basis, applying all the classifiers' rules to a CNN's intermediate representation (with hard thresholding) shall result into a new, transformed representation, which is one-hot and sparse. By optimizing basis vectors and biases for this sparsity objective, the proposed method is able to suggest an interpretable basis without requiring an annotated concept dataset.

IV. Proposed Method

In a typical convolutional neural network (CNN) that is trained for image classification, the intermediate layer representations have a cuboid structure. For a convolutional layer, those representations are calculated by applying the same transformation function (a series of dot products equal in number to the number of filters in the layer) to cuboid patches sliced from the representation of the layer beneath. Thus, the dimensionality of the layer's feature space equals the co-domain dimensionality of this transformation function. In this case, this dimensionality is equal to the number of filters in the layer. Feature vectors at different spatial locations of the

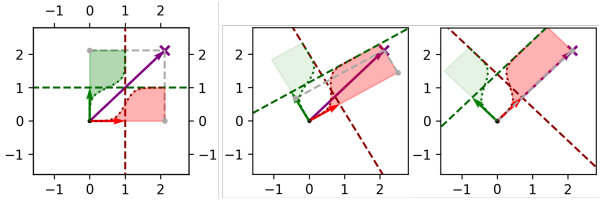

Fig. 4: Consider two feature space basis vectors (first - red, second - green) located at the origin (black). Furthermore, consider each feature space basis vector to be accompanied by a bias threshold which together with the basis vector direction constitutes a linear classifier, with the basis vector pointing towards the direction of higher positive classification confidence. On the **left**, the classifers' separating hyperplanes have been placed at the location of the bias and indicated by (dark red and dark green) dashed lines. Let a feature (purple) lie in this space. The projection points of the feature vector on the basis vectors are marked by gray circular markers. The bottom horizontal and left vertical axes correspond to the standard feature space x and y axis, respectively. The right (top) vertical axis reports the first (second) classifier's confidence levels for each point projected in the basis vector's direction. The exact confidence of the classifier at each projected point on its direction, is given by the sigmoid activation function depicted with the dotted (dark red) (dark green) curve. On the **left**, the classifiers attribute the presence of two concepts in the feature, since the projection of the feature on both basis vectors, exceeds the classifiers' biases by a large margin. This is indicated by (dark red, dark green) shaded areas under the sigmoid curves. The figure in the **middle**, depicts the same situation under a rotation of the basis vectors. In that case, the first classifier makes a confident positive prediction ($\sigma(\cdot) \approx 1.0$) (dark red sigmoid shaded area) for the feature, whereas the second one makes a confident negative prediction ($\sigma(\cdot) \approx 0.0$) (soft green sigmoid shaded area). The figure on the **right** depicts rotation with perfect alignment, where only one of the classifiers classifies the feature positively with high confidence.

cuboid, correspond to different samples from this feature space (Fig. 2 - left). This treatment of the feature space has been also considered in [11], [8], [9], [12]

Let $D \in \mathbb{N}^+$ denote the dimensionality of a layer's feature space, and $\mathbf{x_p} \in \mathbb{R}^D$ an element in this space at the spatial location $\mathbf{p} = (x, y)$ (Fig. 2 - left). In a convolutional layer, D equals the layer's total number of hidden units or, as otherwise mentioned, output channels. Let's consider a set of $I \in \mathbb{N}^+, I \leq D$ linear classifiers to form a (possibly) interpretable basis. The i-th classifier is characterized by its hyper-plane's normal direction $\mathbf{w}_i \in \mathbb{R}^D$ and bias $b_i \in \mathbb{R}$, $i \in \mathcal{I}, \mathcal{I} = \{0, 1, ..., I-1\}$. Additionally, each one of those classifiers is responsible to quantify the presence of one concept in $\mathbf{x_p}$. Last, for the reasons discussed in the Section V, we also consider $\mathbf{w}_i^T \mathbf{w}_i = 1 \, \forall i$, $\mathbf{w}_i^T \mathbf{w}_j = 0 \, \forall \, i, j : i \neq j$, i.e. $\{\mathbf{w}_i\}$ should form an orthonormal basis. We consider I to be a hyper-parameter of the method and, without loss of generality, when $I < D$, the orthogonal basis can be trivially completed to dimensionality D in order to represent a rotation of the feature space. The additional $D - I$ directions can be considered as a non-interpretable *residual*.

The overall concept of our method is depicted in Fig. 3. First, we record CNN intermediate layer representations for images coming from an unlabeled dataset. Starting from the representation of an image $\mathbf{X}$, we project each spatial element $\mathbf{x_p}$ onto all the vectors of the basis $\mathbf{w}_i$ via 1×1 convolution. This operation transforms each pixel of the image representation to the new basis. The result is a new, transformed, cuboid representation. In the new representation, the pixel $\mathbf{p}$ of the i-th feature map has a value equal to the projection of $\mathbf{x_p}$ onto $\mathbf{w}_i$. Subsequently, we threshold the projections with a learned bias b_i and use a sparsity objective to enforce each pixel to have a sparse thresholded representation across feature maps.

To formalize all the previous discussion, consider the standard binary sigmoid classifier $\sigma(\mathbf{w}_i^T \mathbf{x_p} - b_i)$ which, since $||\mathbf{w}_i|| = 1$ and for full expressivity, requires an additional parameter $M_i \in \mathbb{R}^+$, such that, $y_{\mathbf{p},i} = \sigma(\frac{1}{M_i}(\mathbf{w}_i^T \mathbf{x_p} - b_i))$. M_i is controlling the margin between the abscissas corresponding to the extremas of the sigmoid and $y_{\mathbf{p},i} \in (0, 1)$ denotes the confidence of classifier i to classify $\mathbf{x_p}$ positively. Without loss of generality and for mathematical and implementation convenience, we standardize the feature space with batch normalization [24] and without affine parameters. We do so, just after projecting $\mathbf{x_p}$ to $\mathbf{w}_i$ and before subtracting the bias b_i or dividing by M_i. As already mentioned, the projection of each $\mathbf{x_p}$ to the new basis is accomplished via standard 1×1 convolution with D input and I output channels. While searching for $\mathbf{w}_i$, the standardization of the feature space allows treating the magnitude of projections ${\mathbf{x_p}}^T \mathbf{w}_i$, biases b_i, and margin coefficients M_i in the same scale, respectively, regardless of i. Thus, this allows us to make a simplification to the parameter space of our model and consider $b_i = b$ and $M_i = M$ (i.e. equal biases and margins in the standardized space) for all i. Orthogonality of the extracted basis is enforced by using [25]. The learnable parameters of our model are simply $\mathbf{w}_i, b$ and M, while b_i and M_i can be later recovered by inverting the standardization process. A graphical explanation of the principal idea in the 2D feature space is provided in Fig. 4 and the pipeline of the proposed method is given in Figure 5.

In the rest of the section, we introduce the loss terms that we use to derive an interpretable basis. For notation convenience, we assume $\mathbf{p}$ to vary across the spatial dimensions of all image representations in the dataset. Moreover, a pixel $\mathbf{p}$ is considered to be assigned to the i-th concept detector, when for the given $\mathbf{x_p}$, $y_{\mathbf{p},i} \gg 0.5$. In that case, we also say that $\mathbf{x_p}$ is classified positively by the same concept detector. In a similar analogy, we mention $\mathbf{x_p}$ to be classified negatively by the i-th concept detector whenever $y_{\mathbf{p},i} \ll 0.5$.

Sparsity Loss (SL) Let $\mathbf{y_p} = [y_{\mathbf{p},0}, y_{\mathbf{p},1}, ..., y_{\mathbf{p},I-1}]^T$ denote the vector of activations containing the classification results for $\mathbf{x_p}, \forall i \in \mathcal{I}$. The criterion that guides our search for $\mathbf{w}_i$ implies sparsity in this vector of activations. Under the sparsity criterion, each pixel $\mathbf{p}$ is classified positively only by a portion of the classifiers in the new basis. We use entropy as a sparsity

measure and define the sparsity loss $\mathcal{L}^s$ as:

$$\mathcal{L}^s = \mathbb{E}_{\mathbf{p}}\Big[-\sum_{i\in\mathcal{I}} q_{\mathbf{p},i}\log_2(q_{\mathbf{p},i})\Big] \tag{1}$$

with

$$q_{\mathbf{p},i} = \frac{y_{\mathbf{p},i}}{\sum_{i\ \in\mathcal{I}} y_{\mathbf{p},i}} \tag{2}$$

Maximum Activation Loss (MAL) The sparsity criterion alone is not sufficient to extract a meaningful basis. This is better understood when considering the fact that entropy is applied in a relative scaling of the activation magnitudes, due to (2). Thus, when optimizing $\mathcal{L}^s$ alone, a pixel's activations may be considered sparse by eq (1), but $\mathbf{x_p}$ might still be classified negatively by all concept detectors in the basis, i.e. $\max_i y_{\mathbf{p},i} < 0.5$. For a meaningful basis, we would like to have each pixel assigned to at least one concept detector. To this end, we add an additional loss term $\mathcal{L}^{ma}$ that encourages the most confident concept detector to not only be the most confident in a relative scale (compared to other concept detectors) but also in an absolute scale, reporting high confidence levels towards 1 (Fig. 6)

$$\mathcal{L}^{ma} = \mathbb{E}_{\mathbf{p}}\Big[-\sum_{i\in\mathcal{I}} q_{\mathbf{p},i}\log_2(y_{\mathbf{p},i})\Big] \tag{3}$$

In (3), $\log_2$ is chosen for its strong guiding gradient when $\max_i y_{\mathbf{p},i} \ll 0.5$. From another viewpoint, this loss in combination with the sparsity loss, imposes each pixel to be classified positively with high confidence from the most confident classifiers and negative with high confidence from the remaining classifiers.

Inactive Classifier Loss (ICL) The two previous losses while they encourage assigning each pixel to a basis vector, they do not encourage, in any way, diversity in the assignments. For instance, all pixels could be assigned to one concept detector, with the rest of the detectors having no pixels assigned to them. In that case, the classifiers associated with an empty pixel set (i.e. when no pixel is assigned to them), actually never classify a pixel positively and thus the sparsity criterion can be easier fulfilled. Besides, if all pixels in the dataset are classified negatively by a classifier, then this classifier does not convey any meaningful information, it cannot serve as a concept detector and is redundant.

To moderate this issue we introduce the inactive classifier loss. We design a loss term that linearly penalizes basis vectors with a few number of pixel assignments. This number is defined as a percentage threshold over the total number of pixels in the dataset. Instead of specifying this threshold for each $i \in \mathcal{I}$ individually, we introduce a set of hyper-parameters to make this more manageable. Let $\alpha_\mu \in [0,1]$ denote a percentage coefficient with $\sum_\mu \alpha_\mu = 1$ and $\alpha_0 \geq \alpha_1 \geq ... \geq \alpha_{N-1}$, $N \in \mathbb{N}^+$, $\mu = \{0,1,...,N-1\}$. We split $\mathcal{I}$ in N partitions with each partition having $n_\mu \in \mathbb{N}$ elements:

$$n_\mu = \begin{cases} \lfloor \alpha_\mu I \rfloor + 1 & \mu >= I - R \\ \lfloor \alpha_\mu I \rfloor & \text{otherwise} \end{cases} \tag{4}$$

with $R = I - \sum_\mu \lfloor \alpha_\mu I \rfloor$, and $\lfloor\cdot\rfloor$ denoting the floor operation. The previous procedure ensures that $\sum_\mu n_\mu = I$ while n_μ remains integer. Let $\tau \in [0,1]$ denote a percentage threshold over the total number of pixels in the dataset. We distribute τ across the concept detectors using a weighting scheme that utilizes the same weight $\omega_\mu \in \mathbb{R}^+$ for all detectors in the same partition. The i-th concept detector is penalized whenever the percentage of pixels assigned to it falls below the threshold ν_i given below:

$$\nu_i = \frac{\omega_\mu \tau}{\sum_\mu \omega_\mu n_\mu} \tag{5}$$

From (5) it becomes apparent that all concept detectors in the same partition share the same threshold. Finally, we define the *inactive classifier* loss as

$$\mathcal{L}^{ic} = \mathbb{E}_{i\in\mathcal{I}}\Big[\frac{1}{\nu_i}\text{ReLU}\big(\nu_i - \mathbb{E}_{\mathbf{p}}[y_{\mathbf{p},i}{}^\gamma)]\big)\Big] \tag{6}$$

where the factor $\frac{1}{\nu_i}$ before the rectified linear unit activation function (ReLU) [26] normalizes the loss to be 1 when all concept detectors classify negatively the whole pixel dataset. The exponent $\gamma \in \mathbb{R}^+, \gamma > 1$, acts as a sharpening operator on $y_{\mathbf{p},i}$, in order to attenuate non-confident predictions that lie around 0.5.

Maximum Margin Loss (MML) Since M controls the margin between the abscissas corresponding to the extremas of the sigmoid classifier, we add an additional loss term that encourages a large classification margin in a similar sense as in the Support Vector Machine [27]. We enforce M to be a positive scalar via the parameterization $M = 1/t^2$ and simply define the maximum margin loss as:

$$\mathcal{L}^{mm} = \frac{1}{M} = t^2 \tag{7}$$

Conclusively, we introduce four loss terms that guides search for an interpretable basis. First, the Sparsity Loss (SL) which enforces each pixel to be classified positively by only a fraction of the concept detectors in the basis. Second, the Maximum Activation Loss (MML) which in combination with the sparsity loss enforces the most confident predictions in the relative scale (as implied by $\mathbf{q_p}$), to also be confident in an absolute scale (as given by $\mathbf{y_p}$ and close to 1). Third, the Inactive Classifier Loss (ICL), which penalizes classifiers that never classify any pixel positively and last, the Maximum Margin Loss (MML) which enforces large hyperplane separation margin (in the SVM sense) between the positive and negative predictions of the classifiers.

V. Basis Orthogonality

To explain why we apply orthogonal constraints for extracting an interpretable basis, it is better to individually consider cases where those contraints are absent. To begin with, let's consider the case where two concepts in a concept pair belong to different groups of mutually exclusive concepts. In a slightly informal way where strict linear relation is not considered, this makes the two concepts either independent from each other, or positively correlated, since mutual-exclusivity implies negative correlation. In the first case, it is apparent that the respective basis vectors should be orthogonal. To give a counter example, let's consider the concept *car* from the group of *objects* = {*car, tree, person*} and the concept *red*

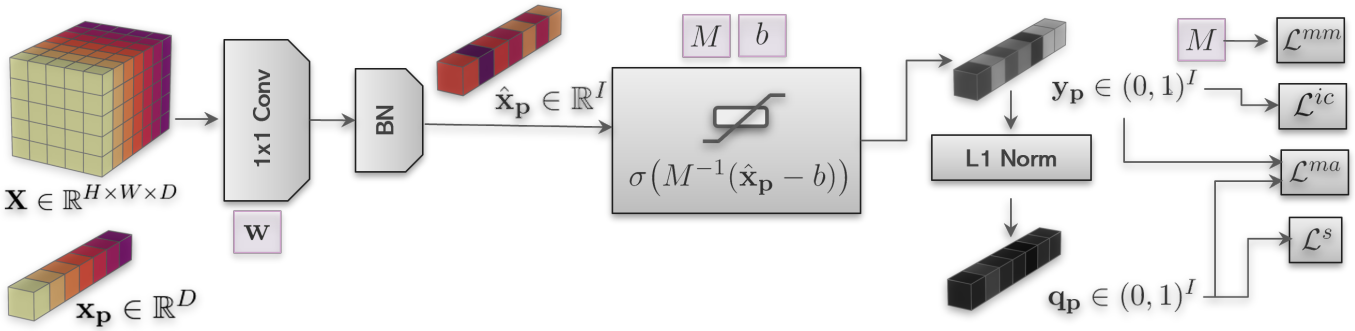

Fig. 5: The basis learning pipeline of the proposed method. Learnable parameters are given in purple next to the operations that actually use them.

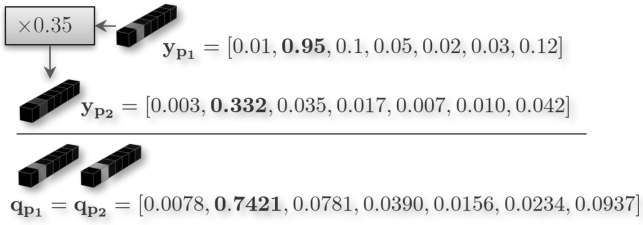

Fig. 6: An example why pixel activation sparsity, which is enforced through entropy, is not alone sufficient to provide a meaningful basis. Since entropy can be applied only on probability distributions, $\mathbf{y}_\mathbf{p}$ is L1 normalized to $\mathbf{q}_\mathbf{p}$ before enforcing sparsity. This may lead to a set of classifiers that satisfy the sparsity criteria on $\mathbf{q}_\mathbf{p}$ but actually none of the classifiers classifies $\mathbf{x}_\mathbf{p}$ positively (i.e. with high confidence $\gg 0.5$). Thus, $\mathbf{x}_\mathbf{p}$ has no concept assigned to it. If we exaggerate to many $\mathbf{p}$, this may lead to a basis that does not classify positively any of the pixels. In the figure, $\mathbf{y}_{\mathbf{p}_2}$ is derived by scaling $\mathbf{y}_{\mathbf{p}_1}$ by 0.35. While both $\mathbf{p}_1$, $\mathbf{p}_2$ have sparse activation in the probability scale (described by $\mathbf{q}_\mathbf{p}$), only $\mathbf{p}_1$ has a concept assigned to it with high confidence. To mitigate this, we introduce the maximum activation loss which enforces strong activation magnitudes from the most positively confident classifiers.

from the group of *colors* = {*red, green, blue*}. In case the angle between the basis vectors of these two concepts is less (greater) than 90°, some feature vectors that are classified as *car* will inevitably also be classified as (not) *red* and vice versa (Fig. 1 - Middle). This relation implies dependence which is contradictory to our initial assumption that the two concepts are independent. While this bias may be encoded in the CNN's weights, this fact also means that the two concepts are not (linearly) disentangled, eventually harming the interpretability of the feature space. Since our primary goal is to search for an interpretable basis, given the previous discussion, we know a-priory that a non-orthogonal basis cannot satisfy the interpretability criteria for independent concepts.

For the second case, where the two concepts are positively correlated, the two concepts could possibly be related with a *has-a* relationship. For instance, *car* has a *car-door* and a *car-wheel*. In this case, an image patch of the concept *car-door* or *car-wheel* may also be classified as *car*. Vice versa, a representation of a *car* may have positive components in the direction of *car-door* and *car-wheel*, to justify the *has-a* relationship. This case is not handled by the proposed method. However, the primitive concepts, *car-door* and *car-wheel*, are mutually exclusive.

Thus, for this last case, considering two concepts coming from the same group of mutually-exclusive concepts, it could be reasonable to expect that this mutual exclusivity, which implies negative correlation, is also encoded in the angle between the respective basis vectors. In that case, the angle between the respective basis vectors could be greater than 90° Fig. 1 - Right. To investigate the degree that this is possible, we formulate the problem in a way that is independent from input data. To construct an (ideal) basis for negatively correlated concepts, one might consider embedding I concept vectors in a D dimensional space by maximizing the minimum angle across all pairs of vectors. As it turns out this is linked to spherical coding theory [28] and the tammes problem [29]. Although more sophisticated approaches exist [30], [31], we tried to approximately solve the tammes problem via directly maximizing the minimum pairwise vector angle with gradient decent. Experimental results showed that the resulting embedding vectors, in cases where $I \geq 64$, are close to orthogonal. Fig 7 depicts distribution statistics for various pairs of I, D with $I \leq D$. Conclusively, we argue that an orthogonal basis can cover (under some approximation) independent and mutually exclusive concepts but not concepts that are positively correlated.

VI. Evaluation Metrics

A. Basis labeling and classifier validation scores

To quantitatively evaluate for interpretability the bases extracted with our method, we use a two step process. First, after deriving a basis, we use the work of Bau at al. [11], [32], to assign a concept label to each classifier associated with the basis vectors. Let $\phi(i, c, \mathcal{K}) \in [0, 1]$ denote a metric score function that is used to measure the *suitability* of the classifier i ($\{\mathbf{w}_i, b_i\}$) to accurately detect concept c in the annotated concept dataset $\mathcal{K}$. The concept label that is assigned to classifier i is the one that maximizes $\phi(i, c, \mathcal{K}_{train})$ across c over the training split $\mathcal{K}_{train}$ of the concept dataset. Subsequently, in the second step, and using the validation split of the concept

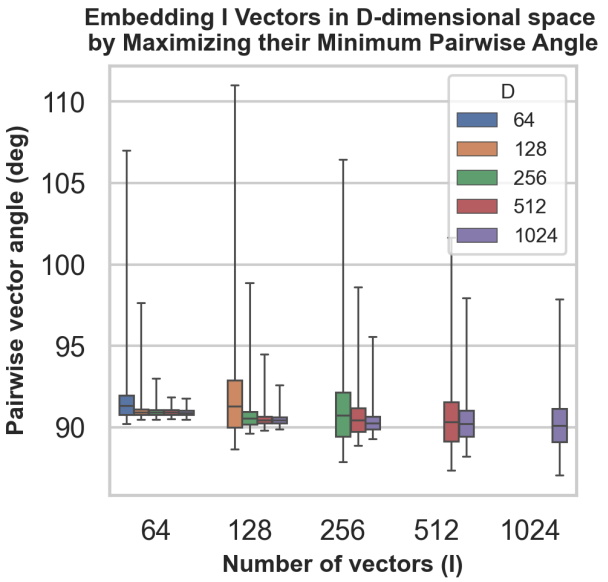

Fig. 7: Pairwise vector angle distribution when solving the tammes problem. Extremas of the error bars correspond to the minimum and maximum vector pair angle. The horizontal line in the box is equals to the mean of the distribution and box widths are equal to the standard deviation.

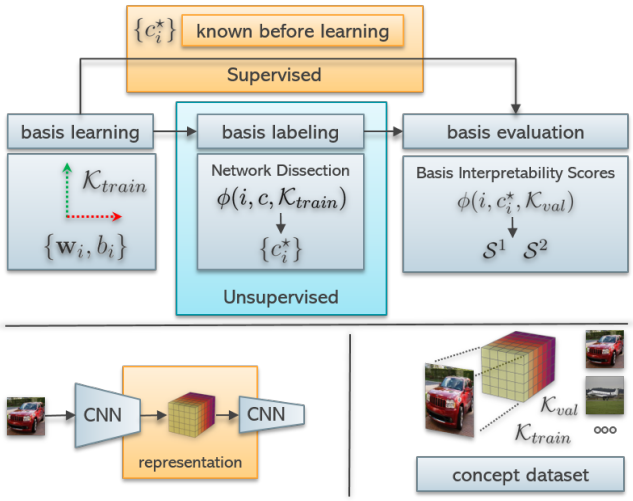

Fig. 8: The pipeline for evaluating the interpretability of a basis. The basis labeling procedure is only required when the learned basis was derived in an unsupervised way or when considering the natural feature space basis. In the supervised case, the concept label for each basis vector is actually known before learning the respective concept detector.

dataset $\mathcal{K}_{val}$, each classifier is assigned a validation score $\phi(i, c_i^\star, \mathcal{K}_{val})$, with $c_i^\star$ denoting the concept label assigned to the classifier during the first step. For the choice of ϕ we use Intersection Over Union (IoU), as originally proposed in [11] and also used in [12], [9]:

$$\phi(i, c, \mathcal{K}) = \frac{\sum_{\mathbf{k}\in\mathcal{K}} |\mathbf{M}^i(\mathbf{k}) \cap \mathbf{L}^c(\mathbf{k})|}{\sum_{\mathbf{k}\in\mathcal{K}} |\mathbf{M}^i(\mathbf{k}) \cup \mathbf{L}^c(\mathbf{k})|} \tag{8}$$

In (8), $\mathbf{M}^i(\mathbf{k})$ denotes the upsampled, hard-thresholded (binarized) map of image $\mathbf{k}$. $\mathbf{M}^i(\mathbf{k})$ is obtained by applying the rule of the i-th classifier $(\mathbf{w}_i^T \mathbf{x_p} - b_i > 0)$ to each $\mathbf{x_p}$ of the upsampled image's representation. Moreover, $\mathbf{L}^c(\mathbf{k})$ denotes the ground truth segmentation map of image $\mathbf{k}$ for concept c and $|\cdot|$ denotes the cardinality of a set. Overall, to label the bases and compute classifier validation scores, we use the exact scheme of [11] with two differences. First, we consider a train/test split of the concept dataset as originally proposed in [9] and second, for hard-thresholding in $\mathbf{M}^i(\mathbf{k})$, we use the biases learned from our method, instead of using the statistical quantile learning of [11].

B. Overall basis interpretability scores

Inspired from [11] and [16] we propose two metrics $\mathcal{S}^1$ and $\mathcal{S}^2$ that can be used to measure the interpretability of a basis. Those metrics, essentially aggregate the aforementioned individual classifier validation scores into scalar values that can summarize the interpretability of a basis.

The first, counts the number of concept detectors in the basis with a validation score better than a threshold ξ. In order to make it threshold agnostic, we measure the area under the indicator function ($\mathbb{1}(x)$) for all $\xi \in [0, 1]$:

$$\mathcal{S}^1 = \int_0^1 \sum_{i=0}^{I-1} \mathbb{1}_{x\geq\xi}\big(\phi(i, c_i^*, \mathcal{K}_{val})\big) d\xi \tag{9}$$

This metric is similar to what was proposed in [16] with two differences. First, we use IoU as the choice of ϕ in order to comply with our intention to use [11] for labeling the basis. And second, unlike [16], we do not normalize (9) with the number of vectors in the basis, in order to be able to make absolute comparisons between scores for bases of different sizes.

The second metric, counts the number of unique concept labels over the set of labels whose respective concept detectors exhibit performance better than ξ. This metric is the same as the one proposed in [11]. Inspired by [16], and with the intention to also make it agnostic to the threshold ξ, we use the area under curve:

$$\mathcal{S}^2 = \int_0^1 \psi(\xi) d\xi \tag{10}$$

with $\psi(\xi) = |\{c_i^\star \,|\, \exists\, i : \phi(i, c^\star, \mathcal{K}_{val}) \geq \xi\}|$, i.e. the number of unique concept detectors exhibiting performance better than ξ.

VII. Experimental Results

Overall Evaluation Approach To the best of our knowledge, the proposed method is the first unsupervised method to suggest an interpretable basis. In addition to this, and once again to the best of our knowledge, except from [11] which performs this in a statistical manner, the proposed method is the first unsupervised method to also provide an estimate for the position of the hyperplane that separates each concept's representations from the representations of other concepts. Therefore, we quantitatively evaluate the interpretability of

the bases extracted with the proposed method against the interpretability of the natural feature space basis (*baseline*). Apart from this, we quantitatively evaluate the bases extracted with our method with the bases extracted via the supervised approach of [8] and thus setting a baseline for future unsupervised works.

An exhaustive search and ablation study over all hyper-parameters is difficult, due to the sheer number of parameters, combinations and computational resource constraints. Nevertheless, the results presented below show that by making simple and intuitive hyper-parameter choices, one may obtain a basis that is more interpretable than the natural. In [11], Bau et. al proved experimentally that the natural feature space basis is more interpretable than other random bases. In this work, we build on the previous findings of [11] and show that the proposed method is able to suggest a basis which is more interpretable than the natural and consequently more interpretable than most other random bases. In short, the main advantage of the proposed method is that it can provide an improvement over the interpretability of the natural basis, and do so, without annotations. Moreover, future, more exhaustive work on fine-tuning strategies has the potential to further improve interpretability.

In all of our experiments we used the Broden [11] concept dataset to probe the networks and obtain intermediate layer feature representations. Except for comparison with the supervised approach (Section VII-C), where we only used the *object* and *part* categories of the dataset, on all other experiments we used the complete set of concept categories, namely {*scene, object, part, texture, material, color*}. In all experiments, we used post ReLU activations of the considered network's *last-layer*. A network's *last-layer* refers to the latest convolutional or max-pooling layer where the representation remains spatial, before the flattening to the latest fully-connected one.

To learn an interpretable basis with the proposed method, we used the training split of the concept dataset. Next, we used the same training split to label the basis using [11], and finally, we calculated the basis interpretability scores (eq. (9), (10)) using the validation split of the same dataset. Annotation labels were only used to label the bases and perform quantitative evaluation, and were not used in any way to learn the aforementioned bases. Regarding the evaluation of the natural basis (*baseline*), we used $\mathbf{w}_i = \mathbf{e}_i$, $\mathbf{e}_i = [\underbrace{0, ..., 0}_{i \text{ times}}, 1, \underbrace{0, ..., 0}_{D-i-1 \text{ times}}]^T$ and we chose the thresholds b_i according to the top 0.005 – quantile among the population of projected representations, as suggested by [11]. The rest of the evaluation pipeline was the same as before. Finally, to establish comparisons, we also used the same interpretability score functions of Section VI, in order to evaluate the bases extracted with the supervised approach of [8]. In that case, the bases were learned in a supervised way using the training split of the concept dataset. Given the a-priory known concept labels of the basis vectors, evaluation was performed on the validation split of the dataset, ommiting the basis labeling procedure which is not required. The overall evaluation pipeline is depicted in Fig. 8.

Basis Learning Details To learn each one of the basis, we used the Adam [33] optimizer with the default beta parameters $(0.9, 0.999)$ provided by the PyTorch [34] implementation. We fixed the learning rate to 0.001 and did not employ any form of learning rate scheduling. In all cases, basis learning lasted for 300 epochs. Batch size was a variable that varied across our experiments and its value was based solely on the available GPU memory resources. The values we used, approximately lied in the interval $\approx [800 - 3600]$.

TABLE I: The loss weight coefficients that we used for learning all our bases. In case of ablation studies, the deviations from these values are given in the respective Section. The superscript of each weight follows the notation of the respective loss.

λ^s	λ^{ma}	λ^{ic}	λ^{mm}
2.0	5.0	5.0	0.5

Hyper-parameters We kept most of the hyper-parameters of our method fixed to the same values across all the presented experiments, except for the parameters we wanted to ablate. We linearly combined the loss terms with the weights given in Table I. Empirical evaluation showed that λ^{ma} should have higher weight than λ^s due to the fact that even if the entropy sparsity criterion is fulfilled, the basis may still be not meaningful (Fig. 6). The choice for the rest of the weights was guided by intuition for the relative importance across loss terms. In all of our experiments we used $I = D$, while extensive study for cases where $I < D$ is left for future work.

Parameter Initialization In all of our experiments we initialize the basis vectors with the vectors of the natural feature space basis (i.e $\mathbf{w}_i = \mathbf{e}_i$). We also initialize t and b with $t = 0.5$ and $b = 0.5$.

A. Ablation studies

In this section, we present three ablation studies regarding Maximum Margin (*MML*) and Inactive Classifier (*ICL*) losses. To do so, we choose two different CNN architectures trained on two different datasets. In particular, we extract bases for the *last layer* of ResNet18 [4] and VGG16 [2] with batch normalization blocks (VGG16BN). The ResNet18 that we used, was trained on Places365 [35], while VGG16BN was trained on ImageNet [36].

Ablation of MML weight in absence of ICL In the first ablation, we set $\lambda^{ic} = 0$ (i.e. completely eliminated the Inactive Classifier Loss) and varied λ^{mm} to take values from the set $\{0.5, 1.0, 1.5\}$. The basis interpretability scores are given in Fig. 9 and 10. For both networks, we observe that all the bases extracted with the proposed method score significantly lower in terms of $\mathcal{S}^1$ than the baseline. In absence of ICL, this fact was actually expected, for the reasons described in Section IV. For ResNet18, the proposed method extracted bases that were slightly more interpretable than the baseline according to $\mathcal{S}^2$, while for VGG16BN and the same metric, none of the learned basis scored higher compared to the baseline. Overall, we could say that for those cases, the sensitivity of the method with respect to λ^{mm} was rather small. This stems from the fact that, according to those metrics, the learned bases are approximately equally interpretable, even though they were learned using different λ^{mm}.

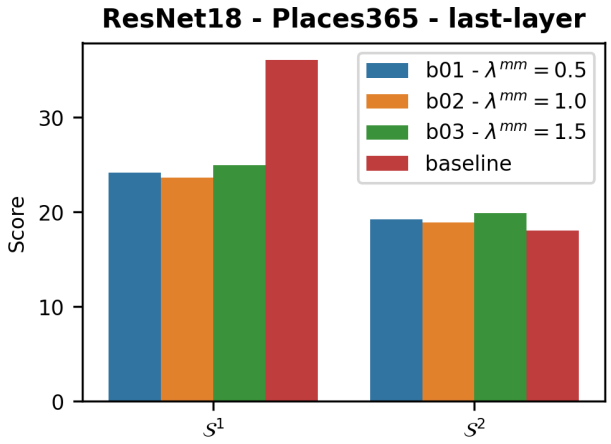

Fig. 9: Ablation study for λ^{mm}. ICL is not used in these experiments. Without ICL, the interpretability of the extracted basis is significantly worse than the baseline in terms of $\mathcal{S}^1$, and slightly better than the baseline in terms of $\mathcal{S}^2$.

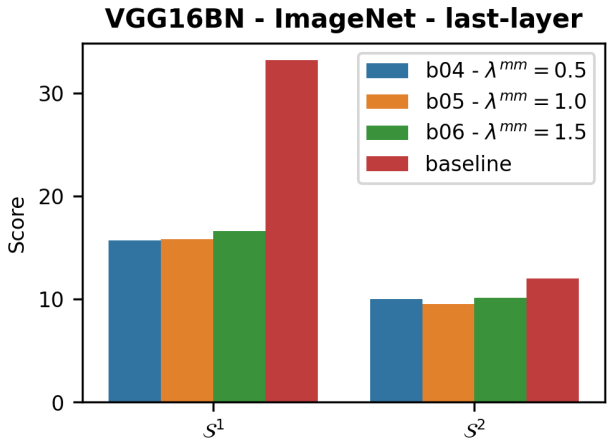

Fig. 10: Ablation study for λ^{mm}. ICL is not used in these experiments. Without ICL, the interpretability of the extracted basis is worse than the baseline in terms of both $\mathcal{S}^1$ and $\mathcal{S}^2$.

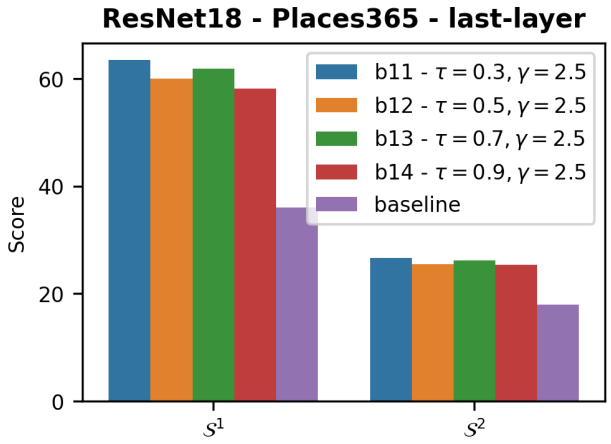

Fig. 11: Ablation with respect to τ. With the addition of ICL the interpretability of the bases extracted with out method is improved compared to the baseline.

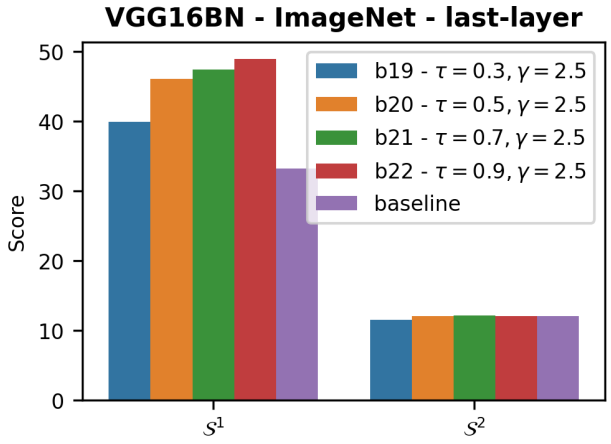

Fig. 12: Ablation with respect to τ. With the addition of ICL the interpretability of the bases extracted with out method is improved compared to the baseline, at least for $\mathcal{S}^1$. For $\mathcal{S}^2$, the interpretability of the same bases are comparable to the interpretability of the baseline.

Ablating τ with ICL In the second ablation, we make use of *ICL* and set $\lambda^{mm} = 0.5$ and $\lambda^{ic} = 5$. For comparisons, we vary τ to take values from the set $\{0.3, 0.5, 0.7, 0.9\}$. In this study we also use one partition ($N = 1, \alpha_0 = 1, \omega_0 = 1$) and set $\gamma = 2.5$. Basis interpretability scores are given in Fig. 11 and 12. The first observation for $\mathcal{S}^1$ is that, in contrast to the previous ablation and for both networks, the extracted bases are significantly more interpretable compared to the baseline. This, experimentally demonstrates the importance of ICL to obtain a meaninful basis. For $\mathcal{S}^2$, a notable improvement over the baseline is provided for ResNet18, while for VGG16BN the interpretability of all the bases, regardless of τ, are comparable to the baseline. Overall, regarding $\mathcal{S}^1$, the value of τ seemed to have larger impact on the bases learned for VGG16BN compared to the bases that were learned for ResNet18, with increasing values of τ resulting into larger interpretability scores. We think it is reasonable to believe, that this behaviour possibly indicates that the impact of τ on the basis interpretability results also depends on the network architecture, the dataset used to train it and its relation with the concept dataset that was used to learn the basis.

Ablating partition count with ICL In the last ablation, we study the effect of partition count to the basis interpretability scores. In these experiments we considered two cases with different number of partitions (Section IV). In both cases, given the number of partitions N, we used the following hyperparameters: $\alpha_\mu = 1/N, \mu = \{0, 1, ..., N-1\}$, and $\omega_\mu = \mu + 1$. In particular, for the first case we used two partitions ($N = 2$) with $\alpha_\mu = 0.5$, $\mu \in \{0, 1\}$, $\omega_0 = 1$, $\omega_1 = 2$ and in the second case we used four partitions ($N = 4$ with $\alpha_\mu = 0.25, \mu \in \{0, 1, 2, 3\}$ and $\omega_0 = 1$, $\omega_1 = 2$, $\omega_2 = 3$, $\omega_3 = 4$. In these

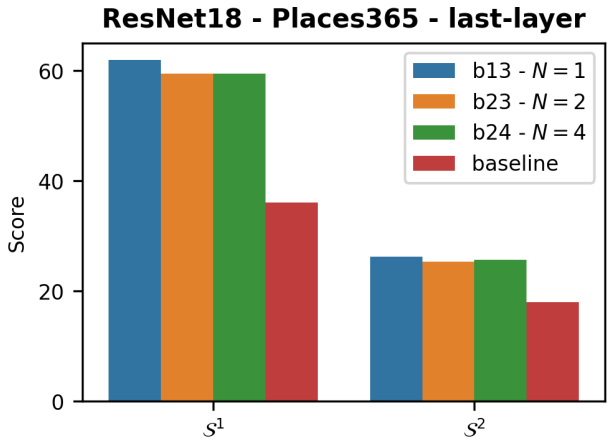

Fig. 13: Ablation with respect to the number of partitions N. For ResNet18, just a single partition resulted into the most interpretable basis. However, for other values of N the results are comparable and an improvement is noted compared to the baseline.

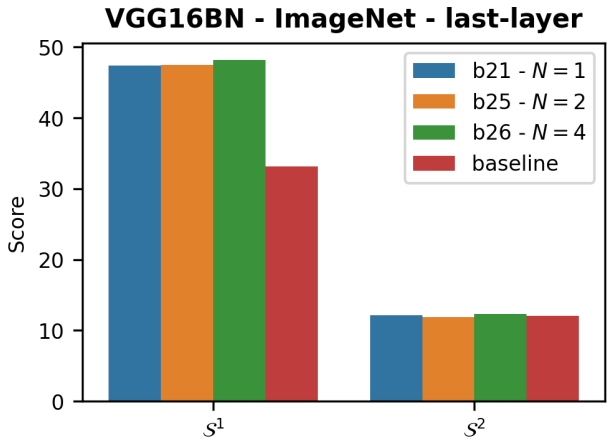

Fig. 14: Ablation with respect to the number of partitions N. For VGG16BN, four partitions resulted into the most interpretable basis. However, for other values of N the results are comparable and an improvement is noted compared to the baseline.

experiments we used $\tau = 0.7$ and $\gamma = 2.5$. Interpretability results are provided in Fig. 13, 14. Regarding ResNet18 (Fig. 13) we observe that using a single partition ($N = 1$) slightly improves the interpretability metrics among the bases that were learned with a larger number of partitions. For VGG16BN, the same slight improvement applies for the basis that was learned with $N = 4$. Overall, we could say, that on those experiments and for the given interpretability metrics, the sensitivity of the method with respect to partition count is rather low.

B. Results for more networks

In this section we apply the proposed method for interpretable basis extraction to two more networks. We consider AlexNet [1] (trained on Places365) and GoogleNet [3] (trained on ImageNet). Regarding hyper-parameters, we use the loss weight factors of Table I, $\tau = 0.7$, $N = 1$, $\alpha_0 = 1.0, \omega_0 = 1.0$, $\gamma = 2.5$. We provide basis interpretability results that show improvement over the baseline in Fig. 15 and 16.

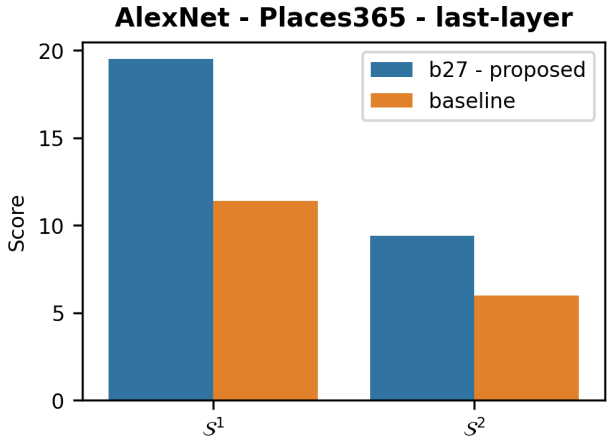

Fig. 15: Interpretability comparison between the baseline and a basis extracted with the proposed method. The proposed method suggested a more interpretable basis than the baseline.

C. Comparison with a supervised approach

In this section we compare the interpretability of bases extracted with the proposed method against the *baseline* and the supervised approach of [8]. Once again, we consider the *last-layers* of ResNet18 (trained on Places365) and VGG16BN (trained on ImageNet). We followed the approach of Interpretable Basis Decomposition (IBD) [8] and learned a basis in a supervised way for the concepts categories of *objects* and *parts*. To learn the basis we used the training split of the concept dataset. The number of basis vectors that were learned from IBD was $I = 660$ while the dimensionality of the feature space for both CNNs is $D = 512$. Regarding the proposed method, for ResNet18, we re-considered the basis *b13* (Fig. 11) which was learned from all images (regardless the category annotations) of the concept dataset's training split. This time though, we only considered the concept categories of *objects* and *parts* to label the basis. We did the same for the natural feature space basis as well. Finally we report $\mathcal{S}^1$ and $\mathcal{S}^2$ on the validation split of the same dataset. A similar approach was taken for VGG16BN, where we re-considered the basis *b26*. The results for the two networks are given in Fig. 17 and 18.

From the previously mentioned figures, we first observe, that the bases learned with IBD have the same score on both metrics. This is actually expected, since all concept labels in a basis learned with IBD are unique. On the contrary, when labeling the natural feature space basis or a basis extracted with the proposed method, the same concept label may be attributed to more than one basis vectors. This also might be a possible explanation for why the proposed method showcases significantly better interpretability scores for $\mathcal{S}^1$ compared to

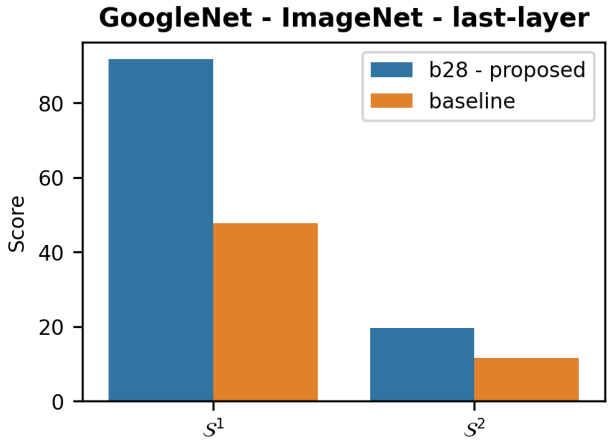

Fig. 16: Interpretability comparison between the baseline and a basis extracted with the proposed method. The proposed method suggested a more interpretable basis than the baseline.

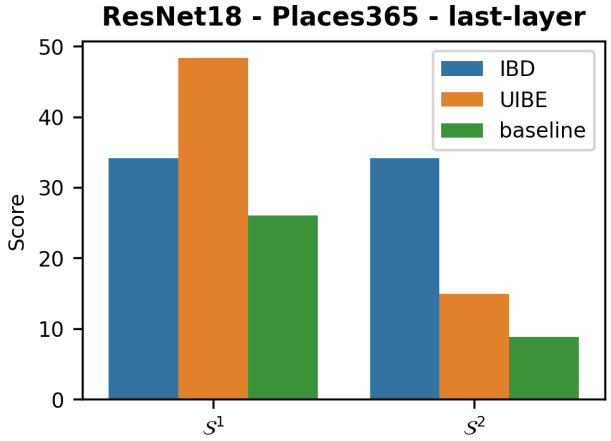

Fig. 17: Comparing basis interpretability of the proposed method (*UIBE*) with the natural feature space basis (*baseline*) and a basis extracted with a supervised approach (*IBD*) [8].

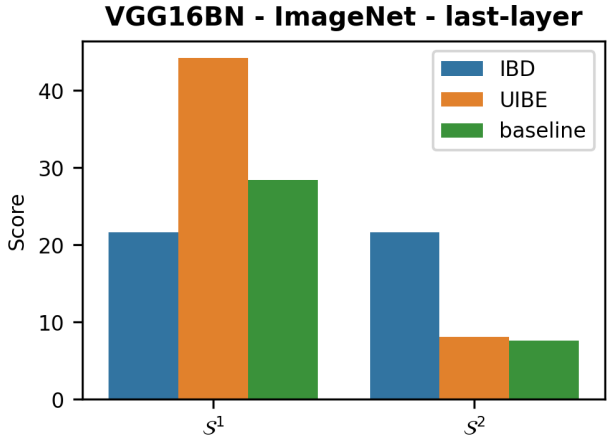

Fig. 18: Comparing basis interpretability of the proposed method (*UIBE*) with the natural feature space basis (*baseline*) and a basis extracted with a supervised approach (*IBD*) [8].

IBD. In other words, IBD is limited to learn a single direction for each one of the concepts, while the bases extracted with the proposed method may cover more than one direction for the same concept. Additionally, the sparsity criterion which we use to learn the interpretable basis, ensures that the different basis vectors cover different parts of the concept dataset. Another factor to consider for the same matter is the basis labeling procedure, which in our case is [11]. Other basis labeling strategies might suggest different labels which might also affect the interpretability scores. It is also noteworthy that the same possible explanation might be given regarding Fig. 18 where even the natural feature space basis scores better than IBD in terms of $\mathcal{S}^1$.

Regarding $\mathcal{S}^2$, the bases extracted with IBD may be considered significantly more interpretable than the bases extracted with the current work, with the latter being even more prominent in the case of ResNet18. We think that this fact is also linked to the previous argument. In particular, since a basis learned without supervision may have duplicate labels, the number of unique concept labels that can be attributed to the basis vectors (which is related to what $\mathcal{S}^2$ measures), is expected to be less than the number of vectors in the basis. However, for a basis learned in a supervised way, this number is exactly equal to the number of vectors in the basis (I). Moreover, in this case, IBD used a basis with a larger number of vectors ($I = 660$) compared to the proposed method (which only uses $I = D = 512$).

Overall, we find it difficult to strictly position the proposed method in relation to a supervised approach for this problem. We think that the current work reveals a possible limitation of the supervised approach which assumes that concept representations lie only on a single direction of the feature space. The proposed method has the potential to overcome this limitation. However, the presented experimental results also suggest that the previously mentioned strength of the proposed method is also its limitation. By devoting more than one basis direction to a single concept, inevitably limits the number of different unique concepts that can described by the basis. A possible direction towards improvement might be to consider an *approximately* orthogonal and over-complete basis of the feature space (i.e. $I > D$), which we leave for future work.

D. Qualitative comparisons

In this section we provide qualitative results which highlight the interpretability improvement gains that are obtained when we transform image feature representations to a basis learned with the proposed method. Thus, Fig. 19 and 20 depict results for ResNet18, Fig. 21 and 22 for VGG16BN, Fig. 23 and 24 for AlexNet and Fig. 25 and 26 for GoogleNet. In those figures, we used [11] to assign concept labels for the bases vectors extracted by our method, as well as to the vectors of the natural feature space bases. Among the group of common concepts that have been assigned to the concept detectors of the two bases, we considered the top-performing concept detector in each basis. The common name of the concept detectors is given on the (sub-)figure's top. The basis name that each concept detector comes from, is written on the left. For each selected concept detector, we present a row of

TABLE II: Statistics of Pairwise Vector Angles for the bases that we learned with the supervised approach of IBD [8].

	Pairwise Vector Angles (deg)			
Network	**Mean**	**Std**	**Min**	**Max**
ResNet18	85.67	3.77	45.5	99.5
VGG16BN	88.23	2.84	57.71	100.0

images whose representations have a spatial element which is ranked among the top-4 activations over the validation split of the concept dataset ($\mathcal{K}_{val}$). The reported IoU scores which are given below the set of images, corresponds to the IoU performance of the respective concept detector over the whole set of images in the validation set of the concept dataset. Each figure is meant to be read as a 2×2 grid of 4 concepts with each cell containing 2×4 images.

E. Are the bases learned with a supervised approach orthogonal ?

In this last section of experimental results, we experimentally seek to validate our hypothesis that an interpretable basis should be orthogonal. While our hypothesis is based on the assumption that the CNN has linearly disentangled concept representations, we still try to, at least partially, answer to what extend this is already happening when we use a supervised method to learn an interpretable basis. Building on our previous experimental results, we consider the bases that we learned with IBD [8] for the *last-layers* of ResNet18 and VGG16BN. We provide statistical measurements for the distribution of angles between basis vectors that are met in those bases. The results are depicted in Table II. It is noteworthy to mention that those bases have $I = 660$ and $D = 512$, with the important relation that $I > D$. Based on our measurements, the bases could be considered approximately orthogonal since the mean angle between any pairs of basis vectors is around $86.5°$ and the standard deviation of the distribution is less than $3.77°$, in the worst case. This fact could further support our intuition that interpretable bases shall be orthogonal. While the present work considers only $I \leq D$, future extensions could study the case where $I \geq D$ with approximate orthogonality constraints.

Fig. 19: Qualitative results for ResNet18.

Fig. 20: Qualitative results for ResNet18.

VIII. Conclusion

We presented an unsupervised, post-hoc method to extract an interpretable basis for a CNN's intermediate layer feature space. Based on current literature, we also proposed two metrics that can be used to measure a basis for its interpretability. We evaluated the effectiveness of the proposed method in standard CNN architectures and demonstrated that intermediate layer representations become more interpretable when projected onto bases extracted with our method. Finally, using the proposed metrics, we compared the outcomes of our method with the outcomes of a method that derives an interpretable basis using supervision. According to the interpretability metrics, the bases extracted with the proposed method, in one aspect, show appreciable interpretability improvements over the bases extracted with the supervised approach. At the same time, in a second aspect, the bases derived with supervision, were significantly more interpretable than the bases that were suggested by our method. This fact might seem peculiar at first. However, a possible explanation was provided and directions for future research for deeper understanding were suggested. We hope that the present work has contributed additional knowledge to interpretable basis extraction and motivates further research for understanding *black-box* models.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv:1409.4842 [cs]*, Sep 2014, arXiv: 1409.4842.

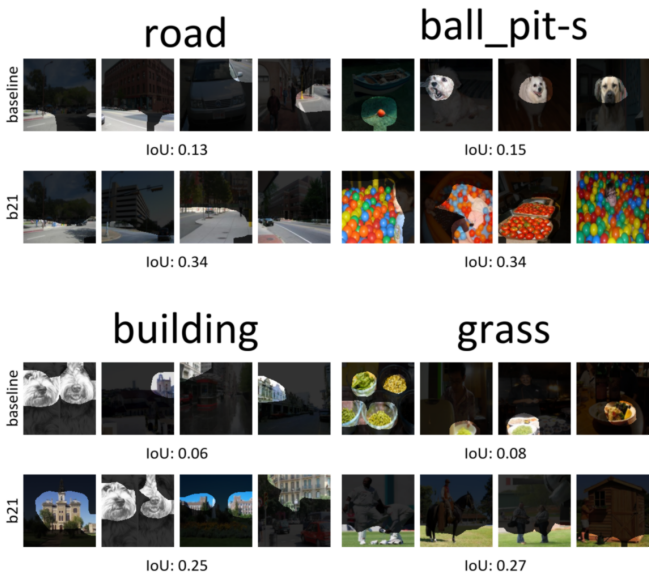

Fig. 21: Qualitative results for VGG16BN.

Fig. 23: Qualitative results for AlexNet.

Fig. 22: Qualitative results for VGG16BN.

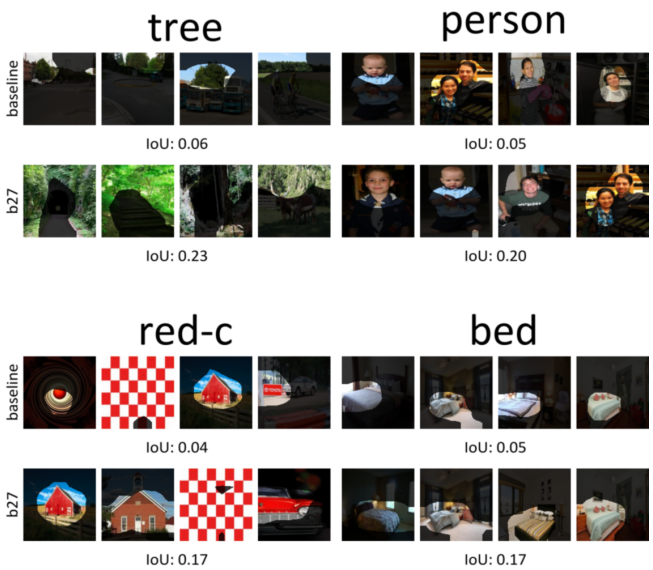

Fig. 24: Qualitative results for AlexNet.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[6] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[7] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*. PMLR, 2018, pp. 2668–2677.

[8] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *ECCV*, 2018, p. 119–134.

[9] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *CVPR*, 2018, pp. 8730–8738.

[10] J. Pfau, A. T. Young, J. Wei, M. L. Wei, and M. J. Keiser, "Robust semantic interpretability: Revisiting concept activation vectors," in *Fifth Annual Workshop on Human Interpretability in Machine Learning (WHI), ICML 2020*, 2020.

[11] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *CVPR*, 2017, pp. 6541–6549.

[12] J. Mu and J. Andreas, "Compositional explanations of neurons," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, p. 17153–17163.

[13] A. Ghorbani, J. Wexler, J. Zou, and B. Kim, "Towards automatic concept-based explanations," *arXiv preprint arXiv:1902.03129*, 2019.

[14] Q. Zhang, Y. N. Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8827–8836.

[15] H. Liang, Z. Ouyang, Y. Zeng, H. Su, Z. He, S.-T. Xia, J. Zhu, and B. Zhang, "Training interpretable convolutional neural networks by differentiating class-specific filters," in *ECCV*. Springer, 2020, pp. 622–638.

[16] M. Losch, M. Fritz, and B. Schiele, "Semantic bottlenecks: Quantifying and improving inspectability of deep representations," *International Journal of Computer Vision*, vol. 129, no. 11, p. 3136–3153, Nov 2021.

[17] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual

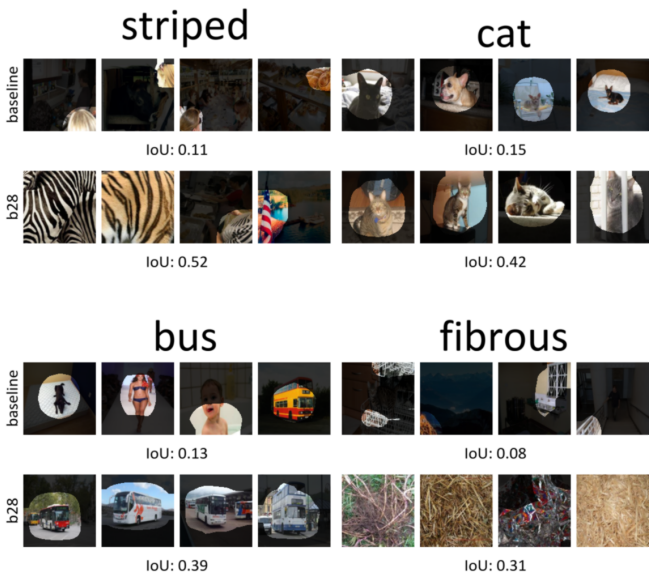

Fig. 25: Qualitative results for GoogleNet.

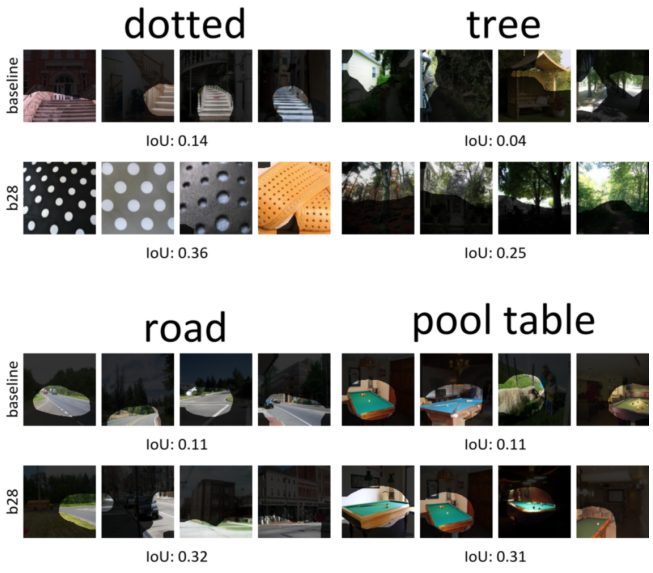

Fig. 26: Qualitative results for GoogleNet.

parsing for scene understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 418–434.

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[19] R. Achtibat, M. Dreyer, I. Eisenbraun, S. Bosse, T. Wiegand, W. Samek, and S. Lapuschkin, "From attribution maps to human-understandable explanations through concept relevance propagation," *Nature Machine Intelligence*, vol. 5, no. 9, pp. 1006–1019, 2023.

[20] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[21] J. Vielhaben, S. Blücher, and N. Strodthoff, "Multi-dimensional concept discovery (mcd): A unifying framework with completeness guarantees," *arXiv preprint arXiv:2301.11911*, 2023.

[22] P. Chormai, J. Herrmann, K.-R. Müller, and G. Montavon, "Disentangled explanations of neural network predictions by finding relevant subspaces," *arXiv preprint arXiv:2212.14855*, 2022.

[23] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint arXiv:1812.02230*, 2018.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[25] M. Lezcano-Casado, "Trivializations for gradient-based optimization on manifolds," in *Advances in Neural Information Processing Systems, NeurIPS*, 2019, pp. 9154–9164.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] L. Whyte, "Unique arrangements of points on a sphere," *The American Mathematical Monthly*, vol. 59, no. 9, pp. 606–611, 1952.

[29] P. M. L. Tammes, "On the origin of number and arrangement of the places of exit on the surface of pollen-grains," *Recueil des travaux botaniques néerlandais*, vol. 27, no. 1, pp. 1–84, 1930.

[30] D. Kottwitz, "The densest packing of equal circles on a sphere," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 47, no. 3, pp. 158–165, 1991.

[31] J. Wang, "Finding and investigating exact spherical codes," *Experimental Mathematics*, vol. 18, no. 2, pp. 249–256, 2009.

[32] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *IEEE TPAMI*, vol. 41, no. 9, pp. 2131–2145, 2018.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[35] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

Alexandros Doumanoglou holds the diploma of Electrical and Computer Engineer from the Aristotle University of Thessaloniki, and joined the Information Technologies Institute in 2012. Currently he does his PhD in Explainable Artificial Intelligence at the department of Advanced Computing Sciences at Maastricht University, under the supervision of Prof. Stylianos Asteriadis. His current research focus is on unsupervised learning, and explainable and interpretable methods for deep learning models.

Stylianos Asteriadis holds the diploma of Electrical and Computer Engineer from the Aristotle University of Thessaloniki, A.U.Th, MSc in Digital Media from School of Informatics of the same University and PhD in Electrical and Computer Engineering from the National Technical University of Athens, (NTUA). He is Associate Professor at the Department of Advanced Computing Sciences at Maastricht University. He coordinates the Cognitive Systems Research Group and teaches HCI, Affective Computing, AI, and Computer Vision.

Dimitrios Zarpalas holds the diploma of Electrical and Computer Engineer from Aristotle University of Thessaloniki, A.U.Th, MSc in Electrical Engineering from The Pennsylvania State University, and PhD in medical informatics (Health Science School, department of Medicine, A.U.Th). He joined the Information Technologies Institute in 2007, and is currently a researcher, grade B. His research interests include tele-immersion applications, 3D computer vision, 3D object recognition, and motion capturing.