Explaining the Impact of Training on Vision Models via Activation Clustering

Anonymous CVPR submission

Paper ID 13609

Abstract

Recent developments in the field of explainable artificial intelligence (XAI) for vision models investigate the information extracted by their feature encoder. We contribute to this effort and propose Neuro-Activated Vision Explanations (NAVE), which extracts the information captured by the encoder by clustering the feature activations of the frozen network to be explained. The method does not aim to explain the model's prediction but to answer questions such as which parts of the image are processed similarly or which information is kept in deeper layers. Experimentally, we leverage NAVE to show that the training dataset and the level of supervision affect which concepts are captured. In addition, our method reveals the impact of registers on vision transformers (ViT) and the information saturation caused by the watermark Clever Hans effect in the training set.

1. Introduction

The development of explainable AI (XAI) has accompanied the emergence of regulations regarding the use of machine learning models, especially for safety-critical applications such as healthcare [6] or surveillance [41, 56]. A branch of this research field investigates the use of concept-based models to control and gain insights into what and how a model learns. The idea is to link the predictions either a priori [57, 60], through training [10], or a posteriori [24] to a set of concepts interpretable by the user. Usually, the concepts are represented by vectors in the embedding space, which are then used to compute the predictions. The idea is to explain the predictions using sentences such as: "*The model predicts this because of that*". Deciding on a priori concepts raises the question of their relevance with respect to the model's internal operations [26]. As for the learned concepts, it is difficult to determine the characteristics that justify their distinctiveness [23].

Convolutional neural networks store and extract semantics, which is the foundation of pyramid networks developed for tasks such as supervised and unsupervised object detection and semantic segmentation [29, 36, 53]. Caron et al. [8] and Oquab et al. [40] confirm that self-supervised vision transformers (ViT) [17] share this property. Building upon this property, recent work in XAI have investigated the unsupervised extraction of such concepts for networks trained for classification and linked them to the prediction using attribution/saliency methods [5, 9, 47, 52]. Chormai et al. [12] decompose a Layer-wise Relevance Propagation (LRP) [5] message through a convolution neural network at a given depth by decomposing the message onto orthogonal spaces. By restricting the message, a single direction focuses the attribution map on a subset of the semantic captured by LRP. Kauffmann et al. [30] instead cluster the feature activations at a certain depth using a specific neuralized clustering layer compatible with LRP. The method reveals that while early layers act primarily as edge detectors, deep layers extract more complex semantics of the input, such as objects.

This paper continues this line of research and proposes to simplify the extraction method as well as to visualize concepts in the form of segmentation instead of heat maps, thus circumventing the drawbacks and criticisms associated with saliency methods [13, 16]. The rationale is that parts of the input deemed similar by the network should produce similar output by the hidden layers and that features of the input deemed relevant for the prediction are preserved in the deeper layers. Our method, Neuro-Activated Vision Explanations (NAVE), clusters a combination of feature activations of an image classifier taken at different depths. NAVE outputs an explanation of the feature activations in the form of a segmentation of the input image. Here again, while explanations of shallow layers mostly reveal their edge detection function, those of deeper layers align with the semantics of the input.

Note that NAVE is unsupervised and does not require any further fine-tuning of the network. Moreover, although our method produces unsupervised segmentations, and we evaluate the semantic consistency of its segments (Section 4.3), it should not be mistaken for a semantic segmentation method. Instead, NAVE is to be considered a post-hoc explanation method for the encoder of vision models. Its output depends on the training dataset (Section 4.1) and the training scheme (Section 4.4.2). Besides, it can be used to study the role of

CVPR #13609

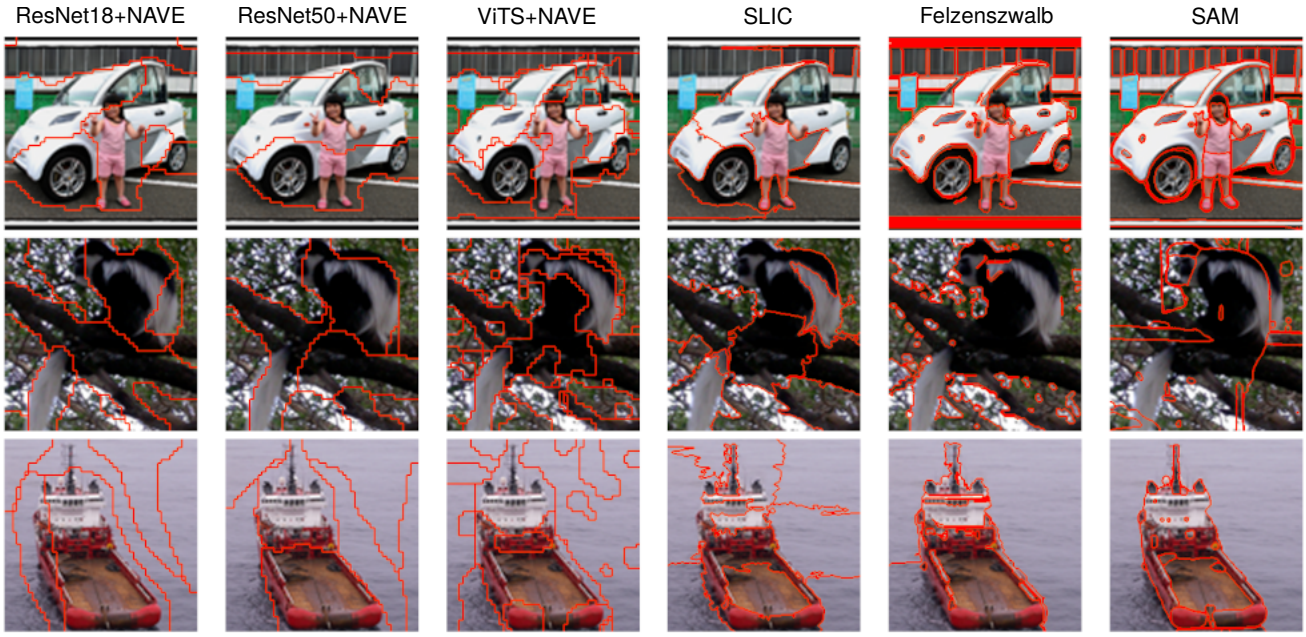

Figure 1. Examples of segmentations of three STL-10 images adjusted to return a similar number of regions. The proposed NAVE algorithm does not miss the monkey's tail (second row) and does not artificially split the sea around the ship (third row). If the cells including the person (first row) follow her shape, they also include both wheels. The multiple edges fool model-agnostic SLIC [1] and Felzenszwalb [22] algorithms. The Segment Anything Model [32] isolates objects and smaller parts, but does not necessarily focus on the same semantics as more general feature encoders.

registers in ViTs (Section 4.5.2) and the information saturation of the encoder due to watermarks/Clever Hans in the training set (Section 4.5.3).

Our contributions are as follows:

(i) We propose Neuro-Activated Vision Explanations (NAVE) as a method to explain encoders of vision models and demonstrate that NAVE captures the semantics in the representations extracted by vision models.

ii) Using NAVE, we can evaluate the impact of training vision models, which we evaluate, for example, via object localization. Specifically, we confirm that recent advances in self-supervised learning close the gap to supervised learning.

(iii) Besides, we provide a new way to evaluate the effect of registers in ViTs and confirm its effectiveness on ViT-small but observe no significant improvement in ViT-base.

(iv) Surprisingly, a ViT-base model trained on ImageNet1k performs worse than a ViT-small model for capturing objects when evaluated on three datasets.

(v) Finally, we leverage NAVE explanations to show the consequences of augmentation and Clever Hans during training in the capabilities of the model to parse the input.

2. Related Work

Saliency methods hold an important role in the XAI toolkit owing to the problem they aim to solve: to unravel the influence of the input's features on the model's predictions. For image classifiers, the task is to highlight the most salient pixels, typically visualized as heatmaps. Model-agnostic saliency methods estimate the sensitivity of the predictions to the perturbations of certain features of the input [44, 45, 55, 59]. Model-aware methods, on the other hand, utilize the parameters of the model to compute the explanations. The pixels importance derives from the amount of information backpropagated from the prediction that reaches them [5, 47, 51, 52]. In both cases, the highlighted areas are not necessarily semantically consistent, meaningful, or relevant.

Another line of research in XAI assumes that models' decision-making can involve several concepts with a human-friendly interpretation. These concept-based models (CBM) aim to project user-defined examples to the embedding and use them to explain the prediction [31]. Self-explainable models, or prototypical models, follow a less supervised approach and learn the set of concepts either in the embedding space [2] or in the activation space of the last block [10]. The drawback of learning a so-called *transparent* classifier with prototypes is that although the performance might match an *opaque* model, the learning is different and so is the embedding and, consequently, how the learned concepts interact [24].

Be it concepts or prototypes, the question of whether the human-understandable concepts align with the models

arises. DISSEC [60] or CRAFT [21] aim to build that bridge by evaluating the model's representation in the light of a selection of examples. More specific to prototypical models, Gautam et al. [23] leverage LRP to unravel the characteristics common to the input and each prototype.

Inspecting activations and other intermediate outputs became particularly important with the advent of vision transformers. The seminal work of Caron et al. [8] shows that the self-attention of the class token of self-supervised ViT carried information about the input's semantics. A family of object localization methods [49, 54] builds upon that result. This work was pursued and brought to a higher level of complexity in Oquab et al. [40], where the authors show that field-of-depth can also be recovered from attention and activations. For convolutional neural networks, several works have investigated the presence and extraction of semantics in the activations [18, 50], revealing that deeper convolutional layers act less like edge detectors and capture higher-level concepts, like objects. Chormai et al. [12] learn an SVD-like decomposition of an LRP message passing through a convolution neural network. By restricting the message to pass through only one of the subspaces, the attribution map focuses on a subset of the semantics captured by LRP. Closer to our work, Kauffmann et al. [30] propose to cluster the activations at a given depth with an LRP-friendly module. Backpropagating the cluster assignments reveals that each is activated by semantically consistent parts of the input.

3. Neuro-Activated Vision Explanations

We now introduce our Neuro-Activated Vision Explanations (NAVE) method, which produces a segmentation of the input aligned with the semantics captured by the model. Figure 1 shows NAVE explanations for various vision architectures and compares them with that of the model-agnostic SLIC [1] and Felzenszwalb [22] segmentation algorithms. We also show the segmentation obtained from the foundation model Segment Anything [32].

The idea of NAVE is as follows. Let us consider a deep neural network f trained, e.g., for the classification of images of dimension (H, W, C) into Q classes, i.e., $f : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^Q$. Then, the architecture of f is usually a sequence of $L > 0$ convolutional layers $\{l_i : \mathbb{R}^{h_i\times w_i\times c_i} \to \mathbb{R}^{h_{i+1}\times w_{i+1}\times c_{i+1}}\}_{1\leq i\leq L}$, followed by a multi-layer perceptron as a classifier, $p : \mathbb{R}^{h_{L+1}\times w_{L+1}\times c_{L+1}} \to \mathbb{R}^Q$. The operations of f on an image $x \in \mathbb{R}^{H\times W\times C}$ can thus be decomposed as follows: $f(x) = p \circ l_L \circ \ldots \circ l_1(x)$. In practice, l_i can involve more than one convolution operation, as well as nonlinear activations, batch normalization, etc. For example, in the case of a ResNet and a ViT, it can be a residual block or a transformer block, respectively.

The operations of NAVE are described in a Python-inspired pseudocode in Algorithm 1[1]. In addition, Figure 2 depicts a schematic representation of our method. For the sake of legibility, we extract all feature activations, but as we shall see later, models such as a ViT only require their last layer activations to be taken.

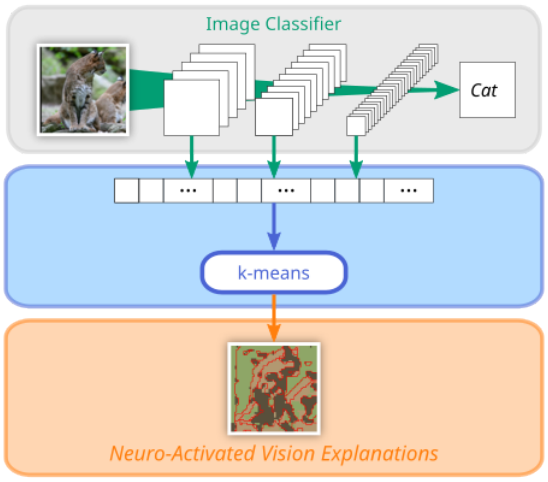

Figure 2. A schematic representation of our NAVE method.

Algorithm 1 NAVE Python pseudocode

```
Require: input image x of dim. (H,W,C),
   number of clusters K
1: act = []                          ▷ activations
2: for i = 1 to L do
3:    x = l_i(x)          ▷ act. of ith layer
4:    u = bicubic_upsample(x, (H,W))
5:    u = Reshape(u, (H*W,c_{i+1}))
6:    u = u / norm(u, ord=2, axis=1)
7:    u = u / (1 + c_{i+1})
8:    act.append(u)
9: end for
10: act = concatenate(act, axis=1)
11: expl = Cluster(K).fit_predict(act)
12: return expl
```

Choosing `(H, W)` as output dimensions will compromise between a higher resolution and a shorter computation time. Note that in line 4, the upsampling uses a bicubic interpolation. To reduce artifacts, we recommend matching the resolution of the first extracted layer. Likewise, if the final segmentation needs to be upsampled, we recommend using a nearest-neighbor interpolation.

By default, `Cluster` is k-means [38] for its speed and simplicity, at the cost of fluctuating results depending on the random initialization. In our experiments, however, we did not notice significant variance in multiple runs of the same

[1] A public implementation will be provided upon acceptance.

setting with different initializations of k-means. Another option is deterministic hierarchical clustering [27], which produces stable results but is expensive to compute.

The input can be a single image or a set of images, in which case, the clusters are shared among the images and thus identify similarities between the images.

4. Experimental Evaluations

In this section, we perform a variety of qualitative and quantitative experiments to demonstrate that our NAVE method extracts meaningful and relevant semantics learned by vision models. Specifically, we systematically evaluate the captured semantics, influence of chosen training sets and schemes, architecture choices, augmentations, and watermark / Clever Hans [34] effects. We start with an overview of all datasets, models, training schemes, and baselines used.

4.1. Experimental Setting

Datasets. For a proxy object localization task, we use VOC07 (training + validation subsets) [19], VOC12 (training + validation subsets) [20], and COCO 20k (training subset) [35]. Furthermore, we consider Chest-X-RAY [3], STL-10 [14], and ImageNet1K [46] to evaluate the influence of the data used to train the models.

Vision Models. As architectures for the vision models, we use vision transformers, specifically ViT-Small (ViTS) and ViT-Base (ViTB) with patch sizes of 14×14 and 16×16 [17], and residual networks, i.e., ResNet18 and ResNet50 [28].

Training Schemes. The vision models we consider can be trained in various ways. We consider randomly initialized models, self-supervised pretraining as in DINO [8] and DINOv2 [40], and supervised classifier pretraining on ImageNet1K [46].

Baselines. For object localization, we consider SLIC [1], LOST [49], TokenCut [54] and the model by Lv et al. [39] as baselines.

Implementation. We implement our experimental evaluation in Python. For k-means, we use the sklearn implementation [43] and initialize with k-means++ [4]. The evaluation of object localization is based on code from LOST [49]. The DINO and DINOv2 models are the same provided by the respective authors, while all our models trained on ImageNet1k are the ones provided by the PyTorch [42] library.

Hardware. All experiments run on a dual AMD Epyc machine with 2×64 cores with 2.25 GHz, 2 TiB of memory, and NVIDIA A100 GPUs with 80 GB memory.

4.2. Capturing Objects with NAVE

Quantifying the meaningfulness of the captured concepts with NAVE is a challenging endeavor. Pragmatically, some form of correspondence between the captured groups and real concepts present in the image should indicate that they are sound. We propose using *object extraction* (instead of localization or segmentation) as a proxy task to assess the capabilities of NAVE in extracting semantics, i.e., how well segmentations from NAVE isolate objects.

For each of the aforementioned datasets, we select a segment instead of performing the full localization, bypassing the original task and focusing on intersecting the clustered activations with object annotations. We follow two strategies: (i) select the segment with the largest intersection-over-union (IoU) between ground truth boxes and its *outer* bounding box; (ii) select the segment with the largest IoU between ground truth boxes and its *inner* bounding box. These strategies always select a segment whose *outer* bounding box intersects one of the true bounding boxes. Hence, the reported scores are presumably larger than those achieved by proper object localization. Yet, recall that this experiment serves as a proxy to assess the capacity of NAVE to identify if and which objects the network identifies. Further details on the strategies and examples are discussed in the supplementary material.

Metric. Following the protocol used in [49], we report the Average Precision at 50% (AP@50%), also known as the *Correct Localization* metric, which is the frequency of the predicted boxes with an IoU score larger than 50% with at least one ground truth bounding box.

The base model is a visual transformer trained using either DINO or DINOv2. NAVE processes the output of the last transformer block (without the attention) and relies on k-means with $K = 5$ clusters. The results are summarized in Table 1. We first report on the performance of four state-of-the-art baselines for object localization. In order to isolate the role of the segment selection strategies, we also report the performance of SLIC segmentation combined with the same selection strategies. The strategies are indicated in parentheses, e.g., outer-box or inner-box.

Table 1. Object localization performance on VOC07, VOC12, and COCO20k measured using AP@50%. Numbers from cited work are taken from the respective references.

Method	Feature	VOC07	VOC12	COCO20k
DINO-seg [8]	DINO-ViTS/16	45.8	46.2	42.1
LOST [49]	DINO-ViTS/16	61.9	64.0	50.7
TokenCut [54]	DINO-ViTS/16	68.8	72.1	58.8
Lv et al. [39]	DINO-ViTS/16	70.6	72.1	63.5
SLIC (outer-box)	-	53.3	57.1	47.2
SLIC (inner-box)	-	42.2	46.0	38.5
NAVE (outer-box)	DINO-ViTS/16	74.3	76.1	68.4
NAVE (inner-box)	DINO-ViTS/16	62.7	62.3	61.0
NAVE (outer-box)	DINOv2-ViTS/14	73.0	74.2	71.0
NAVE (inner-box)	DINOv2-ViTS/14	66.0	67.1	66.2

CVPR
#13609

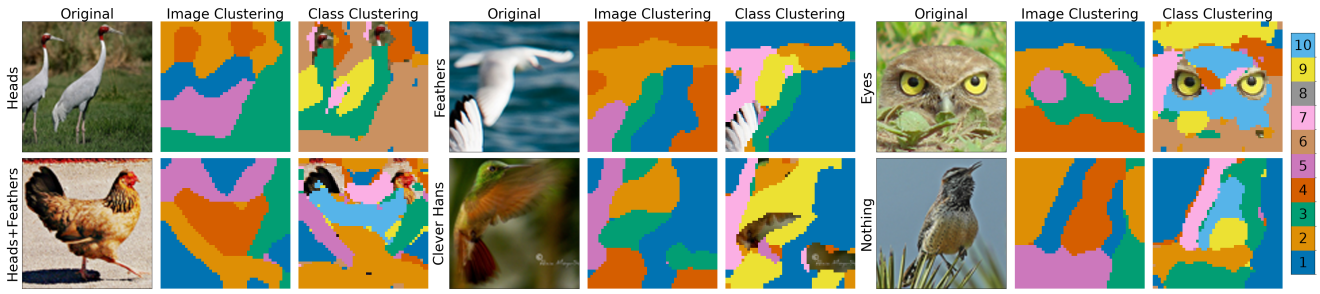

Figure 3. Clusterings learned on a single image or on all images of the bird class are consistent with each other. Class-cluster 8 captures visually different but specific features of the birds, indicating that the model recognizes them and processes them similarly.

These figures indicate that more than 60% of the time, at least one segment has an IoU larger than 50% with at least one object captured by NAVE. Note that the scores of NAVE are on par with the state of the art, even though it may partition the objects into several segments (see Figure 1). The gap in performance between NAVE and SLIC relativizes the advantage provided by the segment selection strategies. Overall, these results suggest NAVE captures semantically meaningful concepts present in input images.

4.3. NAVE Captures Semantics

Next, we investigate how NAVE captures relevant class information. An evaluation based on the bird class of STL-10 serves as qualitative evidence for this.

We start by training a ResNet18-based image classifier on STL-10. Then, we extract the feature activations of the last three residual blocks and cluster the activations with a hierarchical clustering and Ward linkage. We consider two cases. Image clustering, where the clustering is performed per image with $K = 5$ clusters, and class clustering with $K = 10$ clusters, where the clustering is learned based on the feature activations of all images of that class. We apply it to all the training images of the bird class of STL-10, which are all correctly classified. The results are depicted in Figure 3 for six selected bird images, with more examples provided in the supplementary material.

It is remarkable how similar the image-wise and class-wise clusterings are. For all the selected examples, the unmasked Cluster 8 of the class clusterings always has a counterpart in the image clusterings. Since the class clusterings assigned the same labels to similar concepts captured by the network in different images, its interpretation is more accessible. For example, the background is assigned Cluster 1 (dark blue) or Cluster 2 (orange).

We uncovered Cluster 8 as it captures specific parts of the birds: head, tail, long feathers, eyes, but also Clever Hans [34] like the watermark (second row, second bird). Although this cluster is not present in all bird images, the fact that visually and semantically different parts of the birds are clustered together suggests that the model identified them as high-level concepts related to the bird class. Conversely, the fact that NAVE groups together feature activations related to these parts indicates that our method is able to extract the semantics captured by the network.

4.4. NAVE Uncovers the Influence of Training

The following set of experiments investigates the characteristics of training a vision model. Specifically, we focus on the choice of the training set and training scheme. NAVE allows us to compare the embeddings of models trained on task-relevant and task-irrelevant datasets as well as of no training scheme (random initialization), supervised, and self-supervised training schemes.

4.4.1 Choice of the Training Set

In this experiment, we investigate the role of the training dataset into the concepts extracted by the image encoder. The hypothesis is that if a model has never seen certain concepts, such as a person, it is unlikely to be able to extract/recognize it and it will not be distinctively visible in the output of NAVE.

Using the same object extraction evaluation from the previous section, Table 2 shows the AP@50% of NAVE where we test this hypothesis. These experiments use a ViTS/16-based classifier: untrained (random initialization), then trained either on ImageNet1K, STL-10, or Chest-X-RAY. For comparison, we recall the performance of DINO and SLIC.

Training on ImageNet1K consistently yields the best AP@50% for all datasets. It even outperforms an image encoder pretrained with DINO. If the classifier is trained on a small (STL-10) or out-of-domain dataset (Chest-X-RAY), the scores are on par with an untrained model and SLIC.

We illustrate the influence of the training dataset using Figure 4. The original image is extracted from the STL-10 data (training) and depicts a person in front of a car. While STL-10 and ImageNet1K datasets have a “car” class, only the latter has a “person” class. To visualize how this affects NAVE, we compute twenty explanations for a ResNet18,

Table 2. ViTS/16 AP@50% performance on VOC07, VOC12, and COCO20k. Focus on the training dataset. All results are for *inner-box* (see Section 4.2). DINO and SLIC are reported for reference values.

Training Dataset	VOC07	VOC12	COCO20k
ImageNet1K	**68.6**	**69.5**	**64.1**
Random Initialization	45.2	46.9	48.3
STL-10	44.2	44.7	51.3
Chest-X-RAY	42.8	45.1	42.7
DINO	62.7	62.3	61.0
SLIC	43.1	46.0	38.5

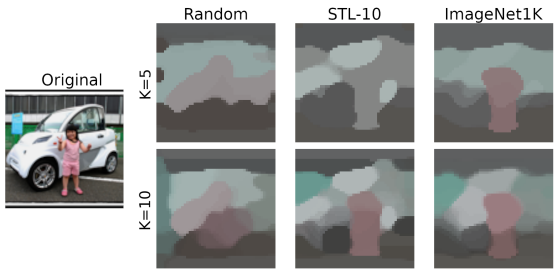

Figure 4. Average pixel color of 20 NAVE explanations extracted from a ResNet18 either (a) untrained (random initialization) or trained on (b) STL-10 or (c) ImageNet1K.

either untrained (random initialization), or trained on STL-10 or ImageNet1K. For each explanation, we compute the average color of each segment and plot in Figure 4 the average color of each pixel over the twenty runs. If the person is often captured, a pink shade following their silhouette should be visible.

First, the person does not appear in images produced by a randomly initialized ResNet18. Second, for the ResNet18 trained on ImageNet1K, a human pink shape stands for $K = 5$ and 10 clusters. For a ResNet18 trained on STL-10, the pink silhouette is visible only for $K = 10$, although for $K = 5$, that silhouette blends in with the objects behind.

Our interpretation is that the model trained on ImageNet1K learned to produce activations that react to a person, and NAVE can, therefore, extract it. The fact the silhouette is present for STL-10 and ImageNet1K with $K = 10$ but not for $K = 5$ suggests that the model does separate objects, but more specific concepts such as a person are not learned when trained on less data.

4.4.2 Choice of the Training Scheme

Using NAVE, we now show how different training schemes produce models with clearly different behavior. To do so, we compare no training (random initialization), supervised training (image classifier on ImageNet1K), and self-supervised training (DINO and DINOv2 embeddings) using a ViTS. Table 3 (top) shows the results.

Table 3. AP@50% performance on VOC07, VOC12, and COCO20k. We only depict results for NAVE (inner-box). **Top:** Focus on training scheme: DINO, DINOv2, pre-trained on ImageNet, randomly initialized. **Middle:** Focus on architecture. **Bottom:** Focus on ViT registers.

Feature	VOC07	VOC12	COCO20k
Training Scheme			
ViTS/16 Random Init.	45.2	46.9	48.0
ViTS/16 ImageNet	**68.7**	**69.5**	64.1
DINO-ViTS/16	62.7	62.3	60.6
DINOv2-ViTS/14	66.0	66.2	**65.1**
Architecture			
ViTS/16 ImageNet	**68.7**	**69.5**	**64.1**
ViTB/16 ImageNet	44.1	45.3	48.6
ResNet50 ImageNet	52.1	51.7	57.3
Registers			
DINOv2-ViTS/14	66.0	**67.1**	**66.2**
DINOv2-ViTS/14 + reg	**66.5**	66.4	65.7
DINOv2-ViTB/14	63.8	64.0	≈ 64
DINOv2-ViTB/14 + reg	65.7	66.8	≈ 65.4

The random model serves as a baseline for how far different training schemes can take the model. However, supervised and self-supervised learning schemes are not clearly superior to one another. Remarkably, while requiring no labels, DINOv2 achieves very similar performance to the fully supervised model. The drawback is that it is trained using considerable resources and requires a much larger set of images. DINO, on the other hand, is outperformed by both schemes.

4.5. Model Inspection with NAVE

With the next set of experiments, we aim to demonstrate that NAVE consistently shows which models are more adequate at capturing relevant information from their inputs. We start with architectural choices and then consider additional transformations on data, such as augmentation or watermarks.

4.5.1 Architecture

High classification performance often identifies good image encoders, but NAVE provides an alternative way to evaluate this. Here, models trained on ImageNet1K serve as a basis for evaluation. As architectures, we consider ViTS, ViTB, and a ResNet50 and report AP@50% in Table 3 (middle).

Surprisingly, ViTB performs poorly despite being the largest model. Our hypothesis is that the ViTB model is too large for the task, allowing it to eventually learn spurious concepts that are not necessarily meaningful for other tasks and make the relevant information challenging to extract for NAVE. Note that since our quantitative evaluation is

CVPR
#13609

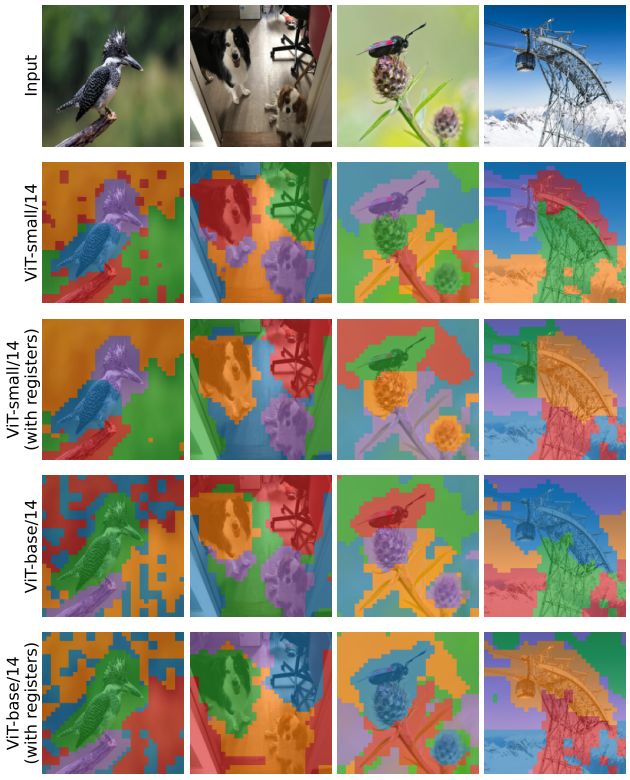

Figure 5. NAVE (with $K = 5$ clusters) reveals differing activations by ViTs trained with DINOv2 (with and without registers [15]) on four sample images taken from [15, 40].

designed to correlate captured concepts with real objects annotated in images, a model that extracts noisy or too complex concepts should perform poorly or similarly to a randomly initialized one. On the other hand, the ViTS significantly outperforms a ResNet50, despite both being similar in parameter budget. The better inductive biases afforded by the architectural choices ViTS allow it to learn more generalizable concepts, which translates into better scores in our evaluation.

4.5.2 ViT and Registers

Recent works on training vision transformers characterize certain issues with learned representations both in the supervised and self-supervised training regimes. Darcet et al. [15] study how individual tokens can present very different norms from others in the same layer, not correlated with the original inputs in any particular way. We show that NAVE can be used to investigate this behavior.

Figure 5 illustrates four individual cases selected to be similar to the cases presented in the original study. Here, we observe the registers do not necessarily address all issues with activations having significantly different norms, in particular with ViT-base/14 (bottom-left case). In some cases, however, there is an advantage, and the grouping of activations quite clearly corresponds to different concepts in the input image, such as the second column. As seen in Table 3 (bottom), adding registers results in different AP@50% scores, but not consistently better or worse.

Compared to the previous section, these experiments show that a ViTB trained with DINOv2 is able to extract representations that result in concepts that are more easily captured by NAVE. Given the larger training set and more general optimization goal used by the method, this expected behavior can be confirmed here as well. Surprisingly, just as the presence or not of registers, its performance is not significantly better than ViTS. Beyond what is already suggested in Figure 5, another explanation is that the internal handling of concepts by the model becomes too complex to be captured by NAVE, limiting its performance within our evaluation framework.

4.5.3 Augmentation and Clever Hans

Augmenting or perturbing the input helps to prevent overfitting [11, 48]. Besides, changing the color or the orientation discourages the model from creating shortcuts [25], for which the Chest-X-Ray dataset for COVID prediction [3] became a textbook example as the model was basing its predictions on luminance, sharpness, and other annotations. Leveraging these clues or biases in the dataset, also called Clever Hans [33], for the prediction is undesirable and often harms the generalization of the model [58]. Previously, several strategies have been studied to avoid them [7, 37].

In this section, we propose to visualize the effect of augmentations and Clever Hans on the feature activations. We base the experiment on the Chest-X-Ray dataset. Building upon [23], we train a ResNet50 for pneumonia prediction and restrict the dataset to a single hospital. We create a copy of the dataset where all the negative images are watermarked with a small black box positioned at random. Then, we consider four training strategies: with or without augmentation and with or without the watermarks. We select two negative images (no pneumonia), with and without a watermark, to illustrate in Figure 6 four distinct possible behaviors of the models uncovered using NAVE explanations. The explanations are computed over 200 images using the outputs of the last two residual blocks and $K = 10$.

For Figures 6.a and b, the models are trained on the original dataset, i.e., they did not see the added watermarks during training. The first observation is that using augmentation during training yields more complex explanations as the lungs are separated and split between their upper and lower parts. For the left example, adding the watermark (second and fourth rows) does not alter the NAVE explanation, while for the second example, it does. The change is more pronounced for the model trained with augmentation. We conjecture that the behavior shift depends on the watermark's position.

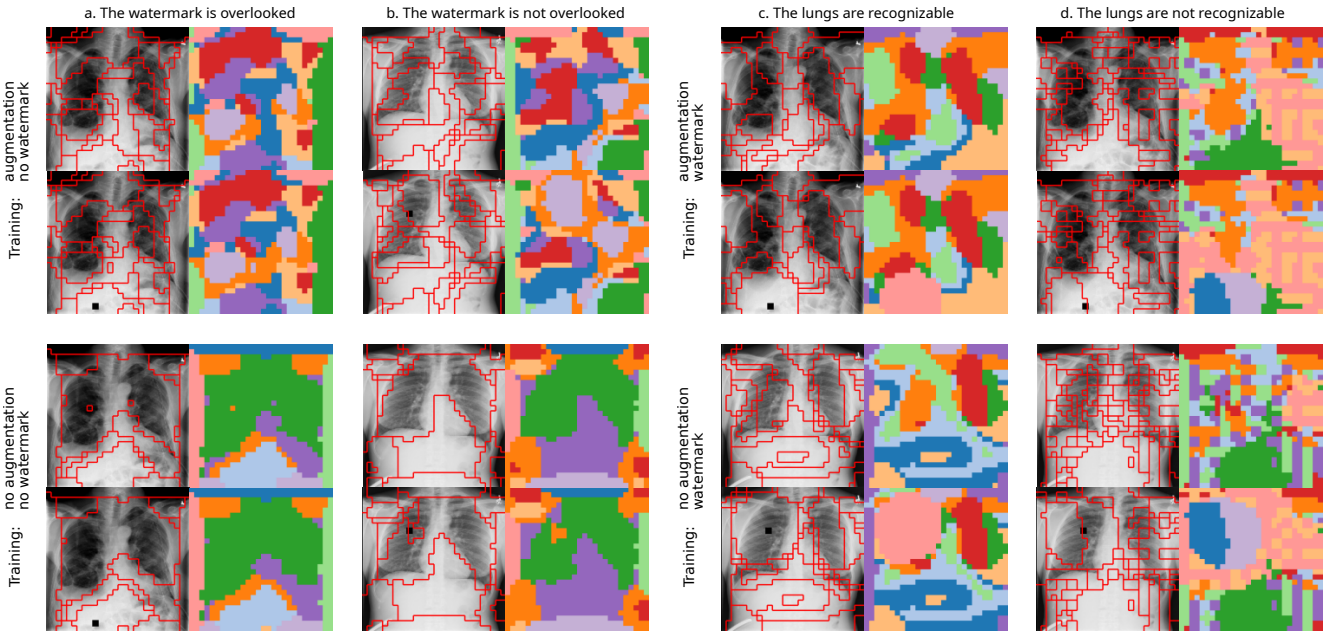

Figure 6. Grouped activations from NAVE on a ResNet50 model trained under various regimes of data augmentation and watermarking. Interestingly, when the watermark is detected, the features around it tend to produce disproportionately large groupings.

The models of Figures 6.c and d are trained on the watermarked dataset. In the cases where a black box is added, both models react strongly. Besides the fact that one model's explanation dedicates a single cluster while the other uses two to circle the watermark, the way the rest of the image is processed is striking. The model trained with a watermark and without augmentation (right columns) does not seem to detect any other organs except for the part of the abdomen below the lungs, which is present in all the images. This extreme behavior suggests that this last model, trained without augmentation but with watermarks, strongly overfits the watermarking. Note that both images contain an annotation in the top right corner, which all explanations seem to overlook.

4.6. Ablation Study

To study the effect of certain design choices and to set our proposed NAVE method into context, we perform an ablation study, which we defer to the supplementary material. In particular, we compare our activation clustering against PCA, as suggested by Oquab et al. [40]. Specifically, we evaluate the influence of the clustering algorithm and the number of clusters (or principal components) used. Finally, we only focused our results here using the more relevant inner box strategy for object-capturing objects. In the supplementary material, we also report results for the outer box strategy.

5. Conclusion

We introduced Neuro-Activated Vision Explanations (NAVE), a novel explanation method for feature activations of vision models. Employing it on class-wide activations, we demonstrated its capability to segregate semantically meaningful regions within an image. Quantitatively, we leveraged object localization as a proxy task, adopting object selection strategies that score good matches between extracted semantics captured by NAVE and real object annotations.

The primary limitation of our approach is the use of k-means clustering, for two reasons. First, it introduces some variability due to initialization. Second, its simplicity is beneficial from an explainability point-of-view, but suffers from limited expressibility, which can hinder its use with models that learn representations that are too complex. Although hierarchical clustering could be a potential solution, it comes with increased computational cost and interpretation complexity. We believe the trade-off imparted by k-means is thus more advantageous.

Despite the aforementioned limitations of k-means, the simplicity offered by NAVE allowed us to study the impact of various training aspects on vision models. We investigated how different sets of data used to train the same architecture led to clearer or less clear learned concepts. Furthermore, we studied the effects of different architectures trained on the same data, revealing that more complex models could also lead to more complex learned concepts, leading to poorer performance in our quantitative evaluation. Specifically on ViT-based models, we also inspected how NAVE revealed different levels of impact in the use of registers, showing more benefit on ViT-small than on ViT-base.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11): 2274–2282, 2012. 2, 3, 4

[2] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018. 2

[3] Julián D Arias-Londoño and Juan I Godino-Llorente. Analysis of the clever hans effect in covid-19 detection using chest x-ray images and bayesian deep learning. *Biomedical Signal Processing and Control*, 90:105831, 2024. 4, 7

[4] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. 4

[5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS ONE*, 10 (7):e0130140, 2015. 1, 2

[6] Katarzyna Borys, Yasmin Alyssa Schmitt, Meike Nauta, Christin Seifert, Nicole Krämer, Christoph M Friedrich, and Felix Nensa. Explainable ai in medical imaging: An overview for clinical practitioners–beyond saliency-based xai approaches. *European Journal of Radiology*, 162:110786, 2023. 1

[7] Kit M Bransby, Arian Beqiri, Woo-Jin Cho Kim, Jorge Oliveira, Agisilaos Chartsias, and Alberto Gomez. Backmix: Mitigating shortcut learning in echocardiography with minimal supervision. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 570–579. Springer, 2024. 7

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 3, 4

[9] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021. 1

[10] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 1, 2

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 7

[12] Pattarawat Chormai, Jan Herrmann, Klaus-Robert Müller, and Grégoire Montavon. Disentangled explanations of neural network predictions by finding relevant subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 3

[13] Benedict Clark, Rick Wilming, and Stefan Haufe. Xai-tris: non-linear image benchmarks to quantify false positive post-hoc attribution of feature importance. *Machine Learning*, 113 (9):6871–6910, 2024. 1

[14] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 4

[15] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. 7

[16] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems*, 2019. 1

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 4

[18] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. Technical report, University of Montreal, 2009. 3

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007. 4

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012. 4

[21] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2711–2721, 2023. 3

[22] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004. 2, 3

[23] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2022. 1, 3, 7

[24] Srishti Gautam, Ahcene Boubekki, Marina MC Höhne, and Michael Kampffmeyer. Prototypical self-explainable models without re-training. *Transactions on Machine Learning Research*, 2024. 1, 2

[25] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 7

[26] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019. 1

[27] Allan D Gordon. A review of hierarchical classification. *Journal of the Royal Statistical Society: Series A (General)*, 150(2):119–137, 1987. 4

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[29] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. 1

[30] Jacob Kauffmann, Malte Esders, Lukas Ruff, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. From clustering to cluster explanations via neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1, 3

[31] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 2

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2, 3

[33] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1): 1096, 2019. 7

[34] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1): 1096, 2019. 4, 5

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4

[36] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 1

[37] Lorenz Linhardt, Klaus-Robert Müller, and Grégoire Montavon. Preemptively pruning clever-hans strategies in deep neural networks. *Information Fusion*, 103:102094, 2024. 7

[38] Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 3

[39] Yunqiu Lv, Jing Zhang, Nick Barnes, and Yuchao Dai. Weakly-supervised contrastive learning for unsupervised object discovery. *IEEE Transactions on Image Processing*, 2024. 4

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 1, 3, 4, 7, 8

[41] Anastasia S Panfilova and Denis Yu Turdakov. Applying explainable artificial intelligence methods to models for diagnosing personal traits and cognitive abilities by social network data. *Scientific Reports*, 14(1):5369, 2024. 1

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019. 4

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4

[44] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. In *29th British Machine Vision Conference*, 2018. 2

[45] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016. 2

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 4

[47] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 1, 2

[48] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*. Edinburgh, 2003. 7

[49] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *Proceedings of the British Machine Vision Conference*, 2021. 3, 4

[50] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 3

[51] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations*, 2015. 2

[52] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 1, 2

[53] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 9626–9635, 2019. 1

[54] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 2022. 3, 4

[55] Kristoffer K. Wickstrøm, Daniel J. Trosten, Sigurd Løkse, Ahcène Boubekki, Karl Øyvind Mikalsen, Michael C. Kampffmeyer, and Robert Jenssen. RELAX: Representation learning explainability. *International Journal of Computer Vision*, 131(6):1584–1610, 2023. 2

[56] Jonathan R Williford, Brandon B May, and Jeffrey Byrne. Explainable face recognition. In *European Conference on Computer Vision*, pages 248–263. Springer, 2020. 1

[57] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33:20554–20565, 2020. 1

[58] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018. 7

[59] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 2

[60] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018. 1, 3