Machine Vision and Applications (2025) 36:33
https://doi.org/10.1007/s00138-024-01653-w

RESEARCH

A concept-aware explainability method for convolutional neural networks

Mustafa Kagan Gurkan[1] · Nafiz Arica[2] · Fatos T. Yarman Vural[3]

Received: 20 February 2024 / Revised: 7 October 2024 / Accepted: 18 December 2024

Abstract

Although Convolutional Neural Networks (CNN) outperform the classical models in a wide range of Machine Vision applications, their restricted interpretability and their lack of comprehensibility in reasoning, generate many problems such as security, reliability, and safety. Consequently, there is a growing need for research to improve explainability and address their limitations. In this paper, we propose a concept-based method, called Concept-Aware Explainability (CAE) to provide a verbal explanation for the predictions of pre-trained CNN models. A new measure, called detection score mean, is introduced to quantify the relationship between the filters of the model and a set of pre-defined concepts. Based on the detection score mean values, we define sorted lists of Concept-Aware Filters (CAF) and Filter-Activating Concepts (FAC). These lists are used to generate explainability reports, where we can explain, analyze, and compare models in terms of the concepts embedded in the image. The proposed explainability method is compared to the state-of-the-art methods to explain Resnet18 and VGG16 models, pre-trained on ImageNet and Places365-Standard datasets. Two popular metrics, namely, the number of unique detectors and the number of detecting filters, are used to make a quantitative comparison. Superior performances are observed for the suggested CAE, when compared to *Network Dissection (NetDis)* (Bau et al., in: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2017), *Net2Vec* (Fong and Vedaldi, in: Paper presented at IEEE conference on computer vision and pattern recognition (CVPR), 2018), and *CLIP-Dissect (CLIP-Dis)* (Oikarinen and Weng, in: The 11th international conference on learning representations (ICLR), 2023) methods.

Keywords Convolutional neural networks · Concept-based explanation · Filter-concept association · Model comparison via explanations

1 Introduction

Deep networks are considered as black boxes, between the low abstraction level input and high abstraction level output. Various implicit and explicit factors, such as model architecture, sub-optimal training methods, dataset bias, the discrepancy between the training and test sets, and statistical insufficiency of data can cause the models to build upon wrong reasonings [4, 5]. Thus, it is required to evaluate the crucial properties of the models such as fairness, reliability, trustworthiness, usefulness, comprehensibility, and safety [6–8]. Explainability methods can help to evaluate some of the above factors [9, 10]. Hence, there is considerable interest in explaining the internal structure of the models after their training.

In the last decade, the work in this area focused on two main streams. The *feature attribution* methods [4, 11–21] show the importance of image pixels in the form of segmented images, sensitivity or pixel attribution heatmaps. However, these methods are vulnerable to the noise embedded in the image [5, 22–24]. The *feature visualization* methods aim to recognize the features learned within hidden

✉ Mustafa Kagan Gurkan
mustafakagan.gurkan@bahcesehir.edu.tr

Nafiz Arica
nafiz.arica@pirireis.edu.tr

Fatos T. Yarman Vural
vural@ceng.metu.edu.tr

[1] Bahcesehir University, 34353 Besiktas, Istanbul, Turkey

[2] Piri Reis University, 34940 Tuzla, Istanbul, Turkey

[3] Middle East Technical University, 06800 Cankaya, Ankara, Turkey

Published online: 11 January 2025

units and attempt to reconstruct the input image from the feature maps [11, 25, 26]. Some of these methods visualize the patterns that maximize the activations [27, 28]. Unfortunately, optimization algorithms suggested in these methods are expensive, result in fragments of mixtures, and are vulnerable to adversarial attacks [29].

Recently, a third stream focusing on *concept-based analysis* gains momentum. These methods rely on the human way of explaining the model decision by describing it with concepts such as actions, objects, and properties. In [1, 2], human-interpretable concepts are assigned to each filter using thresholding and *Intersection over Union (IoU)*. Meanwhile, [30] decomposes output classes into multiple concepts and measures their contribution to the model decision. In [31], a metric is proposed for measuring the relational significance of each concept. In [5], concept activation vectors are defined for distinguishing layer-wise activations against images with and without the concepts. These activation vectors are used in [32] for concept definition, extracted from the segmentation maps of the input images. In [33], the model is trained to represent an object part by preparing templates for each feature map and using them as masks to filter out noisy activations. In [34], batch normalization is replaced with *Concept Whitening* to visualize how different concepts are learned. In [35, 36], a human-in-the-loop method is suggested by setting concept prediction as an intermediary step and allowing intervention to modify predicted concepts. Several approaches have favored leveraging the CLIP encoder model [37], which generates embeddings for both text and images and links concepts to images. In [3], a concept-activation matrix is introduced. It is constructed through the inner products of these embeddings to identify the most similar concept for each unit. In [38], CLIP is paired with heatmap masks to detect and eliminate spurious concepts from explanations. In [39], the explanation output of concept discovery using CLIP embeddings is further refined by producing a class heatmap with Shapley values and another that contrasts class-wise and sample-specific maps.

While the above explainability methods show promise, they also present some limitations. Relying on secondary neural networks introduces additional concerns regarding their interpretability. Methods tailored to a specific model during training may not generalize well to other models. Also, layer-wide responses can dilute if only a few filters exhibit strong activation. Post-hoc approaches that associate each unit with a concept may identify multiple concepts, but due to disentangled representations, only one concept is typically considered. Furthermore, they apply thresholding on activation values and perform *IoU*, which causes two crucial problems: i) High quantile thresholds introduce error by filtering out most of the activated parts, and ii) IoU calculation requires annotated segmentation maps of the image at inference time, which is not available, in practice.

In this study, we present a novel concept-based method called ***Concept-Aware Explainability (CAE)*** to elucidate the reasoning behind CNN model predictions. The suggested CAE model can be summarized as follows:

- First, we define a random variable called ***detection score*** to introduce a probabilistic model for the activation maps obtained at the output of the final convolutional layer of a pre-trained CNN. Based upon this random variable, we propose a new measure, called ***detection score mean*** to quantify the association between a set of concepts and filters.
- Next, we define two sorted lists of lists using the detection score mean values. The first is called the list of ***Concept-Aware Filters (CAF)*** and is a sorted list for each concept concerning the filters. The second one is called the list of ***Filter-Activating Concepts (FAC)*** and is a sorted list for each filter concerning the concepts.
- Finally, we use the ***CAF*** and ***FAC*** lists to generate explainability reports, which explore the predictions of the model by associating its filter activations and the predefined concepts.

The proposed Concept-Aware Explainability (CAE) model differs from other explainability methods in several key ways:

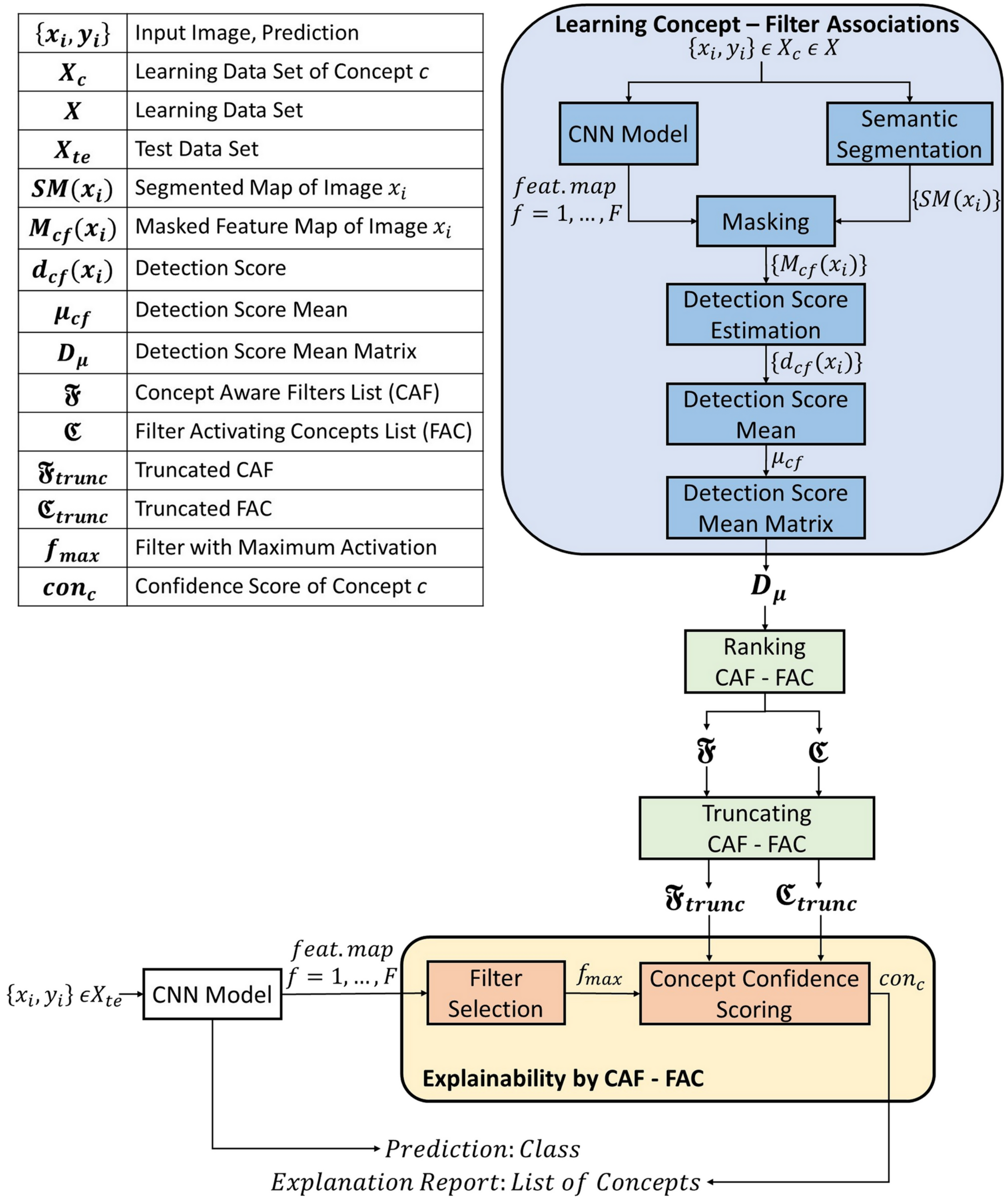
Firstly, it is **model-agnostic**, which can be applied to any CNN model without requiring changes to the model architecture, assistance from external neural networks, or specialized data preparation specific to the model.

Secondly, **it avoids information loss**, caused by the quantile thresholds. Unlike the methods that apply quantile thresholds, CAE preserves information about the concepts by exploring the potential of each unit without such constraints.

Thirdly, it has the capability of **multiple concept associations to the filters**. Instead of limiting each filter to a single concept, CAE allows multiple concept associations per filter. This approach leads to richer and more comprehensive explanations compared to methods that force disentangled representations.

Finally, CAE does not require a segmented map of the test image during inference. It uses previously learned concept-filter associations to generate explanation reports, making it more efficient and versatile.

In summary, CAE provides a more flexible, detailed, and model-independent approach to explaining CNN predictions.



$\{x_i, y_i\}$	Input Image, Prediction
X_c	Learning Data Set of Concept c
X	Learning Data Set
X_{te}	Test Data Set
$SM(x_i)$	Segmented Map of Image x_i
$M_{cf}(x_i)$	Masked Feature Map of Image x_i
$d_{cf}(x_i)$	Detection Score
μ_{cf}	Detection Score Mean
D_μ	Detection Score Mean Matrix
$\mathfrak{F}$	Concept Aware Filters List (CAF)
$\mathfrak{C}$	Filter Activating Concepts List (FAC)
$\mathfrak{F}_{trunc}$	Truncated CAF
$\mathfrak{C}_{trunc}$	Truncated FAC
f_{max}	Filter with Maximum Activation
con_c	Confidence Score of Concept c

Fig. 1 Proposed Architecture. Learning Concept-Filter Associations: Detection score mean computation for each concept-filter pair to quantify their relationship. Ranking CAF and FAC: Identification of CAF and FAC lists ranked by detection score mean values. The output is two lists of sorted lists, $\mathfrak{C}$ with the most activating concepts per filter, and $\mathfrak{F}$ with the most aware filters per concept. Truncating CAF and FAC: Create more compact $\mathfrak{C}$ and $\mathfrak{F}$ lists by limiting the number of filters per concept and number of concepts per filter. Explainability by CAF - FAC: Explain the prediction with the most confident concepts. Confidence of each concept is measured using $\mathfrak{C}$ and $\mathfrak{F}$ lists

2 A concept-aware explainability method for CNNs

The suggested method has three major modules (see Fig. 1);

1. Learning the associations between the concepts and filters by introducing a new measure called detection score mean.
2. Identifying the sorted list of concept-aware filters (CAF) concerning the concepts and the sorted list of filter-activating concepts (FAC) concerning the filters.

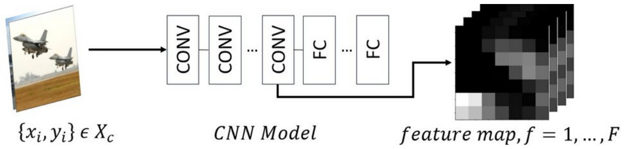


Fig. 2 A pre-trained CNN model with the activation maps of each filter at the final layer of convolution blocks

3. Generating explainability reports using the CAF and FAC lists.In the following subsections, we describe the modules mentioned above.

2.1 Learning the concept-filter associations

In this module, we aim to learn the amount of associations between each concept-filter pair of a pre-trained CNN model.

2.1.1 DataSet generation with the labeled concepts

First, we generate a learning set of images X_c, each containing labeled region(s) of concept c for $c = 1, \ldots, C$, where C is the total number of concepts in an image dataset. This task is achieved by a semantic segmentation algorithm, where we obtain the segmentation map of labeled regions. Mathematically, the segmentation map $SM(x_i)$, of each image $x_i \in X_c$ is defined as,

$$SM(x_i) = \bigcup_{\forall c \in x_i} R_c, \tag{1}$$

where R_c's are the regions of image x_i with the label of concept c. We assume the segmentation map $SM(x_i)$ can have one or multiple concept regions.
Next, we obtain the feature activation maps for each image $x_i \in X_c$ at the output of the final convolution layer of a pre-trained CNN (Fig. 2).

To measure the response of each filter *f* to a concept *c*, we need to determine the activation of the feature map corresponding to the region R_c of the segmentation map $SM(x_i)$. For this purpose, the feature maps are up-sampled to match $SM(x_i)$ in size, and it is masked by the region R_c (Fig. 3). The resulting masked feature map, $M_{cf}(x_i)$, gives the feature activation of image x_i at the output of the filter *f* for the concept c.

2.1.2 Detection score mean estimation between the concept-filter pairs

In this section, we introduce a new measure for quantifying the concept-filter associations based on the masked activation maps $M_{cf}(x_i)$ obtained for each image x_i, using the datasets $X = \bigcup_{\forall c} X_c$.

Definition 1 (*Detection Score Mean and Detection Score Variance*) Consider the activation map $M_{cf}(x_i)$ of image x_i, masked by region R_c at the output of filter *f*, where each pixel at coordinate (*m*, *n*) has the activation value *a*(*m*, *n*). We define the random variable called **detection score**, $d_{cf}(x_i)$, as the average activation value obtained at the output of the filter *f*, for concept region R_c of image x_i. Mathematically, $d_{cf}(x_i)$ is defined as,

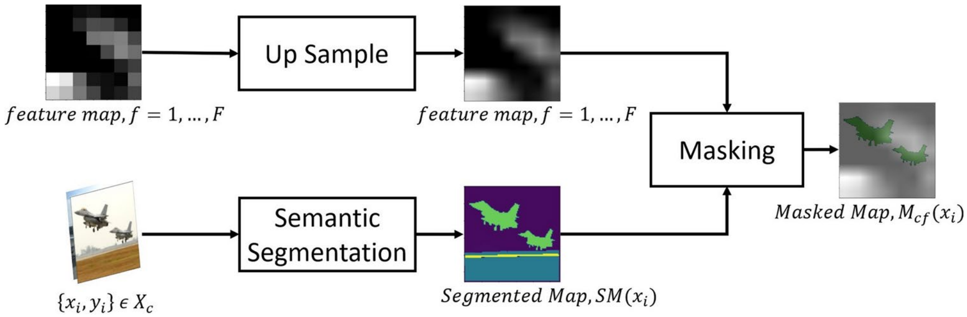
$$d_{cf}(x_i) = \frac{1}{N_c} \sum_{\forall (m,n) \in M_{cf}} a(m,n), \tag{2}$$

where N_c is the number of pixels in the region R_c. We assume that the associated probability density function of $d_{cf}(x_i)$ over all the images $x_i \in X_c$ is a unimodal Gaussian density function $P(d_{cf}) = N(\mu_{cf}, \sigma_{cf})$.
The **detection score mean** is the mean value of $P(d_{cf})$ over all images in the training dataset, which belong to the same category *c* obtained at the output of the filter *f*. It is estimated by the maximum likelihood method, as follows;

$$\mu_{cf} \approx \frac{1}{N_{X_c}} \sum_{\forall x_i \in X_c} d_{cf}(x_i), \tag{3}$$

where N_{X_c} is the number of images in the dataset, X_c, with a set of images containing regions R_c for concept c. Similarly, the **detection score variance** of $P(d_{cf})$ can be estimated by

Fig. 3 Semantic Segmentation: Segmenting the image and labeling each pixel after a concept. Masking: Masking the feature map to focus on activation values of the concept region (color figure online)


feature map, $f = 1, \ldots, F$
Up Sample
feature map, $f = 1, \ldots, F$
Masking
Masked Map, $M_{cf}(x_i)$
$\{x_i, y_i\} \in X_c$
Semantic Segmentation
Segmented Map, $SM(x_i)$

$$\sigma^2_{cf} \approx \frac{1}{N_{X_c}} \sum_{\forall x_i \in X_c} (d_{cf}(x_i) - \mu_{cf})^2. \tag{4}$$

The detection score mean measures the mean activation value of a region representing a concept over all the images in the dataset X_c. As the mean activation value gets high, the filter response for that particular concept has a high mean value, which shows that the filter is highly associated with the underlying concept. As it gets low, we assume that the filter does not care about that particular concept. Similarly, a low variance value indicates certainty about the learned activation of a filter for a concept, and high variances show that the filter acts randomly in learning that concept.

Based upon the detection score mean values between each filter-concept pair, we define the **detection score mean matrix**D_μ and **detection score variance matrix**D_σ, in the following definition:

Definition 2 (*Detection Score Mean Matrix and Detection Score Variance Matrix*) **Detection score mean matrix** and **detection score variance matrix** are defined as follows,

$$D_\mu = [\mu_{cf}] \tag{5}$$

$$D_\sigma = \left[\sigma^2_{cf}\right], \tag{6}$$

respectively. The above matrices are of size $C \times F$, where C is the number of concepts in the concept dictionary and F is the number of filters in the convolution layer.
Note that the entries of the D_μ show the amount of association between each filter concept pair. Similarly, the entries of the D_σ show the degree of uncertainty about these associations.

2.2 Ranking the filters and concepts for generating the CAF and FAC lists

Each row of the detection score mean matrix D_μ shows the activation values of a concept obtained at the output of each filter. Hence, sorting the detection score mean values at each row provides a list from the most activated to the least activated filter. Similarly, each column of D_μ shows the amount of the activation of a filter for each concept, and sorting them yields a list for each filter from the most to the least activating concepts. These sorted lists are called ***Concept-Aware Filters (CAF)*** list and ***Filter-Activating Concepts (FAC)*** list, which are defined below.

Definition 3 (*Concept-Aware Filters (CAF) List*) Given a detection score mean matrix D_μ, define the row vector for concept *c*, as a list,

$$\boldsymbol{\mu_c} = \{\mu_{cf}\}_{f=1}^{F} \tag{7}$$

where *f* is the filter index of each entry of the list. Then, we can define the *Filter Sort* function $S_c(f)$ as,

$$S_c : f \longrightarrow f' \tag{8}$$

which sorts the μ_c list of detection score mean values in descending order and defines the new filter index f' for each entry in the sorted list,

$$\boldsymbol{\mu_{c,sorted}} = \{\mu_{cf'}\}. \tag{9}$$

The new index f' shows the filter index in the sorted list from the most activated filter to the least activated filter for concept *c*. This sorted list is called the list of **Concept-Aware Filters (CAF]** and is shown as,

$$\mathfrak{F}_c = \{f : f = S_c^{-1}(f')\}. \tag{10}$$

Repeating the above sorting procedure for all the concepts in the dictionary yields the list of CAF lists, $\mathfrak{F}$, comprising a sorted list of filters,

$$\mathfrak{F} = \{\mathfrak{F}_c\}_{c=1}^{C}. \tag{11}$$

The list $\mathfrak{F}$ consists of filter indices *f*, sorted concerning the detection score mean values of the CAF lists for the concepts.

Definition 4 (*Filter-Activating Concepts (FAC) List*) Given a detection score mean matrix D_μ, define the column vector for filter *f*, as a list,

$$\boldsymbol{\mu_f} = \{\mu_{cf}\}_{c=1}^{C}, \tag{12}$$

Then, we can define a new function *Concept Sort*$S_f(c)$, a dual form of *Filter Sort*$S_c(f)$, as follows;

$$S_f(c) \rightarrow c' \tag{13}$$

which sorts the $\boldsymbol{\mu_f}$ list in descending order and define the new concept index c' for each entry in the sorted list,

$$\boldsymbol{\mu_{f,sorted}} = \{\mu_{c'f}\}. \tag{14}$$

Then, the concept indices c' can be referred to determine the list of best-fitting concepts for the filter *f*,

$$\mathfrak{C}_f = \{c : \, c = S_f^{-1}(c')\,\} \tag{15}$$

and the FAC list, $\mathfrak{C}$, can be obtained by applying *Concept Sort* function for all filters,

$$\mathfrak{C} = \{\mathfrak{C}_f\}_{f=1}^{F}. \tag{16}$$

2.3 Truncating the CAF - FAC lists

The CAF and FAC sorted lists consist of all concept-filter pairs. However, it is well-known that a limited number of concepts activates each filter, and each concept activates a limited number of filters. Hence, we truncate the total number of concepts and filters to create more compact and meaningful CAF and FAC lists. We empirically define $Y < F$ as the number of filters that are activated by concept *c* and keep only the first *Y* entries of the $\mathfrak{F}_c$. Likewise, we define $Z < C$ as the number of concepts assigned to filter *f* and keep the first *Z* entries of $\mathfrak{C}_f$. This truncation generates two compact lists called $\mathfrak{F}_{trunc}$ of size $C \times Y$, and $\mathfrak{C}_{trunc}$ of size $F \times Z$:

$$\mathfrak{F} \xrightarrow{truncate(Y)} \mathfrak{F}_{trunc} \tag{17}$$

$$\mathfrak{C} \xrightarrow{truncate(Z)} \mathfrak{C}_{trunc}. \tag{18}$$

The parameters *Y* and *Z* are configurable hyper-parameters and may depend on multiple factors, such as filter characteristics, the size of the convolution layer, and the size of the concept dictionary. Their optimization is a crucial design issue. Setting them low may eliminate some useful filters or concepts and thus cause a loss of knowledge, whereas opting for a high value may include irrelevant filters or concepts in the analysis and introduce noise into explanation reports.

2.4 Filter selection and concept confidence scoring

The proposed CAE method attempts to explain the rationale of the prediction of the model for a test image by estimating the concept(s) embedded in the test image detected by the filters (Fig. 1). The process starts by feeding an image from the test set, $x_i \in X_{te}$, to the model and retrieving the feature activation maps obtained at the output of the filters of the final convolution layer. The prediction is also stored for the final report. Then, assuming that the high activation values have a large impact on the prediction, we select the filters that output the highest total activation for the test image. Mathematically, given a test image $x_i \in X_{te}$, let us define the total activation of a feature map of size $M \times N$ for filter *f* as,

$$A_f = \sum_{m}^{M} \sum_{n}^{N} a(m,n), \tag{19}$$

where *a*(*m*, *n*) is the activation of the pixel of the test image at the coordinates *(m,n)*. Then, the list of activations of all the filters is defined as,

$$A = \{A_f\}_{f=1}^{F}, \tag{20}$$

and the filter with maximum total activation f_{max} can be identified as,

$$A_{f_{max}} \geq A_f, \quad \forall f \quad f = 1, ..., F \tag{21}$$

Once the filter f_{max} is selected, we retrieve the concepts assigned to f_{max} from $\mathfrak{C}_{trunc}$, which can be defined as $\mathfrak{C}_{f_{max},trunc}$. Then, for each concept *c* in $\mathfrak{C}_{f_{max},trunc}$, we check the truncated CAF list, $\mathfrak{F}_{trunc}$ to identify the filters assigned to each concept in $\mathfrak{F}_{c,trunc}$ and if f_{max} is one of them. If positive, the confidence score of the concept, con_c, is incremented (see Algorithm 1)

$$con_c = con_c + 1.$$

We remove $A_{f_{max}}$ from A and iterate the above operation to determine the next filter f_{max}, which has the maximum total activation. Since iterating over all the filters counts for even the slightest activation, we stop the iterations at an empirically determined $\mathcal{N}$ number, which depends on the size of the convolution layer.

```
Input:
$\mathfrak{F}$: Concept-Aware Filters (CAF)
$\mathfrak{C}$: Filter-Activating Concepts (FAC)
$Y$: Number of Filters per Concept
$Z$: Number of Concepts per Filter
$A$: List of Filter Activations

Truncate CAF:
$\mathfrak{F}_{c,trunc} = \mathfrak{F}_c[Y,:] \quad \forall c \quad c = 1,...,C$    ▷ Keep first $Y$ elements

Truncate FAC:
$\mathfrak{C}_{f,trunc} = \mathfrak{C}_f[Z,:] \quad \forall f \quad f = 1,...,F$    ▷ Keep first $Z$ elements

for j=1 to $\mathcal{N}$ do    ▷ Iterate $\mathcal{N}$ times
    Filter Selection:
    $f_{max} = f$ if $A_{f_{max}} \geq A_f \quad \forall f, f = 1,...,F$

    Concept Confidence Scoring:
    for c in $\mathfrak{C}_{f_{max},trunc}$ do    ▷ Get the concepts assigned to $f_{max}$
        if $f_{max}$ in $\mathfrak{F}_{c,trunc}$ then    ▷ Check if $f_{max} \in \mathfrak{F}_{c,trunc}$
            $con_c = con_c + 1$
            $A_{f_{max}} = 0$
        end if
    end for
end for

Report Concepts:
```

Algorithm 1 Explainability by CAF and FAC Lists

2.5 Generating explanation report

Algorithm 1 outputs a sorted list of concepts for a test image. The final explanation report keeps the most confident concepts and their associated filters. Concepts with zero confidence score and filters that are not activated by any concept are discarded. The generated explainability report can be used for

- *Model explanation* The relevance of the concepts in the reported list to the predicted class explains the reason behind the correct and wrong predictions. The report can also shed light on faulty explanation attempts such as the impact of concept similarities and potential bias in the learning dataset.
- *Model comparison* It helps to compare models and their training datasets, using the capability for capturing a particular concept or the success of the models for disentangling the concepts embedded in an image.

Table 1 Pre-trained CNN Models used in experiments and their training datasets

Dataset Model	Places365-Standard	ImageNet
ResNet18	ResNet18-Places365	ResNet18-ImageNet
VGG16	VGG16-Places365	VGG16-ImageNet

3 Experimental setup

In the following subsections, we describe the concept dictionary used to generate the explainability reports and the selected CNN models together with their training datasets. Finally, we provide an empirical method for truncating the CAF and FAC lists.

3.1 Concept dictionary

Detection score mean estimation and the CAF and FAC lists form the backbone of our concept-aware explainability method. Therefore, the generation of the concept dictionary is an important task. In this study, we use the concept dictionary of the Broden dataset [1], which was also used in the previous explainability studies [1, 2, 5].

Broden is a densely labeled segmentation dataset with more than 60,000 samples representing 1197 concepts in 6 categories: *objects, scenes, parts, textures, materials, and colors*. For a controllable analysis, our concept dictionary is reduced only to *objects* category. In each segmented image of size 112 × 112, the concepts with a region smaller than 50 pixels are considered negligible. Furthermore, concepts with less than 100 samples are ineligible due to insufficient representation. Following these eliminations, our dataset $X = \bigcup_{c=1}^{C} X_c$ reduces to 32,313 images, where the number of concepts is $C = 200$. Note that even if each concept c has its dataset $X_c \in X$, the images comprise multiple concepts and thus can be part of many X_c.

3.2 CNN models

In our experiments, we use ResNet18 and VGG16 models, each of which is trained with Places365-Standard [40] and ImageNet [41], as summarized in Table 1.

Places365-Standard dataset is for the scene recognition task. It contains 1.8 million training images for 365 output classes. Our concepts in the Broden dataset are also from scenery images. Hence, they can be used in explaining the predictions of the models trained with the Places36Standard dataset.

ImageNet dataset is for object recognition/localization tasks. It contains 1000 *object* classes, wherein only 580 of them match or are related to one of our concepts. The samples of remaining classes can still be used while evaluating the learning ability of the model on the background concepts.

3.3 Truncating the CAF and FAC lists

As highlighted in Sect. 2.3, finding the hyperparameters Y and Z, for truncating the size of the sorted list of filters $\mathfrak{F}$ and sorted list of concepts $\mathfrak{C}$ is crucial. For this purpose, we adopted an empirical approach, and as a reference model, we used the ResNet18-Places365 model.

First, we identify the first 100 images $x_i \in X_c$, for each concept c, with the largest total activation of the feature maps to get $A = \{A_f\}_{f=1}^{F}$ as given in Eq. (20). Then, we repurpose the *Filter Sort* function in Eq. (8) as,

$$S_c : f \longrightarrow f^* \tag{22}$$

to sort the activation values A in descending order and define the new filter index f^* for each entry in the sorted list,

$$A_{sorted} = \{A_{f^*}\}. \tag{23}$$

The new index f^* shows the filter index in the sorted list from the most activated filter to the least activated filter for the image x_i. This sorted list of filter indices can be defined as,

$$\mathfrak{L}_c = \{ f : \, f = S_c^{-1}(f^*) \, \}. \tag{24}$$

At this point, we have two lists of filters: $\mathfrak{L}_c$ sorted by their activations for the given image x_i and $\mathfrak{F}_c$ sorted by their detection score mean values on concept c. In the next step, we truncate both lists by Y and compare the total activations of the filters in the corresponding lists, represented as

$$A_{\mathfrak{L}_{c,trunc}} = \sum_{\forall f \in \mathfrak{L}_{c,trunc}} A_f, \tag{25}$$

and

$$A_{\mathfrak{F}_{c,trunc}} = \sum_{\forall f \in \mathfrak{F}_{c,trunc}} A_f. \tag{26}$$

Figure 4 shows the examples with the best representation among the 100 samples for each concept. The x-axis in the graphs refers to varying values of Y, whereas the y-axis shows the total activation of the filters in the list. The $A_{\mathfrak{L}_{c,trunc}}$ is depicted as the red curve, whereas the blue curve illustrates $A_{\mathfrak{F}_{c,trunc}}$. Even in the best representations, the $A_{\mathfrak{F}_{c,trunc}}$ converges around 30 filters, and considering both ResNet18 and VGG16 models have the same size in the final convolution layer, we set $Y = Z = 30$. It should be noted that other samples may converge for the smaller values of Y.

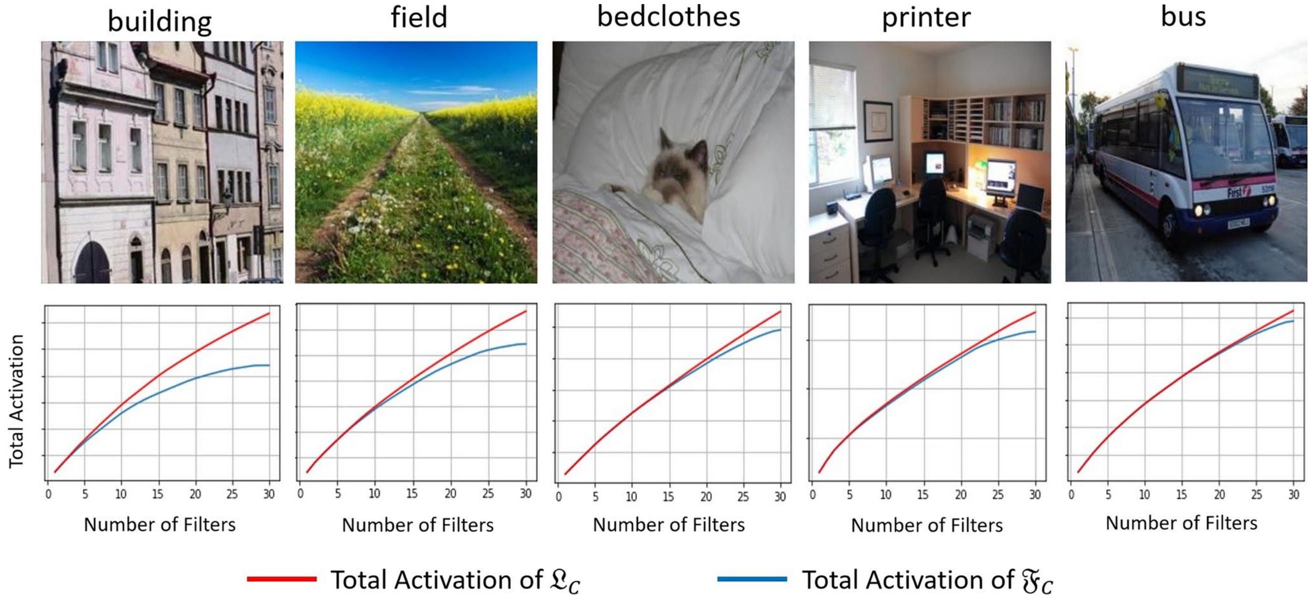


Fig. 4 For the given concept, samples with the best representation among 100 samples with the largest representation. Reference Model: Resnet18-Places365. The range for the number of filters is 1-30. (red) List of filters sorted by their activation, $A_{\mathfrak{L}_{c,trunc}}$, (blue) List of filters sorted by their detection score mean values, $A_{\mathfrak{F}_{c,trunc}}$ (color figure online)

Table 2 Comparison of CAE with existing studies based on Number of Unique Detectors: Each cell shows the number of concepts identified by at least one filter in the final convolution layer of each model

	ResNet18-Places365	VGG16-Places365	ResNet18-ImageNet	VGG16-ImageNet
NetDis [1]	29	25	28	24
Net2Vec [2]	41	54	57	46
CLIP-Dis (cos) [3]	111	105	102	92
CLIP-Dis (swpmi) [3]	169	162	170	159
Dis-CAE	**108**	**140**	**80**	**109**
CAE	**200**	**200**	**199**	**200**

The numbers for both the disentangled and normal modes of our approach are given in bold

Table 3 Comparison of CAE with existing studies based on Number of Detecting Filters: Each cell shows the number of filters in the final convolution layer of each model that can detect at least one concept

	ResNet18-Places365	VGG16-Places365	ResNet18-ImageNet	VGG16-ImageNet
NetDis [1]	481	484	395	442
Net2Vec [2]	191	341	512	339
CLIP-Dis (cos) [3]	512	512	512	512
CLIP-Dis (swpmi) [3]	512	512	512	512
Dis-CAE	**512**	**512**	**512**	**512**
CAE	**512**	**512**	**512**	**512**

The numbers for both the disentangled and normal modes of our approach are given in bold

4 Experimental analysis and results

In the following subsections, we compare our explainability method with similar methods. Then, we conduct a thorough analysis of our CAE method to explain and compare the models in Table 1.

4.1 Comparison with existing explainability methods

Our approach identifies the best set of filters per concept and the best set of concepts per filter. This aspect is similar to the ***Network Dissection (NetDis)*** [1], ***Net2Vec*** [2], and ***CLIP-Dissect (CLIP-Dis)*** [3] which link all the filters within a particular layer to concepts. These studies only estimate concept-filter associations to explain the training process of the model. Therefore, a valid comparison can be done using ***CAF*** and ***FAC*** lists of our method. The following two metrics, proposed in [1], are used for comparison:

- *Number of Unique Detectors* Given a model and a concept dictionary, the number of unique detectors corresponds to the number of concepts identified by at least one filter. Higher values show the model's capability to recognize more concepts and thus its diversity.
- *Number of Detecting Filters* Given a model and a concept dictionary, the number of detecting filters corresponds to the number of filters that can detect at least one concept. Higher values indicate that the model uses more filters for representation.The concept-filter association in NetDis has a limitation of assigning only one concept to each filter. We apply this limitation to our approach and define a new mode, called ***Disentangled CAE (Dis-CAE)*** for a fair comparison.

The number of unique detectors is related to the detection capability of the model. On the other hand, the explainability methods that analyze the model provide the list of final concept-filter associations. This list is then used at inference time for explanations. Consequently, the size of the actual concept dictionary is determined by these unique detectors. In other words, if an explainability method fails to associate any concept with a filter during training, that concept will not be available for explanation during inference.

The data in Table 2 shows that the NetDis and Net2Vec methods can only map a small portion of the concept dictionary to filters, which limits their ability to provide comprehensive explanations. This improves slightly with CLIP-Dis, running on the ViT-B/32 transformer-based CLIP encoder using cosine similarity. However, despite its limited scope, even our Dis-CAE model remains competitive and outperforms its counterpart in the VGG architecture. While CLIP-Dis using the softwpmi similarity function identifies more concepts than the other models, approximately 30-40% of these concepts are linked to just one filter. This suggests weak representation on a convolutional layer with 512 units. In contrast, the CAE method associates nearly all concepts with multiple filters, allowing for stronger representation and full utilization of the concept dictionary to generate explanations.

A low number of detecting filters implies that many filters are deemed unhelpful for explanation purposes. This can lead to information loss during inference, as the activations of these filters will be entirely disregarded. As shown in Table 3, the NetDis and Net2Vec methods struggle to associate concepts with certain filters, resulting in a limited perspective on the model's activations for a given test image. In contrast, other models, including Dis-CAE and CAE, successfully associate concepts with every filter, providing a comprehensive view of filter activations when generating explanation reports.

4.2 Concept-focused learning and concept similarities

We can use the CAF and FAC lists to analyze the learning tendencies of the models in terms of highly activated filters or favored concepts. Table 4 lists the five concepts that activate the highest number of filters for each model. All models can identify the concepts characterized by distinct shapes consistently. Models trained with ImageNet focus more on animals, whereas others are more diverse.

The CAF list $\mathfrak{F}$ can also yield valuable insight into the model's perception of similarity between concepts. Let us define ***Similarity Score*** as,

$$SS(g,h) = \frac{card(\ \mathfrak{F}_{\mathfrak{g}} \cap \mathfrak{F}_{\mathfrak{h}}\ )}{Y}, \tag{27}$$

where $\mathfrak{F}_{\mathfrak{g}}$ and $\mathfrak{F}_{\mathfrak{h}}$ are the list of filters assigned to concepts *g* and *h* and *Y* is the size of the lists. This score can help to identify;

- Redundancy in the learning dictionary. Similar concepts can be set as synonyms. Then, one of them can be removed from the dictionary.
- Implicit bias in the training dataset (if all dogs are black in the training images, *black* and *dog* concepts may be similar).
- Similarities, when the objects complement each other.Table 5 lists the concepts with the highest similarity scores of concept pairs for each model. For instance, in a scene image featuring a *door*, the presence of a *door frame* is a natural assumption. Likewise, if a *blind* is depicted in the image, it is probably positioned on a *window*. In some cases, the models associate the objects with resembling shapes as similar, like *traffic light* vs. *streetlight* or *sofa* vs. *ottoman*.

4.3 Model explanation

The proposed CAE method reports a list of concepts embedded in the image. However, its main objective is not to enumerate the concepts in the image but to show the model's belief regarding their presence. Therefore, an incorrect concept in the explanation report can be as insightful as a correct one while assessing the model's reliability.

In this section, we present the explanation reports of our approach against various images and emphasize its assistance in comprehending the reasoning behind both accurate and wrong predictions. For this purpose, we create the following two test sets.

Table 4 Most Detected Concepts: 5 concepts that activate the highest number of filters in each model

	ResNet18-Places365	VGG16-Places365	ResNet18-ImageNet	VGG16-ImageNet
Most detected concepts	Pool table	Gate	Bird	Dog
	Train	Boat	Dog	Bird
	Bird	Train	Cow	Sheep
	Bridge	Bridge	Sheep	Cow
	Airplane	Airplane	Horse	Horse

Table 5 Concept Similarities: The concept pair with the highest similarity score for each model

	ResNet18-Places365	VGG16-Places365	ResNet18-ImageNet	VGG16-ImageNet
Top similarity score	Windowpane & blind 93%	Street light & traffic light 87%	Sofa & ottoman 93%	Door & door frame 90%

- *Test Set 1* ($\boldsymbol{TS_1}$): 365 images from Places365-Standard test set. We select one random sample for each output class.
- *Test Set 2* ($\boldsymbol{TS_2}$): 898 images from ImageNet test set. We select one random sample for each output class, except for some not covered in our concept dictionary, such as *insects* or *fish*.

During our experiments, we use the TS_1 and TS_2 on Places365 and ImageNet-based models, respectively. Also, the number of iterations $\mathcal{N}$ for filter selection and confidence scoring is selected as 20, based on the reports in [42] where 20 filters are proven to be sufficient for the final convolution layer of the VGG16 network trained on the Places365 dataset to represent the model's decision path and maintain an acceptable prediction accuracy.

4.3.1 Explanation of correct predictions

Figure 5 shows examples of accurate predictions. For the sake of simplicity, only three of the most confident concepts are listed in the explanation report. The confidence scores of these concepts are given in Table 6.

Upon examining the results, we can confirm that the reported concepts match well in articulating the characteristics of the target class. Furthermore, the confidence scores are usually good, with the ResNet18-based models exhibiting higher rates. Therefore, it can be concluded that the models are identifying correct concepts while making their decisions.

4.3.2 Explanation of wrong predictions

Throughout our experiments, we observed mispredictions occasionally. Some of these samples are given in Fig. 6 and

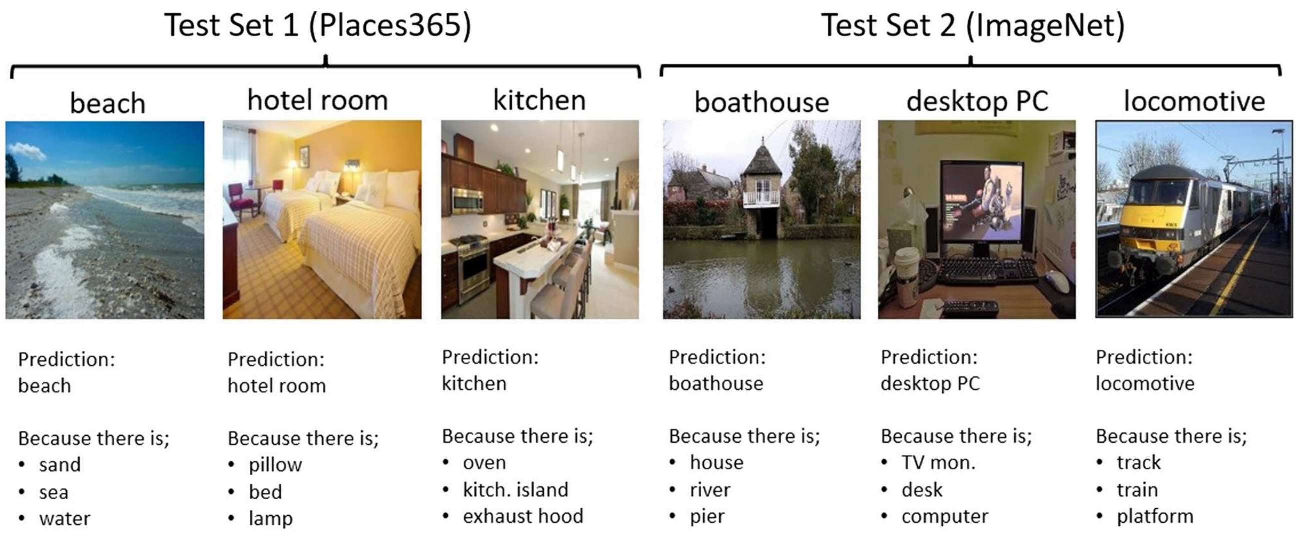

Test Set 1 (Places365)
Test Set 2 (ImageNet)
beach
hotel room
kitchen
boathouse
desktop PC
locomotive
Prediction: beach
Because there is; • sand • sea • water
Prediction: hotel room
Because there is; • pillow • bed • lamp
Prediction: kitchen
Because there is; • oven • kitch. island • exhaust hood
Prediction: boathouse
Because there is; • house • river • pier
Prediction: desktop PC
Because there is; • TV mon. • desk • computer
Prediction: locomotive
Because there is; • track • train • platform

Fig. 5 Explanation of Correct Predictions (by ResNet18). Output labels are at the top of the images. The prediction and the most confident concepts as explanations are at the bottom

Table 6 Most Confident Concepts in Explaining Correct Predictions in Fig. 5

(a)

	Label: Beach		Label: Hotel Room		Label: Kitchen	
	ResNet18-Places365	VGG16-Places365	ResNet18-Places365	VGG16-Places365	ResNet18-Places365	VGG16-Places365
Most Confident Concepts	Sand 85%	Sand 65%	Pillow 90%	Pillow 60%	Oven 80%	Stove 55%
	Sea 70%	Sea 45%	Bed 80%	Bed 50%	Kitch. isl. 80%	Kitch. isl. 45%
	Water 65%	Water 45%	Lamp 75%	Blanket 40%	Exh. hood 70%	Microwave 45%

(b)

	Label: Boathouse		Label: Desktop PC		Label: Locomotive	
	ResNet18-ImageNet	VGG16-ImageNet	ResNet18-ImageNet	VGG16-ImageNet	ResNet18-ImageNet	VGG16-ImageNet
Most Confident Concepts	House 50%	Hedge 45%	Keyboard 50%	Keyboard 55%	Track 70%	Train 55%
	River 45%	River 40%	Computer 45%	Computer 45%	Train 60%	Container 40%
	Pier 40%	House 35%	Desk 40%	TV mon. 40%	Platform 60%	Track 30%

(a) Samples from TS_1 (Places365)

(b) Samples from TS_2 (ImageNet)

the confidence score of the most confident concepts in the reports are listed in Table 7.

The results show that in some cases we may reason with the wrong predictions. For instance, the image labeled as *amusement arcade* features its restaurant area. Therefore, it is reasonable that both models predict it as a *food court*. Likewise, the large concepts in the *hen* image such as *grass* and *fence* are correctly recognized, and the models associate these more with *worm fence* or *turnstile*. In other examples such as *bathroom* or *sandbar*, the reported concepts align well with the actual scene and the mispredictions are notably similar to correct labels. These could easily be labeled as *shower* or *seashore* by another human annotator.

In terms of statistics, these failed predictions reduce the accuracy of the models. However, our explanation approach may help explain the model's rationale and thus can impact its reliability positively.

4.3.3 Concept similarity in explanation reports

As stated in Sect. 4.2, concepts sharing similar traits or complementing each other may activate the same filters, leading to redundancy in the reports. Moreover, if the report size is small, similar concepts may dominate, resulting in a loss of valuable information.

The samples in Fig. 7 and their reports in Table 8 show examples of this issue. In the *street* image, the explanation report predicts multiple vehicular concepts although there only exists *cars*. A comparable outcome is observed in the *moving van* image. Considering the *Similarity Score* values among the *car, van,* and *truck* concepts are in the range of

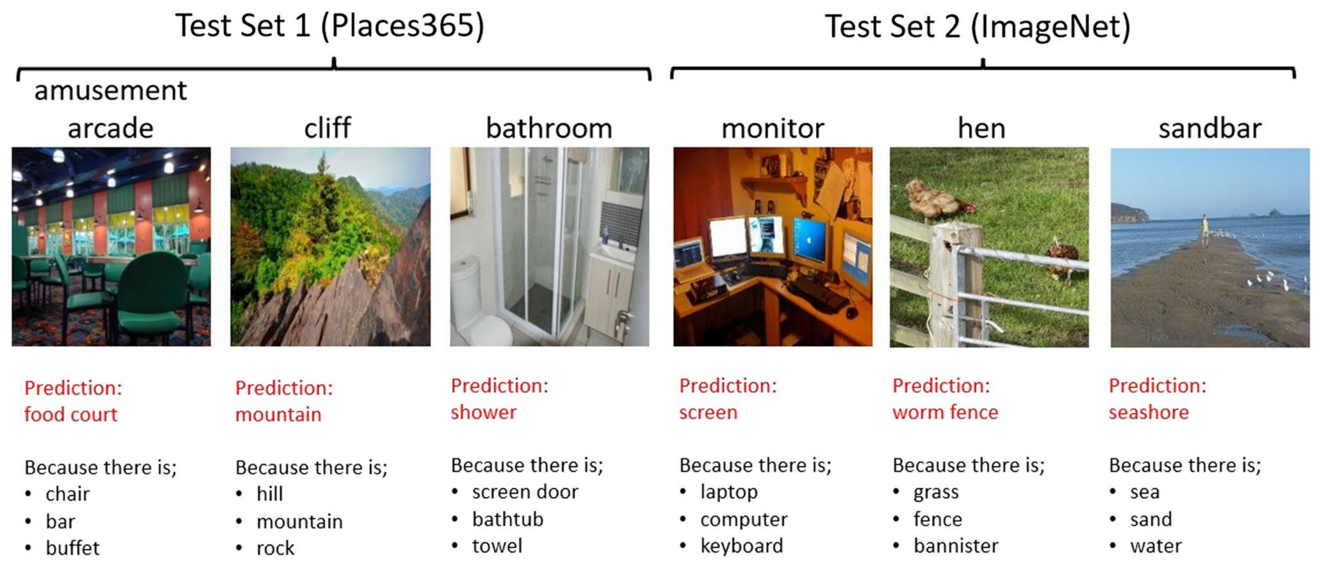


Fig. 6 Explanation of Wrong Predictions (by ResNet18). Output labels are given at the top of the images. The prediction (in red) and the most confident concepts as explanation are given at the bottom (color figure online)

Table 7 Most Confident Concepts in Explaining Wrong Predictions in Fig. 6

(a)						
	Label: Amusement Arcade		Label: Cliff		Label: Bathroom	
	ResNet18-Places365	VGG16-Places365	ResNet18-Places365	VGG16-Places365	ResNet18-Places365	VGG16-Places365
Most Confident Concepts	Chair 30%	Stool 35%	Hill 65%	Mountain 30%	Scr. door 75%	Scr. door 55%
	Bar 30%	Bar 35%	Rock 60%	Rock 15%	Bathtub 65%	Door 50%
	Buffet 30%	Chair 30%	Mountain 50%	Hill 15%	Towel 60%	Curtain 40%
(b)						
	Label: Monitor		Label: Hen		Label: Sandbar	
	ResNet18-ImageNet	VGG16-ImageNet	ResNet18-ImageNet	VGG16-ImageNet	ResNet18-ImageNet	VGG16-ImageNet
Most Confident Concepts	Laptop 80%	Computer 65%	Grass 20%	Grass 25%	Sea 75%	Sand 45%
	Computer 75%	TV mon. 60%	Fence 20%	Pipe 25%	Sand 75%	Road 40%
	Keyboard 70%	Laptop 55%	Bannister 20%	Bannister 25%	Water 45%	Sea 35%

(a) Samples from TS_1 (Places365)

(b) Samples from TS_2 (ImageNet)

0.9 to 0.7, this result can be expected. Likewise, the categorical similarities between animals (i.e. *SS (cow, animal) = 0.8)* influence the report for the *rodeo arena* image. Similar patterns are evident for the *badlands* and *screen* images as well.

The structural resemblances can also factor in similarity impact. For instance, *bird* and *airplane* concepts share 8 out of 30 filters in the CAF list of the ResNet18-ImageNet model, for a similarity score of 0.27. However, in the *black stork* image the *bird* flying with its wings wide open has a shape similar to an *airplane* in the air. That's probably why most of the 8 filters associated with both concepts are among the 20 highest activated filters of the model, which leads the CAE method to include both concepts in the explanation report.

In essence, these findings highlight the significance of overseeing the learning dataset. It is advised to not only build a diverse dictionary but also select the samples to prevent introducing bias. If modifying the learning dataset is not feasible, this risk can be mitigated by conducting a similarity analysis given in Sect. 4.2 and refining the concept dictionary by defining synonyms.

4.3.4 Faulty explanations

The proposed explanation approach has some flaws as well. Figure 8 shows examples of failed attempts due to various reasons such as;

- *Samples representing very few concepts* In the *ocean* and *goldfinch* images, there are only a couple of concepts. If

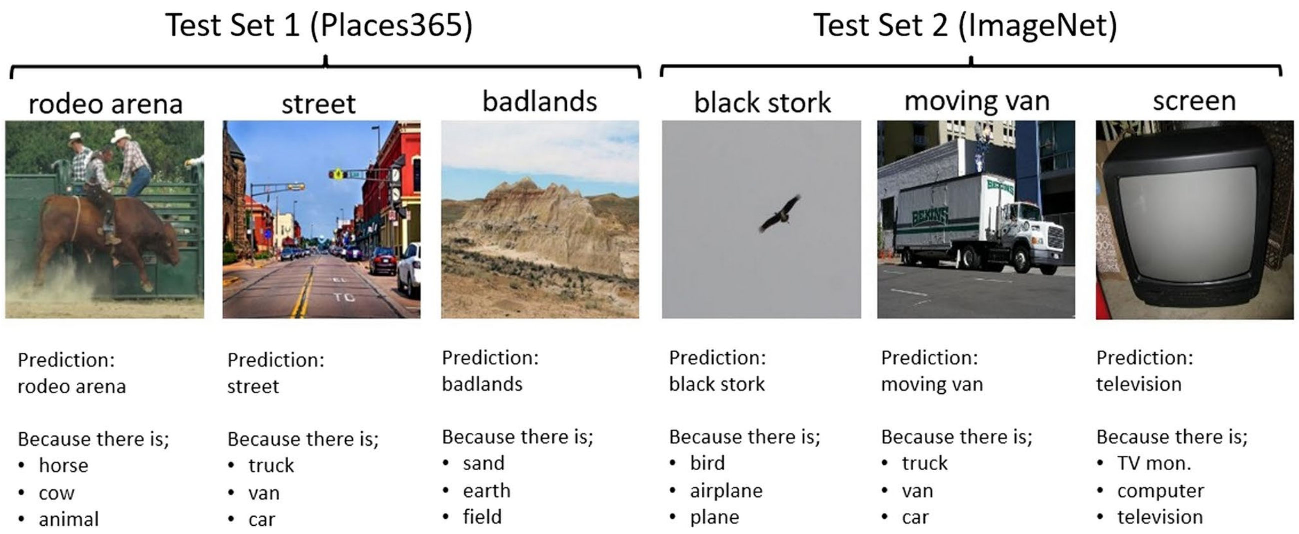


Fig. 7 Concept Similarity Impact in Explanations (by ResNet18). Output labels are given at the top of the images. The prediction and the most confident concepts as explanation are given at the bottom

Table 8 Domination of Similar Concepts in Explaining Images in Fig. 7

(a)						
	Label: Rodeo Arena		Label: Street		Label: Badlands	
	ResNet18-Places365	VGG16-Places365	ResNet18-Places365	VGG16-Places365	ResNet18-Places365	VGG16-Places365
Most Confident Concepts	Horse 80%	Horse 40%	Truck 75%	Truck 45%	Sand 50%	Sand 35%
	Cow 60%	Dog 40%	Van 70%	Van 40%	Earth 45%	Ground 30%
	Animal 60%	Cow 25%	Car 60%	Car 35%	Field 45%	Field 30%
(b)						
	Label: Black Stork		Label: Moving Van		Label: Screen	
	ResNet18-ImageNet	VGG16-ImageNet	ResNet18-ImageNet	VGG16-ImageNet	ResNet18-ImageNet	VGG16-ImageNet
Most Confident Concepts	Bird 50%	Plane 20%	Truck 50%	Van 45%	TV mon. 45%	TV mon. 40%
	Airplane 40%	Bird 15%	Van 45%	Bus 40%	Computer 45%	Computer 35%
	Plane 35%	Airplane 15%	Car 35%	Truck 35%	Television 25%	Television 30%

(a) Samples from TS_1 (Places365)

(b) Samples from TS_2 (ImageNet)

the number of reported concepts is more, a lot of concepts are listed as wrongfully identified, although potentially with lower confidence scores.

- *Potential bias in the learning dataset* Activation may spread to neighboring concepts due to the convolution effect. If the samples of a concept contain a second concept quite often, the CAF lists may relate these two. For instance, farm animals like *horse* or *cow* are reported for the *farm* image. Likewise, *bathtub*, *washer*, or *dishwasher* are listed in the report of the *medicine chest* image.
- *Insufficient concept dictionary* The concept dictionary is limited and, likely, some samples may not be explainable with these concepts. For example, *reptile* is unknown in our dictionary and thus, the report focuses on potentially white-colored concepts for the *alligator lizard* image.
- *Samples with high complexity* In some images, differentiating concepts may be very hard. For instance, the *landfill* image contains a lot of small concepts that can not be properly identified.Some of these issues are related to the quality of the concept dictionary and can be handled by its modification. In the case of very few concepts to identify, the report size can be set dynamically. In short, the abovementioned issues and potential remedies can be planned as future work to enhance the results.

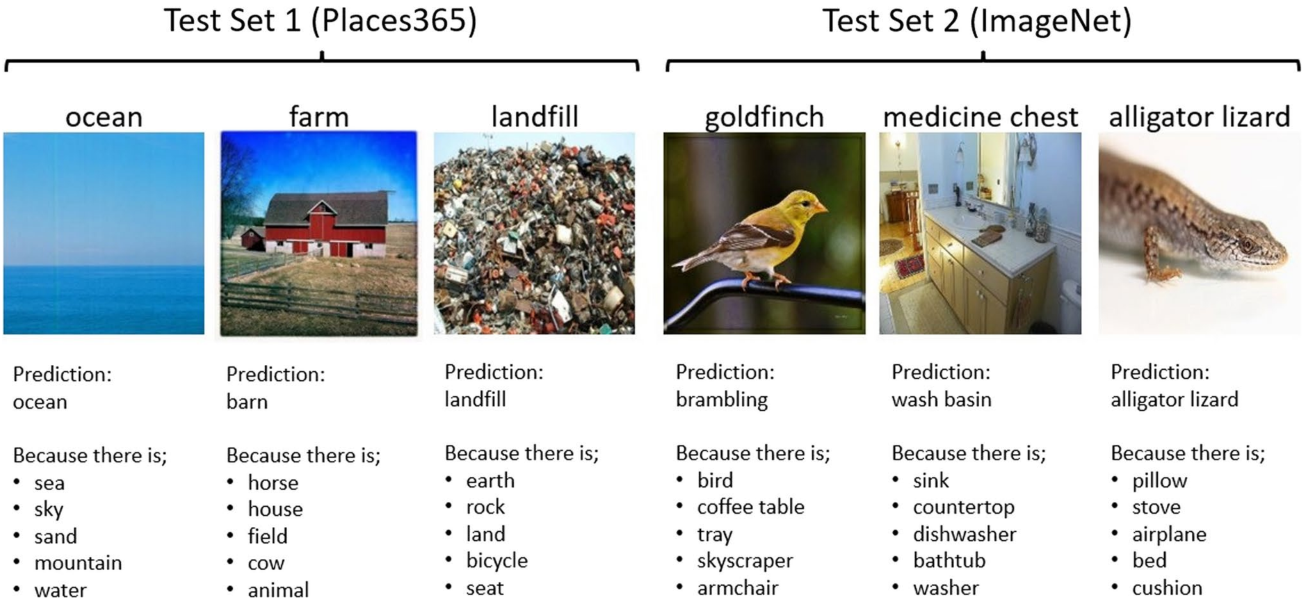

Test Set 1 (Places365)
Test Set 2 (ImageNet)
ocean
farm
landfill
goldfinch
medicine chest
alligator lizard
Prediction: ocean
Because there is; • sea • sky • sand • mountain • water
Prediction: barn
Because there is; • horse • house • field • cow • animal
Prediction: landfill
Because there is; • earth • rock • land • bicycle • seat
Prediction: brambling
Because there is; • bird • coffee table • tray • skyscraper • armchair
Prediction: wash basin
Because there is; • sink • countertop • dishwasher • bathtub • washer
Prediction: alligator lizard
Because there is; • pillow • stove • airplane • bed • cushion

Fig. 8 Faulty Explanations (by ResNet18). Output labels are given at the top of the images. The prediction and the most confident concepts as explanations are given at the bottom

Table 9 Average number of correctly predicted concepts in the explanation report

Test Set 1		Test Set 2	
ResNet18-Places365	VGG16-Places365	ResNet18-ImageNet	VGG16-ImageNet
3.6	3.16	1.5	1.56

TS_1 and TS_2 are derived from Places365-Standard and ImageNet datasets, respectively

4.4 Model comparison

The proposed approach can also help to compare models and their training datasets through the examination of the explanation reports. To achieve this, we must establish a set of comparison criteria. In the subsequent sections, we define these criteria and analyze the results.

4.4.1 Correct concepts in explanation

The trust instilled in the model's correct prediction increases if the list of concepts in the explanation report is close to the actual concepts. Hence, verification of how accurate the reports are is important. For this analysis, we use the samples of TS_1 and TS_2 created in Sect. 4.3. As they lack segmentation and labeling, we need human effort to detect matches between actual and reported concepts. The size of the report is limited to ten concepts with the highest confidence score.

Table 9 provides the average number of correctly predicted concepts for each model. In Places365-Standard samples, the ResNet18 model is superior, while VGG16 exhibits slightly better performance in ImageNet samples. On the other hand, the models trained with and tested against Places365 report significantly more concepts accurately. However, we should note that ImageNet samples contain fewer concepts overall. Hence, utilizing the number of correct predictions in this fashion is not suitable for comparing the impact of training datasets.

At this point, we also assessed whether the convolution operation adversely affects our calculations by spreading the activation across neighboring concepts. To explore this, we implemented thresholding on feature maps, before identifying highly activated filters. However, we observed slight improvement and decided to proceed without thresholding.

4.4.2 Recall of concepts in explanation report

As stated in 4.4.1, the number of correct concepts in the explanation is insufficient for a meaningful comparison of training datasets. Consequently, we decided to use the recall value of the report. To be more specific, let us define I and E as the list of concepts in the image x_i and the explanation report, respectively,

$$I = \{c_i\}\ , \quad E = \{c_r\}. \tag{28}$$

The ***Report Recall (RR)*** can be computed as;

$$RR = \frac{TP}{TP + FN} \tag{29}$$

where

Table 10 Report Recall values for all the models on TS_3 and TS_4

	ResNet18-Places365	VGG16-Places365	ResNet18-ImageNet	VGG16-ImageNet
Test Set 3	25.4%	22.8%	17.3%	17.5%
Test Set 4	26%	26%	17.3%	20%

$$\forall c \in I, \quad c \rightarrow \begin{cases} TP & \text{if } c \in E \\ FN & \text{otherwise.} \end{cases} \tag{30}$$

For proper comparison, the test samples should ensure all the concepts in the concept dictionary are represented at least several times. Also, the list of concepts I should be known beforehand. Furthermore, the samples should exhibit sufficient complexity such that one or two concepts do not dominate. To meet these requirements, we create two new test sets TS_3 and TS_4, both sourced from the Broden dataset. To be specific,

- *Test Set 3* ($\boldsymbol{TS_3}$): 973 samples incorporating 7 concepts.
- *Test Set 4* ($\boldsymbol{TS_4}$): 883 samples incorporating 10 concepts.To compute *RR*, the size of the explanation report is set to match the number of concepts in the image. The results are given in Table 10. Once more, the models trained with Places365 show better performance, albeit with a narrower margin. This may be caused by a couple of reasons such as potential dataset imbalance in ImageNet as the models are trained to focus on specific concepts for object recognition task or the inadequacy of the concept dictionary.

4.4.3 Per concept analysis

The *RR* can be regarded as a valuable metric for generalization. However, in some cases, it becomes essential to assess the model behaviors against specific concepts. For instance, a model may completely ignore a particular concept and its absence can impact the overall results significantly.

In this section, we shift our focus to examine how models respond to specific concepts. Relying solely on recall value as a metric may be insufficient for this purpose because reporting a false positive concept can be as important as missing an actual concept. Hence, we decided to use ***F1 Score*** as the comparison metric. By revisiting Eq. (28) to define the set of concepts in the report and the image, we can formulate the ***F1 Score*** as;

$$F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{31}$$

where

Table 11 Concepts with highest F1 Scores for each model on test sets TS_3 and TS_4

	ResNet18-Places365	VGG16-Places365	ResNet18-ImageNet	VGG16-ImageNet
Test Set 3	Car 60.3%	House 62.5%	Dog 63.5%	Skyscraper 55.6%
	Food 59.3%	Boat 58%	Food 59%	Boat 54.5%
	Train 54.9%	Car 54.7%	Cat 55.9%	Bicycle 53.3%
Test Set 4	Car 65.3%	Track 60%	Cat 55.2%	Dog 60.9%
	Bed 60.9%	Road 52.6%	Food 54.8%	Bicycle 47.4%
	Sidewalk 57.8%	Grass 51.4%	Car 54.8%	Windowpane 46.6%

$$c \rightarrow \begin{cases} TP & \text{if } c \in I, \quad c \in E \\ FP & \text{if } c \notin I, \quad c \in E \\ FN & \text{if } c \in I, \quad c \notin E \end{cases} \tag{32}$$

The experiments are conducted for all the concepts in the concept dictionary, across all four models and test sets TS_3 and TS_4. Table 11 gives the three concepts with the highest F1 scores for each model. Despite some concepts such as *cat* and *food* staying on top for ResNet18-ImageNet, or *car* having high scores in general, the results display substantial variations. Therefore, we can deduce that even though the models have the same training set, they show significant differences in learning concepts and they can outperform each other in a specific class that aligns with these concepts.

5 Discussions and conclusion

This paper introduces a novel concept-based explanation method for pre-trained CNN models. The approach includes a learning phase, where a set of images segmented and densely labeled after a predefined list of concepts is processed to quantify the relationship between each concept-filter pair. This information is then used to create lists of the most aware filters per concept and the best-fitting concepts per filter. The explanation phase utilizes these lists to predict what the model detects in an image.

To demonstrate its efficacy, our approach undergoes various experiments. The theoretical examination of concept-filter relations shows that certain concepts may exhibit significant activation in feature map representations. Also, similarities can pose challenges in differentiating the concepts, and filters might respond to certain objects more frequently than others. The explanation reports are valuable not only in justifying successful predictions but also in highlighting the factors that confuse the model's decision-making process in incorrect predictions. Moreover, the importance of selecting concepts and performing similarity analysis during dictionary formation is underlined through illustrative examples. Additionally, experiments are carried out to demonstrate how the proposed method can be employed for comparing models in terms of their explainability. These involve the use of different metrics, like Report Recall or F1 score, to assess the models' performances concerning all concepts in an image or specific concepts.

Despite promising results, there is still potential for improvement. The dictionary employed in the experiments comprises only 200 concepts and is proven to be insufficient for explaining many test samples. The list can be enhanced by incorporating not only more objects but also parts of objects, colors, textures, etc.

The reliability of the segmented images used as learning dataset raises concerns, as many of these images primarily consist of very small patches representing concepts, and occasional labeling issues are present. Moreover, some concepts have significantly more samples compared to others. Furthermore, the scale of the experiments can be expanded by incorporating additional CNN models trained with a broader range of datasets. Therefore, repeating the experiments with a more robust, homogeneous, and efficient learning dataset on more models can lead to more comprehensive and insightful results.

Another potential improvement could be in the filter selection stage. Rather than examining filter-wide activations, an alternative option may involve segmenting the test images and comparing these segments with the CAF lists. However, it's important to note that this is currently theoretical and would require extensive effort to validate its correctness.

In conclusion, the ***Concept-Aware Explainability*** method introduced in this paper can serve various purposes such as providing insights into the model's training process, identifying the concepts where a model excels or struggles, understanding what a model perceives during prediction, and facilitating model comparisons. Nevertheless, it remains open to further refinement and merits additional efforts to enhance its outcomes.

Acknowledgments The corresponding author thanks the Scientific and Technological Research Council of Turkey for the financial support provided during this work.

Author contributions The corresponding author M.K.G. worked in the conceptualization of the idea, design of the methodology, formal analysis of the data, software programming and validation, and writing of the main manuscript text. N.A. worked in the conceptualization of the idea and supervised the design of the methodology and validation of the results. F.Y.V. supervised the design of the methodology, formal analysis of the data and validation of the results. F.Y.V also contributed in the writing of the main manuscript text. All authors reviewed the manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: quantifying interpretability of deep visual representations. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (2017). https://doi.org/10.1109/CVPR.2017.354
2. Fong, R., Vedaldi, A.: Net2Vec: quantifying and explaining how concepts are encoded by filters in deep neural networks. In: Paper presented at IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018 (2018). https://doi.org/10.1109/CVPR.2018.00910
3. Oikarinen, T., Weng, T.-W.: CLIP-Dissect: automatic description of neuron representations in deep vision networks. In: The 11th international conference on learning representations (ICLR), Kigali, Rwanda (May 2023)
4. Selvaraju, R.R., Cogswall, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Paper presented at IEEE international conference on computer vision (ICCV), Venice, Italy, 22–29 October 2017 (2017). https://doi.org/10.1109/ICCV.2017.74
5. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., Sayres, R.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: Paper presented at international conference on machine learning (ICML), Stockholm, Sweden, 10–15 July 2018 (2018)
6. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017). arxiv.org/abs/1711.11279
7. Ras, G., Xie, N., Gerven, M., Doran, D.: Explainable deep learning: a field guide for the uninitiated. J. Artif. Intell. Res. (2022). https://doi.org/10.1613/jair.1.13200
8. Narwaria, M.: Does explainable machine learning uncover the black box in vision applications? Image Vis. Comput. **118**, 104353 (2022). https://doi.org/10.1016/j.imavis.2021.104353
9. Barredo Arieta, A., Diaz-Rodriguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
10. Saeed, W., Omlin, C.: Explainable AI (XAI): a systematic meta-survey of current challenges and future opportunities. Knowl.-Based Syst. **263**, 110273 (2023). https://doi.org/10.1016/j.knosys.2023.110273

11. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Paper presented at European conference on computer vision (ECCV), Zurich, Switzerland, 6–12 September 2014 (2014). https://doi.org/10.1007/978-3-319-10590-1_53
12. Sundarajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the 34th international conference on machine learning (ICML) (2017).https://doi.org/10.5555/3305890.3306024
13. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: SMOOTHGRAD: removing noise by adding noise (2017). arxiv.org/abs/1706.03825
14. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS ONE (2015). https://doi.org/10.1371/journal.pone.0130140
15. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. Int. J. Comput. Vision **126**, 1084–1102 (2018). https://doi.org/10.1007/s11263-017-1059-x
16. Li, H., Tian, Y., Mueller, K., Chen, X.: Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation. Image Vis. Comput. **83–84**, 70–86 (2019). https://doi.org/10.1016/j.imavis.2019.02.005
17. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Paper presented at IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 26 June– 1 July 2016 (2016) https://doi.org/10.5555/3305890.3306024
18. Desai, S., Ramaswamy, H.G.: Ablation-CAM: visual explanations for deep convolutional network via gradient-free localization. In: Paper presented at IEEE winter conference on applications of computer vision (WACV), Snowmass, CO, USA 1–5 March 2020 (2020). https://doi.org/10.1109/WACV45572.2020.9093360
19. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., S., D., Mardziel, P., Hu, X.: Score-CAM: score-weighted visual explanations for convolutional neural networks. In: Paper presented at IEEE conference on computer vision and pattern recognition workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020 (2020). https://doi.org/10.1109/CVPRW50498.2020.00020
20. Li, Q.: Saliency prediction based on multi-channel models of visual processing. Mach. Vis. Appl. **34**, 47 (2023). https://doi.org/10.1007/s00138-023-01405-2
21. Han, H., Faust, R., Keith Norambuena, B.F., Lin, J., Li, S., North, C.: Explainable interactive projections of images. Mach. Vis. Appl. **34**, 100 (2023). https://doi.org/10.1007/s00138-023-01452-9
22. Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K.T., Dähne, S., Erhan, D., Kim, B.: The (un)reliability of saliency methods, 267–280 (2019)
23. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity Checks for Saliency Maps. In: Proceedings of the 32nd international conference on neural information processing systems (NIPS) (2018). https://doi.org/10.5555/3327546.3327621
24. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is Fragile. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019). https://doi.org/10.1609/aaai.v33i01.33013681
25. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: Paper presented at IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 26 June– 1 July 2016 (2016). https://doi.org/10.1109/CVPR.2016.522
26. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. Int. J. Comput. Vis. **128**, 1867–1888 (2020). https://doi.org/10.1007/s11263-020-01303-4
27. Nguyen, A., Yosinski, J., Clune, J.: Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks (2016). arxiv.org/abs/1602.03616
28. Zimmermann, R.S., Borowski, J., Geirhos, R., Bethge, M., Wallis, T.S.A., Brendel, W.: How well do feature visualizations support causal understanding of CNN Activations? In: Paper presented at advances in neural information processing systems (NeurIPS), 7–10 December 2021 (2021)
29. Nanfack, G., Fulleringer, A., Marty, J., Eickenberg, M., Belilovsky, E.: Adversarial attacks on the interpretation of neuron activation maximization (2023). arxiv.org/abs/2306.07397
30. Zhou, B., Sun, Y., Bau, D., Torralba, A.: Interpretable basis decomposition for visual explanation. In: Paper presented at European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018 (2018). https://doi.org/10.1007/978-3-030-01237-3_8
31. Anjomshoae, S., Omeiza, D., Jiang, L.: Context-based image explanations for deep neural networks. Image Vis. Comput. **116**, 104310 (2021). https://doi.org/10.1016/j.imavis.2021.104310
32. Ghorbani, A., Wexler, J., Zou, J., Kim, B.: Towards automatic concept-based explanations. In: Proceedings of the 33rd international conference on neural information processing systems (NIPS) (2019). https://doi.org/10.5555/3454287.3455119
33. Zhang, Q., Wu, Y.N., Zhu, S.-C.: Interpretable convolutional neural networks. In: Paper presented at IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018 (2018). https://doi.org/10.1109/CVPR.2018.00920
34. Chen, Z., Bei, Y., Rudin, C.: Concept whitening for interpretable image recognition. Nat. Mach. Intell. **2**, 772–782 (2020). https://doi.org/10.1038/s42256-020-00265-z
35. Koh, P.W., Nguyen, T., Tang, Y.S., Mussmann, S., Pierson, E., Kim, B., Liang, P.: Concept bottleneck models (2020). arxiv.org/abs/2007.04612
36. Havasi, M., Parbhoo, S., Doshi-Velez, F.: Addressing leakage in concept bottleneck models. In: Paper presented at advances in neural information processing systems (NeurIPS), 12–14 December 2022 (2022)
37. Radford, A. et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the international conference on machine learning (ICML) (2021)
38. Kalibhat, N. et al.: Identifying interpretable subspaces in image representations. In: Proceedings of the 40th international conference on machine learning (ICML), pp. 15623–15638 (2023). https://doi.org/10.5555/3618408.3619045
39. Ahn, Y., Kim, H., Kim, S.: WWW: a unified framework for explaining what, where and why of neural networks by interpretation of neuron concepts. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR), Seattle, WA, USA, pp. 10968–10977 (2024). https://doi.org/10.1109/CVPR52733.2024.01043
40. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. http://places2.csail.mit.edu/download.html (2017)
41. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. https://www.image-net.org/download.php (2009)
42. Bau, D., Zhu, J.-Y., Strobelt, H., Lapedriza, A., Zhou, B., Torralba, A.: Understanding the role of individual units in a deep neural network. In: Proceedings of the national academy of sciences of the United States of America (PNAS) **117**(48), 30071–30078 (2020). https://doi.org/10.1073/pnas.1907375117

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.