

Exploiting Text-Image Latent Spaces for the Description of Visual Concepts

Laines Schmalwasser^{1,2}[0009-0006-1120-1299], Jakob Gawlikowski¹[0000-0003-2492-4358], Joachim Denzler²[0000-0002-3193-3300], and Julia Niebling¹[0000-0001-5413-2234]

¹ Institute of Data Science, German Aerospace Center, 07745 Jena, Germany

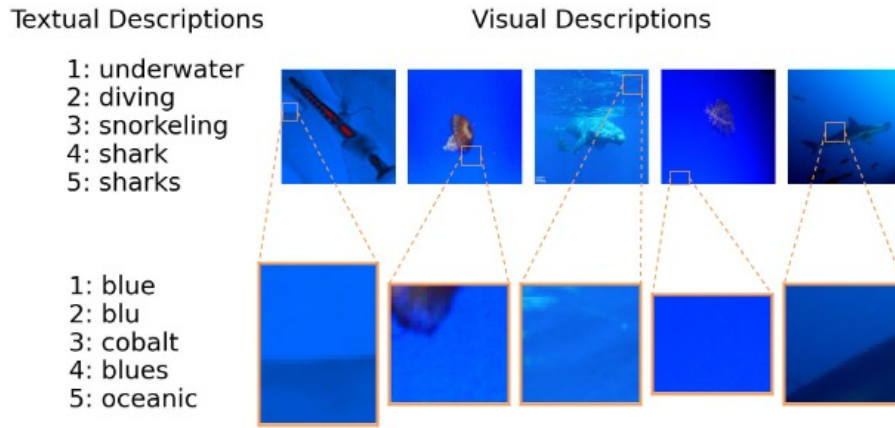
² Computer Vision Group, Friedrich Schiller University Jena, 07743 Jena, Germany
laines.schmalwasser@dlr.de

Abstract. Concept Activation Vectors (CAVs) offer insights into neural network decision-making by linking human friendly concepts to the model’s internal feature extraction process. However, when a new set of CAVs is discovered, they must still be translated into a human understandable description. For image-based neural networks, this is typically done by visualizing the most relevant images of a CAV, while the determination of the concept is left to humans. In this work, we introduce an approach to aid the interpretation of newly discovered concept sets by suggesting textual descriptions for each CAV. This is done by mapping the most relevant images representing a CAV into a text-image embedding where a joint description of these relevant images can be computed. We propose utilizing the most relevant receptive fields instead of full images encoded. We demonstrate the capabilities of this approach in multiple experiments with and without given CAV labels, showing that the proposed approach provides accurate descriptions for the CAVs and reduces the challenge of concept interpretation.

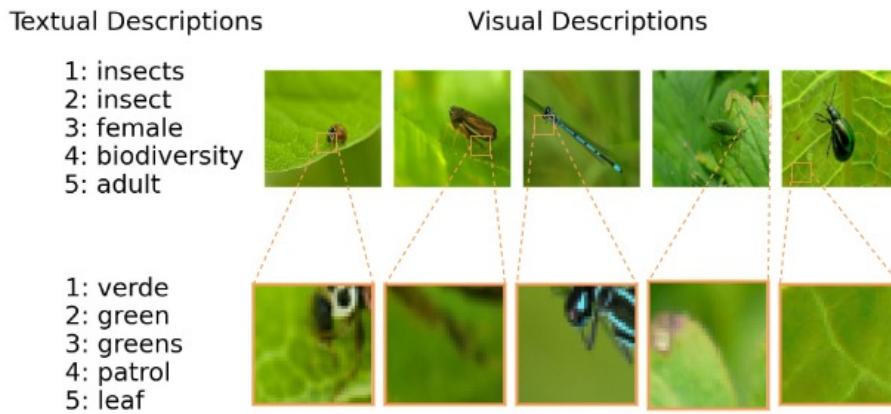
Keywords: XAI, Explainability, Concepts, Textual Description, Text-Image-Embeddings

1 Introduction

One major challenge of deep neural networks is their black-box nature which makes the interpretation of their behavior difficult. To mitigate this drawback, multiple approaches have been proposed to highlight relevant parts of the input data for a given prediction, for example, LIME [27], SHAP [19], GradCAM [29], LRP [1] and Feature Visualization [24]. Another idea is to explain the internal mechanism of a deep neural network in terms of concepts that are understandable and easy to communicate to humans [4,14,26]. One attempt to identify such concepts is with so-called Concept Activation Vectors (CAVs) [14]. A CAV is a vector in the feature space of the activations of a specific network layer. It is designed to point to the direction of activations that are connected to a specific human understandable concept.



(a) Top derived descriptions: *underwater* vs. *blue*



(b) Top derived descriptions: *insects* vs. *verde*

Fig. 1: Examples of two CAVs computed from the first residual block of a ResNet50, trained on Animals with Attributes 2 [32]. The first row of each subfigure shows the full representative images of the CAVs and the textual descriptions generated based on the full images. The second row shows the representative receptive fields for the same CAVs and the textual descriptions are derived from the receptive fields.

The idea behind CAVs is that a human defined concept that contributes to the model decisions has a representation in the model’s embedding space. For

example, the concept *stripe pattern* should have a corresponding representation when the model uses it in the decision-making process to predict a zebra.

In the literature, approaches have been suggested to find CAVs in a supervised and an unsupervised manner: While for the supervised approaches example images that contain the desired concepts are utilized [14,20,34], the unsupervised approaches use, for example, network bottlenecks to extract CAVs [33,34].

We aim to describe the utilized concepts of a pretrained network without any assumptions about the concepts and without the need for example images for the concept. Hence, we focus on the description of an unsupervised discovered set of CAVs. As the discovered CAVs are given as vectors in the feature space, the encoded concepts need to be described for humans. A common way is to show images of a given dataset, which are most similar to the respective CAV in the hidden representation. However, this introduces the need for interpretation to derive a compact and communicable meaning from the given images.

To avoid the need for human interpretation, we propose to determine a ranking of textual descriptions for each concept. Depending on the CAV, the textual descriptions to be ranked, and the fine granularity of the text embedding, the highest ranked descriptions can be highly redundant. Therefore, we further derive a single common description based on the k highest ranked descriptions. Depending on the ranking, this common description can differ from the highest ranked description.

We build up on existing approaches to describe the information filtered by individual neurons in a textual way, for example, [23]. In this approach a neuron is described by generating a textual description for the relevant images of a neuron for which the neuron has the highest activation. The textual description is chosen as the best fitting one out of multiple candidates. In contrast to individual neurons, a major advantage of CAVs is that they represent vectors in the feature space and not only individual scalar neuron outputs. The total number of CAVs is usually significantly lower than the number of neurons in the corresponding layer.

The textual descriptions of the individual neurons in [23] are based on the full images that are relevant for the considered neuron. However, when the variety of images in a data set is not large enough, it is often not possible to separate highly correlated concepts, especially concepts of different degrees of abstraction, purely based on the full images. One example of the issue of highly correlated concepts are the concepts *insects* and *verde*, see Figure 1a. An example of concepts of different degrees of abstraction are the concepts *underwater* and *blue*, as in many cases *underwater* is a specification of *blue*, see Figure 1b. To address this limitation, we propose to use receptive fields instead of the full images for the generation of the textual descriptions. By replacing the full images with receptive fields, we can focus on the parts of the images, where an evaluated concept is most present. This reduces the noise that can affect the textual description of the concept.

In summary, the interpretation process of a neural network by ranking textual descriptions of human understandable concepts is represented by CAVs. Further,

we derive a single common textual description to decrease the redundancy. Our main contributions are:

- We enhance the automatic concept discovery in a trained model by interpreting the visual CAVs with textual descriptions.
- We derive a common concept description from the top- k computed textual descriptions to reduce redundancy.
- We propose using receptive fields to derive the textual descriptions and introduce concept scores to measure the relevance of the receptive fields. By that the textual descriptions focus on the relevant parts of the images, e.g. only the parts of the image seen by the model up to that layer.

2 Related Work

Concepts. The idea that certain directions in a model’s latent representation align with human-understandable concepts was initially proposed by Kim et al. [14]. They propose to learn a hyperplane in the activation space of a neural network layer that separates images, which include the concept, from other images. The normal of the hyperplane in the direction of the images encoding the concept is the Concept Activation Vector (CAV). Since then, a lot of effort was put into the automatic discovery of such concepts activation vectors [8,9,22,33,36]. Interesting for our work is the novel concept discovery algorithm proposed by Yeh et al. [33], which combines interpretability with a new notion of *completeness* which measures how sufficient a set of CAVs is for the explanation of a model’s prediction behavior. They also introduce a method to rank the found CAVs by importance called ConceptSHAP which adapts Shapley values [28]. Shapley values assign importance to a feature by calculating its average contribution in all possible combinations. One drawback of approaches for automatic CAV discovery is that they rely on images as references for the explanation of a CAV.

Network Dissection. The idea of dissecting a network is to inspect the function of individual neurons in the network to get insights into the model. The first work about network dissection provided a method to quantify the interpretability of latent representations by comparing neuron activations with segmentation masks from a concept dataset [2]. This approach aligns individual neuron activations of a model with specific visual concepts given by the segmentation masks. One major limitation of this approach is, that the masks needed to be annotated by humans. Based on this, a segmentation model was proposed in [3] to annotate the masks for each concept. MILAN [11] extends the labeling of neurons to open-ended natural language descriptions: This approach generates descriptions of neurons by finding language strings that maximize the mutual information of the image regions where the neuron is active. To generate the language description, an image-to-text model is required, trained on a labeled data set. To avoid the need for labeled data, CLIP-Dissect [23] leverages the multimodal training of CLIP [25], a method that embeds image and text data to a joint feature space.

Joint Text-Image Embeddings. In recent years, there have been significant advancements in learning joint text-image embeddings [12,18,25,35]. Text-image embeddings can be utilized to perform various tasks, such as zero-shot classification. Contrastive learning based approaches, such as CLIP [25], are trained to maximize the similarity between positive examples (e.g., images and matching image captions) and to minimize the similarity to negative examples (e.g., non-matching image-caption pairs). Approaches such as CLIP have shown good zero-shot image classification performance on multiple data sets by evaluating the similarity between the feature embeddings of the class labels and the images.

Post-Hoc Concept-Bottleneck Models. An alternative approach to generating post-hoc concept explanations is to first create a set of known CAVs and then find the subset of those CAVs that yield the best performance for a given model [20,34]. Those approaches assume to have CAVs for all important concepts and then select the CAVs that can describe the essence of what was learned by the model. In our approach, the set of CAVs is discovered automatically by inspecting the model in more detail like in [33], and then designated by textual descriptions.

3 Method

We propose a method that derives textual descriptions for the concepts a neural network utilizes to solve an image classification task. The method consists of three steps, and each step represents a different level of concept description for a given neural network:

1. The **discovery of concepts** by concept activation vectors (CAVs), represented as directions in the feature space,
2. the **visual description** of the concepts (encoded by the CAVs) with representative images,
3. and the **textual description** of the concepts with words.

The steps are visualized in Figure 2. In the following, the inputs, the three steps of the method, and the computed outputs are introduced in more detail.

Inputs. The method is based on a neural network trained on an image classification task, f , that maps input images to a K -dimensional output vector representing class probabilities. For a given layer l , for which concepts shall be extracted from the network, the network is decomposed into two functions h_l and ϕ_l , such that $f = h_l \circ \phi_l$. Further, let $\mathcal{D}_{probe} = \{x_1, \dots, x_n\}$ be a probing set, i.e., a set of n images that can be used for the visual description of the extracted CAVs. The textual descriptions of the concepts are based on a predefined and task dependent set of words T . For example, T can contain describing attributes [2], or the top 20.000 words of the English language [13].

Concept Discovery. We describe the embedding of layer l with Concept Activation Vectors (CAVs). A CAV is a vector that points in the direction of a

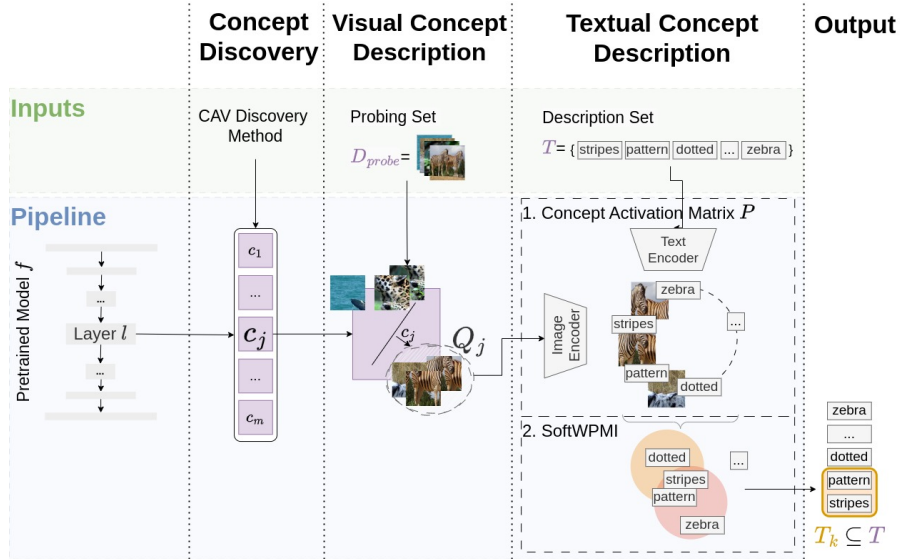


Fig. 2: Overview of our approach to describe the layer l of a pretrained model f . The *inputs* are a concept discovery method, a probing set D_{probe} , and a set of textual descriptions T . We apply *concept discovery* methods to find a set of CAVs, generate a set of *visual concept descriptions* Q_j for each CAV c_j , then *textual concept descriptions* and finally *output* the top- k descriptions $T_k \subseteq T$.

concept learned by the model and is embedded in the feature space of the activations of layer l . The concepts learned at layer l are then represented by a set of m CAVs, $C_l = \{c_{1,l}, \dots, c_{m,l}\}$. We drop the index l in the following when considering only one specific layer. While the proposed method is independent of the underlying concept extraction approach, we follow the approach of Yeh et al. [33] to derive all concepts utilized for a given image classification task.

Visual Concept Description. For the visual description of a given CAV c_j , we follow the former work [33] to derive a set $Q_j \subset D_{probe}$ of most relevant images from the probing set. This approach is illustrated in Figure 3 and will be described in the following. The relevance of an image $x_i \in D_{probe}$ is determined based on the similarity between the CAV $c_j \in \mathbb{R}^k$ and its latent representation at layer l . In detail, consider the latent representation of an image x_i at layer l , which is

$$\phi_l(x_i) =: (\hat{x}_{i,l}^1, \dots, \hat{x}_{i,l}^F) \in \mathbb{R}^{F \times k}.$$

The vectors $\hat{x}_{i,l}^1, \dots, \hat{x}_{i,l}^F$ are called *local feature vectors* of x_i and correspond to the activations of each channel of the convolutional neural network after layer l . We will omit the index l when the connection to the specific layer is clear.

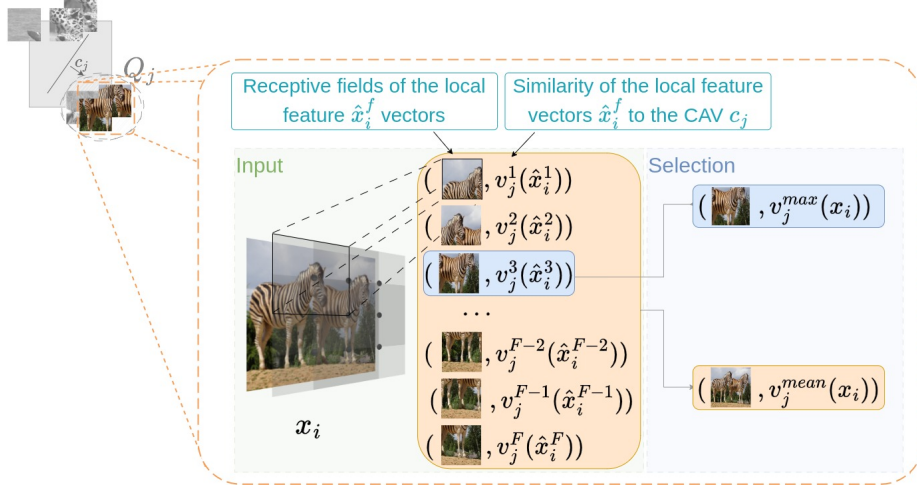


Fig. 3: Selection of the visual representations for a given CAV c_j , compare with Figure 2 column *Visual Concept Description*. The vector $(v_j^1(\hat{x}_i^1), \dots, v_j^F(\hat{x}_i^F))$ represents the concept scores between each receptive field of x_i and the CAV c_j . While [23] select full images based on the mean score of all receptive fields, we also consider the receptive field with the highest concept score. Thus, we improve the visual input of the joint vision-text embedding by cropping x_i to the respective receptive field. This creates a more truthful and more detailed representation of the concepts learned in the hidden space.

For each local feature vector \hat{x}_i^f and each CAV c_j a *concept score*, which measures the similarity based on the scalar product, i.e.

$$v_j^f(\hat{x}_i^f) := \hat{x}_i^{fT} c_j.$$

This leads to a vector $v_j(x_i) \in \mathbb{R}^F$ of F concept scores,

$$v_j(x_i) = (v_j^1(\hat{x}_i^1), \dots, v_j^F(\hat{x}_i^F)) \in \mathbb{R}^F. \quad (1)$$

Following [33], a larger concept score means a higher similarity of the corresponding receptive field of \hat{x}_i^f to the concept encoded by the CAV c_j .

While $v_j(x_i)$ is a vector of similarities, the set of relevant images Q_j is chosen based on scalar values because they can be ordered. Former works such as [23] select full images of $\mathcal{D}_{\text{probe}}$ for the set Q_j . To achieve this, they consider the average over the individual concept scores of the local feature vectors,

$$v_j^{\text{mean}}(x_i) = \frac{1}{F} \sum_{f=1}^F v_j^f(\hat{x}_i^f) \in \mathbb{R}. \quad (2)$$

As we are more interested in the most representative part of an image for a concept, we consider the maximum concept score of all local feature vectors:

$$v_j^{\max}(x_i) = \max_{f \in \{1, \dots, F\}} v_j^f(\hat{x}_i^f) \in \mathbb{R}. \quad (3)$$

Based on these two metrics, we introduce three different strategies to derive a set of most relevant images Q_j from $\mathcal{D}_{\text{probe}}$. Note that the subset Q_j can either contain the full image x_i or a receptive field associated with a local feature vector \hat{x}_i^f . We follow [22] and select the 100 most relevant images.

- F_{mean} : Select the images with the highest $v_j^{\text{mean}}(x_i)$.
- F_{max} : Consider those images with the highest $v_j^{\max}(x_i)$ and choose the respective receptive fields where the maximum is reached.
- $F_{\text{mean} \rightarrow \text{max}}$: Select images like F_{mean} but choose the receptive field with highest concept score $v_j^f(\hat{x}_i^f)$.

We search for the parts of the images with the highest presence of the concept encoded by the CAV. With F_{mean} we select the full images with the highest overall presence of the concept. As a result, the textual descriptions are calculated based on the full images. However, often the model can only see parts of the images at the layer where the CAVs were found. Due to this, and the fact that concepts may be more present in single parts of an image, we apply strategies to find the relevant receptive fields. Using F_{max} we select the receptive field of each image with the highest concept score. We propose $F_{\text{mean} \rightarrow \text{max}}$ to combine the advantages of both strategies. This means that we find the images where the concept is highly present in the full image and reduce the noise introduced by other concepts by selecting the respective receptive field with the highest concept score.

Textual Concept Description. To derive a textual description for the visual descriptions collected in Q_j , we utilize joint text-image embeddings and corresponding image and text encoders $E_{\mathcal{I}}$ and $E_{\mathcal{T}}$ which map from the space of images, \mathcal{I} , and the space of texts, \mathcal{T} , respectively, to a joint feature space. This is, for example, provided by the CLIP model [25]. We compute a similarity matrix P based on the cosine similarity of the text and image embeddings of the textual descriptions set $T = \{t_1, \dots, t_s\}$ and images in Q_j ,

$$P_{ij} = \frac{E_{\mathcal{I}}(x_i)^T E_{\mathcal{T}}(t_j)}{\|E_{\mathcal{I}}(x_i)\|_2 \|E_{\mathcal{T}}(t_j)\|_2}.$$

Intuitively, we want to find the textual descriptions that have a high similarity to all images in Q_j . To do this, we utilize the *Soft Weighted Pointwise Mutual Information* (SoftWPMI) [23], which indicates how well a word describes the mutual information of the representative images. SoftWPMI requires a weighting of the images in Q_j , which is determined by the concept scores. In particular, this vector q_j is calculated depending on the strategy to derive the set of most relevant images Q_j :

$$q_j = \begin{cases} (v_j^{\text{mean}}(x_i))_{x_i \in Q_j} & \text{if } F_{\text{mean}} \\ (v_j^{\text{max}}(x_i))_{x_i \in Q_j} & \text{if } F_{\text{max}}, F_{\text{mean} \rightarrow \text{max}} \end{cases} \quad (4)$$

Finally, we find the subset T_k with the top- k textual descriptions by:

$$T_k := \arg \max_{\hat{T} \subset T: |\hat{T}|=k} \sum_{t \in \hat{T}} \text{SoftWPMI}(t, q_j, P) \quad (5)$$

Note that, in practice, $\text{SoftWPMI}(t, q_j, P)$ is computed for each $t \in T$ separately, and finally, we take the top- k textual descriptions. For the common textual description, we compute the weighted average of the top- k descriptions in the feature space, with the weighting based on the SoftWPMI values. The common representation is then chosen as the textual description in T that is closest to this weighted average. Please note that we set all negative SoftWPMI values in \hat{T} to zero since we are only interested in positive similarities.

Output. The method returns the common description and the subset T_k from the human understandable textual descriptions set T , which are most similar to the concept represented by the CAV c_j .

4 Experiments

Our experimental procedure consists of three stages. First, we utilize CAVs with known concept labels to show that our approach is capable to yield meaningful textual explanations of CAVs. Second, we compare the different mappings F_{mean} , F_{max} , $F_{mean \rightarrow max}$ for the generation of the set of best fitting textual descriptions. And finally, we consider a more complex scenario and explain a set of CAVs extracted from a model where we have no prior knowledge about the underlying concepts.

Table 1: Each row shows the Top-5 textual descriptions of a CAV computed with the proposed approach (ranked from left to right) and the derived common concept description. Each CAV is supposed to represent one class of the CIFAR10 dataset [16]. Imagenet is utilized [6] as \mathcal{D}_{probe} and google20k as the set of textual descriptions, T .

CAV-Label	Common Description	1	2	3	4	5
airplane	aircraft	aircraft	aviation	plane	airplanes	planes
automobile	vehicle	vehicle	vehicles	car	ambulance	automobile
bird	bird	avian	bird	birding	birds	juvenile
cat	cat	cat	kitts	kitty	kitten	katz
deer	deer	grazing	gnu	deer	female	wildlife
dog	dog	puppy	dog	canine	pundit	dug
frog	mating	mating	meal	head	emerging	frog
horse	horse	equine	horseback	horse	horses	equestrian
ship	sailing	sailing	yacht	sail	yachts	sailors
truck	trucks	truck	trailer	trucks	trailers	movers

4.1 Explaining a Set of CAVs with Known Concept Labels

To be able to validate general idea of our approach, we follow Kim et al. [14] and design a set of CAVs where each CAV describes one class of a given data set. We achieve this by generating a set of CAVs after the last convolutional layer of a model and set the number of CAVs equal to the number of classes. It is important to note that the suggested strategy is closely related to the performance of the CLIP model. Hence, a bad classification performance of CLIP directly affects our approach in a negative way.

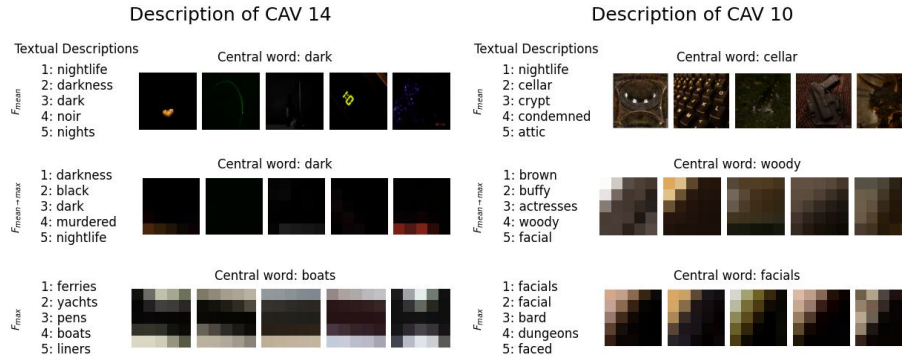
Setup. To make sure that the CLIP model itself performs well in this validation example, we use the datasets CIFAR10 [16] and MNIST [7] which have a zero-shot performance of 96.2% and 87.2%, with the vision encoder CLIP-ViT L/14 from CLIP [25]. For CIFAR10 we adapted a pre-trained ResNet50 [10] and finetuned it. The finetuned ResNet50 reaches an accuracy of 0.94. For MNIST we finetuned a simple ConvNet with 3 layers reaching an accuracy of 0.98. In this experiments we explain the the embedding after the last convolutional layer of the models (ResNet50 and ConvNet). For the set of textual descriptions we use google20k [13]. Details to the MNIST experiments can be found in the appendix.

Results. The results of this experiment for CIFAR10 are displayed in Table 1 (The table for MNIST can be found in the Appendix). The top-5 words, as well as the concept closest to the centroid for each class, are shown. Our approach is able to match each CAV which encodes a class as concept with fitting textual descriptions from the 20.000 textual suggestions given. The exception is the CAV encoding *Frog*. For MNIST our approach finds fitting textual descriptions for all classes except the CAV encoding “one” which is described by *makefile*.

4.2 Concept Discovery and Description

Compared to the class-wise concepts in the previous sections, automatically discovered CAVs usually describe more abstract concepts as colors and shapes. We utilize the approach of [33] to discover a set of CAVs automatically. The final set of CAVs is selected based on a hyper parameter search and the test accuracy of the classification task. The hyper parameter search includes the number of concepts, the threshold value β , and scalars $\lambda_1 > 0$ and $\lambda_2 > 0$. The parameters λ_1 and λ_2 are needed for the utilized concept discovery approach of [33]. They weight the similarity between the concepts and their most relevant images (λ_1) and the pairwise dissimilarity between the concepts (λ_2). Further, we calculated for each class the ConceptSHAP and explanation quality following [33]. The ConceptSHAP gives us an importance value for each CAV with respect to the class. The explanation quality serves as a measure how well a class is described by the set of CAVs discovered. In the following, we first compare the different approaches to select the relevant images, i.e., the receptive field-based approaches and the full image approaches.

Evaluation of Image Set Selection We consider concepts extracted from early layers, where concepts are assumed to be more abstract than in later layers.



(a) Most influential CAV for the class “cat” (b) Second most influential CAV for the class “cat”

Fig. 4: Comparison of the approaches to generate textual descriptions. Shown are the two most influential CAVs for the class “cat” after the first residual block of a ConvMixer [31]. The model was trained on *dark* cats and *light* dogs, a subset of the Cats vs. Dogs dataset [5]. The first approach uses the images with the highest mean activation for the CAV, the second takes the highest receptive fields of the images with the highest mean activation and the third takes the most activated receptive fields of all receptive fields over the whole probing data set. The probing dataset is the validation set from ImageNet [6] and the concept set is google20k [13]

With this we can also evaluate the effect of F_{max} and $F_{mean \rightarrow max}$ on highly correlated concepts and concepts of different degree of abstraction. We further introduce the abstract concept *dark* into the model by performing a classification of cat and dog images, where the training samples consist of dark cats and the bright dogs. We expect the trained model to mainly rely on those features due to the simplicity bias of neural networks [30].

Setup. We trained our model on a modified Cats vs. Dogs (CvD) dataset [5]. The Cats vs Dogs dataset was developed by Kaggle [5] and, following [17,15], we split it by color, such that it consists of *dark* cats and *light* dogs. We call this dataset Dark Cats vs. Dogs (DCvD). In the following we refer to the original and the modified dataset as unbiased and biased dataset. Since all cats are *dark* and all dogs are *light* we make the assumption, that the color is a relevant concept for models trained on this dataset. To validate this we train a ConvMixer [31] with a depth of seven. The ConvMixer reaches an accuracy of 0.93 on the biased data and only an accuracy of 0.69 on the unbiased data (details in the appendix). This difference in accuracy indicates that the model learned to associate the color *black* with cats. We extract the set of CAVs after the first residual block of the model. The derived set consists of 20 CAVs and the classification based on the active and inactive CAVs yields an accuracy of 0.96 on the biased data. The hyper parameters used to learn the set are $\lambda_1 = 0.2$, $\lambda_2 = 0.2$ and $\beta = 0.18$.

After we filter the CAVs where the dot product is over 0.95 we are left with 15 relevant CAVs. As the set of textual descriptions, google20k is used.

Results. Figure 1 shows the two most important CAV from left to right for the class cat. The CAVs are selected by the ConceptSHAP values. For each CAV we display the three approaches to select relevant images based on the concept scores. For each approach the textual descriptions and the top five images from the set of most relevant images are shown. It can be seen for CAV 14 that the approach F_{mean} returns as highest textual description *nightlife* and F_{max} returns *ferries* (See Figure 4a). Only $F_{mean \rightarrow max}$ returns a fitting highest textual description with *darkness*. Looking at the other descriptions we see that F_{mean} also yields similar textual descriptions in the top 5 descriptions. This results in the central word of F_{mean} and $F_{mean \rightarrow max}$, matching our expectations.

For the CAV 10 we can see that all approaches return different textual descriptions (See Figure 4b). F_{mean} returns *nightlife* and F_{max} returns *facials* which are both complex concepts. The approaches recognize different concepts which are relevant for the images. This is neither good nor bad. Only $F_{mean \rightarrow max}$ returns a simple concept with *brown*.

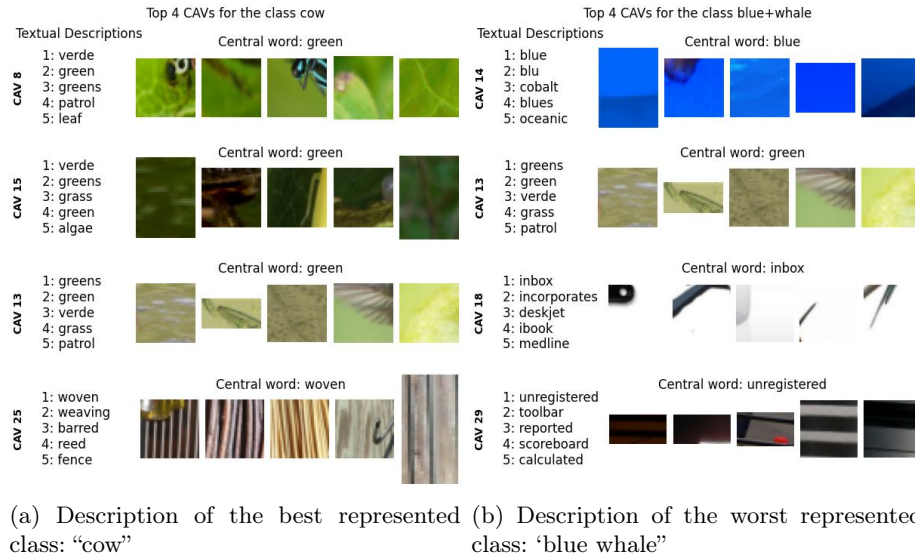


Fig. 5: For each class the textual descriptions and the most activated receptive fields of the CAVs with the strongest influence are shown. The image set was selected by $F_{mean \rightarrow max}$. The set of CAVs describes the hidden representation after the first residual block of a ResNet50 finetuned on Awa2. The probing dataset is the validation set from ImageNet and the concept set is google20k.

Animals with Attributes The objective of this experiment is to explore the performance of our approach for scenarios with increased complexity and to show its potential. The experiment is based on the Animals with Attributes2 dataset [32], which contains 37322 images from 50 different animals.

Setup. We finetuned a ResNet50 on the dataset AwA2 [32] that reaches a test accuracy of 0.9. The concept discovery method found a set of 30 CAVs after the first residual block. The found set of CAVs achieves an accuracy of 0.87 with the hyper parameter $\lambda_1 = 3.1$, $\lambda_2 = 3.1$ and $\beta = 0.02$. After filtering all duplicates 15 CAVs are left, describing the concepts learned by the first residual block.

Results. The results of this experiment can be seen in Figure 5. Here, Figure 5a shows the class which is best described by the set of CAVs and Figure 5b shows the class which is worst described by the set of CAVs. Further, for each class the most influential CAVs ranked by ConceptSHAP are displayed. The descriptions are generated with the $F_{mean \rightarrow max}$ approach. It can be observed that the model strongly connects the concept *green* with the class “cow” (See Figure 5a). The class “blue whale” is connected to the concept *blue* (See Figure 5b). When inspecting the descriptions of the CAVs 18 and 29 a mismatch becomes apparent. The descriptions for those CAVs seem to be hardly related and are not matching to the receptive fields.

5 Discussion

The experiments on the sets of CAVs with the known concept label show that the approach is capable of matching CAVs with the corresponding textual descriptions from a large set of general descriptions. This underlines that our approach is in general capable of identifying joint textual descriptions, even though the performance highly depends on the quality of the utilized joint text-image features space. For the experiment on CIFAR10, one can further see the redundancy in the best-fitting descriptions which is successfully removed by selecting a common concept description (Table 1). Further, one can see the approach’s capabilities to detect biases in the training and/or probing images, e.g., the top five descriptions of the class *ship* are all related to sailing.

For the different approaches to select representative images for given CAVs, the ones using receptive fields help to correctly describe more abstract concepts that especially occur in earlier layers of a neural network (Figure 1). Interestingly, 15 CAVs are detected as relevant, which is more than to separate the concepts of dark and bright. This can be explained by the fact, that dark and bright colors can also occur in the backgrounds of the images and hence the distinction purely based on color concepts is not feasible. However, the relevance of the *dark* concept shows that it is highly relevant to classify cats. The increased focus on abstract concepts when utilizing the receptive fields can also be explained by the nature of the CLIP model. CLIP was trained on images and corresponding captions, where specific colors (e.g., *green*) might be less relevant than the overall image description (e.g., *insect*). In Figure 5b, the CAVs 18 and 29, which are relevant for the class “blue whale”, are examples where the approaches fail to

generate matching textual descriptions. This can be attributed to limitations in the utilized CLIP model. For example, CAV 18 seems to show the concept *white* but the textual descriptions are *inbox*, *incorporate*, This could be improved by applying a more fine-grained selection of the inputs for the joint text-image model or by utilizing other text-image feature spaces.

6 Conclusion

In this work, we proposed an approach to assist the interpretation of CAVs by suggesting textual descriptions and selecting common words for the individual CAVs. To improve the textual descriptions of CAVs found for earlier layers, we consider that for earlier layers of a model, the CAVs do not know the whole input and propose to use receptive fields for the generation of the textual descriptions. Through experiments on sets of CAVs where the underlying concepts are known, we showed that our method is capable of yielding meaningful descriptions for CAVs and that the usage of receptive fields improves the explanation quality for earlier layers. While this research already offers insights into the concept discovery process, further works on the computation of meaningful concepts as well as an exploration of other image-to-text projections are planned. The evaluation of the found textual descriptions regarding human understanding is also a topic for further research. To better understand the behaviour of the model, it would be interesting to extend the results of concept discovery methods with mismatched data. For the description of specific concepts, further insights into the capabilities of joint text-image feature spaces and the needed characteristics of probing sets are interesting for us, as well as the consideration of explicitly fine-tuning text-image embeddings to basic concepts.

Acknowledgements We thank Niklas Penzel for preparing the Dark Cats vs. Dogs (DCvD) dataset and training the corresponding model.

References

1. 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 **10**(7), e0130140 (Jul 2015). <https://doi.org/10.1371/journal.pone.0130140>
2. 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. pp. 6541–6549 (2017)
3. Bau, D., Zhu, J.Y., Strobel, H., Lapedriza, A., Zhou, B., Torralba, A.: Understanding the role of individual units in a deep neural network. Proceedings of the National Academy of Sciences **117**(48), 30071–30078 (2020)
4. Chen, Z., Bei, Y., Rudin, C.: Concept whitening for interpretable image recognition. Nature Machine Intelligence **2**(12), 772–782 (2020)
5. Cukierski, W.: Dogs vs. cats (2013), <https://kaggle.com/competitions/dogs-vs-cats>

6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
7. Deng, L.: The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine* **29**(6), 141–142 (2012)
8. Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., Serre, T.: Craft: Concept recursive activation factorization for explainability. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2711–2721 (2023)
9. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. *Advances in neural information processing systems* **32** (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Hernandez, E., Schwettmann, S., Bau, D., Bagashvili, T., Torralba, A., Andreas, J.: Natural language descriptions of deep visual features. In: International Conference on Learning Representations (2021)
12. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)
13. Kaufman, J.: google-10000-english: A list of the 10,000 most common English words. <https://github.com/first20hours/google-10000-english> (nd), accessed: 2024-03-15
14. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International conference on machine learning. pp. 2668–2677. PMLR (2018)
15. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning not to learn: Training deep neural networks with biased data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9012–9020 (2019)
16. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Toronto, ON, Canada (2009)
17. Lakkaraju, H., Kamar, E., Caruana, R., Horvitz, E.: Discovering blind spots of predictive models: Representations and policies for guided exploration. arXiv preprint arXiv:1610.09064 (2016)
18. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557 (2019)
19. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. *Advances in neural information processing systems* **30** (2017)
20. Moayeri, M., Rezaei, K., Sanjabi, M., Feizi, S.: Text-to-concept (and back) via cross-model alignment. In: International Conference on Machine Learning. pp. 25037–25060. PMLR (2023)
21. Müller, S.G., Hutter, F.: Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 774–782 (2021)
22. Oikarinen, T., Das, S., Nguyen, L.M., Weng, T.W.: Label-free concept bottleneck models. In: The Eleventh International Conference on Learning Representations (2022)

23. Oikarinen, T., Weng, T.W.: Clip-dissect: Automatic description of neuron representations in deep vision networks. In: The Eleventh International Conference on Learning Representations (2022)
24. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. *Distill* (2017). <https://doi.org/10.23915/distill.00007>, <https://distill.pub/2017/feature-visualization>
25. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
26. Reimers, C., Runge, J., Denzler, J.: Determining the relevance of features for deep neural networks. In: European Conference on Computer Vision. pp. 330–346. Springer (2020)
27. Ribeiro, M.T., Singh, S., Guestrin, C.: "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. pp. 1135–1144 (2016)
28. Roth, A.E.: *The Shapley Value: Essays in honor of Lloyd S. Shapley*. Cambridge University Press (1988)
29. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-cam: Why did you say that? arXiv preprint arXiv:1611.07450 (2016)
30. Shah, H., Tamuly, K., Raghunathan, A., Jain, P., Netrapalli, P.: The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems* **33**, 9573–9585 (2020)
31. Trockman, A., Kolter, J.Z.: Patches are all you need? *Transactions on Machine Learning Research* (2023)
32. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence* **41**(9), 2251–2265 (2018)
33. Yeh, C.K., Kim, B., Arik, S., Li, C.L., Pfister, T., Ravikumar, P.: On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems* **33**, 20554–20565 (2020)
34. Yuksekgonul, M., Wang, M., Zou, J.: Post-hoc concept bottleneck models. In: The Eleventh International Conference on Learning Representations (2022)
35. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11975–11986 (2023)
36. Zhang, R., Madumal, P., Miller, T., Ehinger, K.A., Rubinstein, B.I.: Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11682–11690 (2021)

Appendix: Exploiting Text-Image Latent Spaces for the Description of Visual Concepts

7 Additional Examples

7.1 Explaining a Set of CAVs with Known Concept Labels

Table 1: Top-5 closest descriptions are shown from left to right and joint concept description for each CAV. The CAVs encode the classes from MNIST

CAV-Label	Common Description	1	2	3	4	5
zero	circular	circular	ring	rings	circle	oval
one	domains	makefile	hostname	indices	authored	deprecated
two	twenty	twentieth	two	twenty	second	twelve
three	three	three	tres	thirds	iii	third
four	four	four	fourth	fourteen	forty	quad
five	five	five	fifth	sixth	fifteen	fifty
six	sixty	sixty	om	viii	lev	horns
seven	seven	seven	seventh	vii	seventy	hebrew
eight	eight	eight	eighty	infinite	nine	infinity
nine	nine	nine	eight	ninth	ninety	eighty

In Table 1 we present the results of our approach for MNIST [7]. It can be seen, that the approach yields matching descriptions for all CAVs except *one*.

7.2 Evaluation of Image Set Selection

In Figure 1 we present additional CAVs which are important for the class “dog”. For each CAV we show the different approaches to generation of the set of best fitting textual descriptions.

7.3 Animals with Attributes

In Figure 2 we show more textual descriptions for different classes from the AWA dataset [32].

8 Models

ConvNet. For the MNIST dataset, we trained a model with the following specifications: The model included three convolutional layers with inner channel sizes of 32, 64, and 128. We used cross-entropy as the loss function and Adam as the optimizer, with a learning rate of 0.001 and a weight decay of 0.0005. During

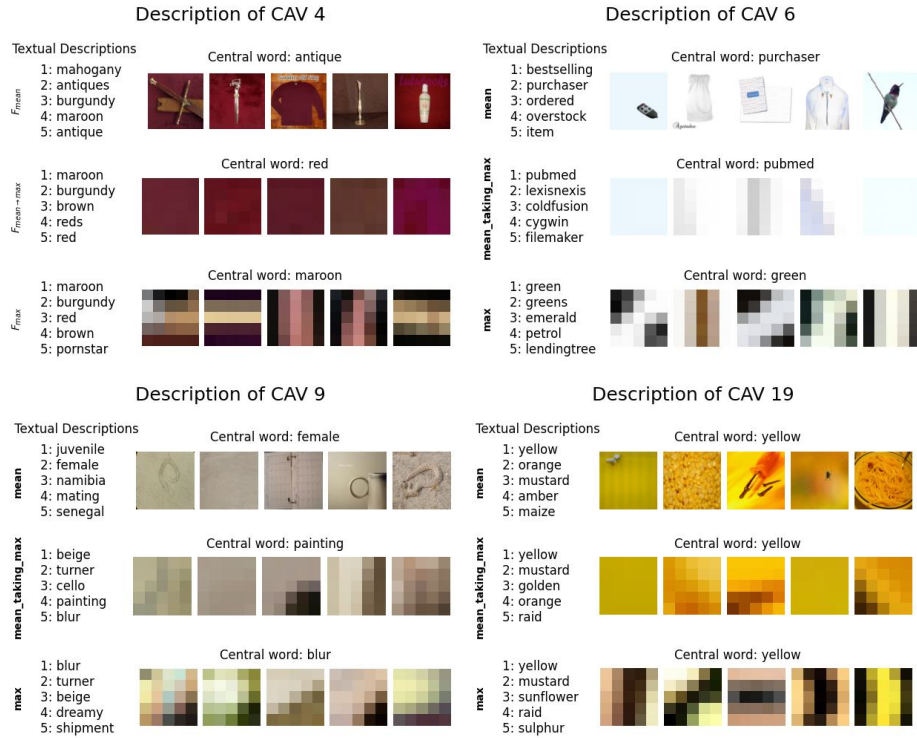


Fig. 1: Comparison of the approaches to generate textual descriptions. Shown are influential CAVs for the class “dog” after the first residual block of a ConvMixer [31]. The probing dataset is the validation set from ImageNet [6] and the concept set is google20k [13]

training, we implemented early stopping. We monitored validation loss with a patience of 10 epochs and training accuracy with a patience of 15 epochs. The maximum number of epochs was set to 1000.

ConvMixer. For the Dark Cats vs Dogs dataset [5], we trained a ConvMixer [31] model with the following specifications: The model had a dimension of 256, a depth of 8, a kernel size of 7, and a patch size of 5. We used cross-entropy as the loss function and AdamW as the optimizer, with a learning rate of 0.001 and a weight decay of 0.0005. We performed data transformations by resizing the images to 128x128 pixels, using TrivialAugment [21] for augmentation, and normalizing the images with mean values of 0.5, 0.5, 0.5, and standard deviation values of 0.5, 0.5, 0.5. During training, we implemented early stopping. We monitored validation loss with a patience of 10 epochs and training accuracy with a patience of 15 epochs. The maximum number of epochs was set to 1000.

Finetuned ResNet50s. For the CIFAR-10 [10] and the Animals With Attributes dataset [32], we finetuned a ResNet50 model [16] with the following

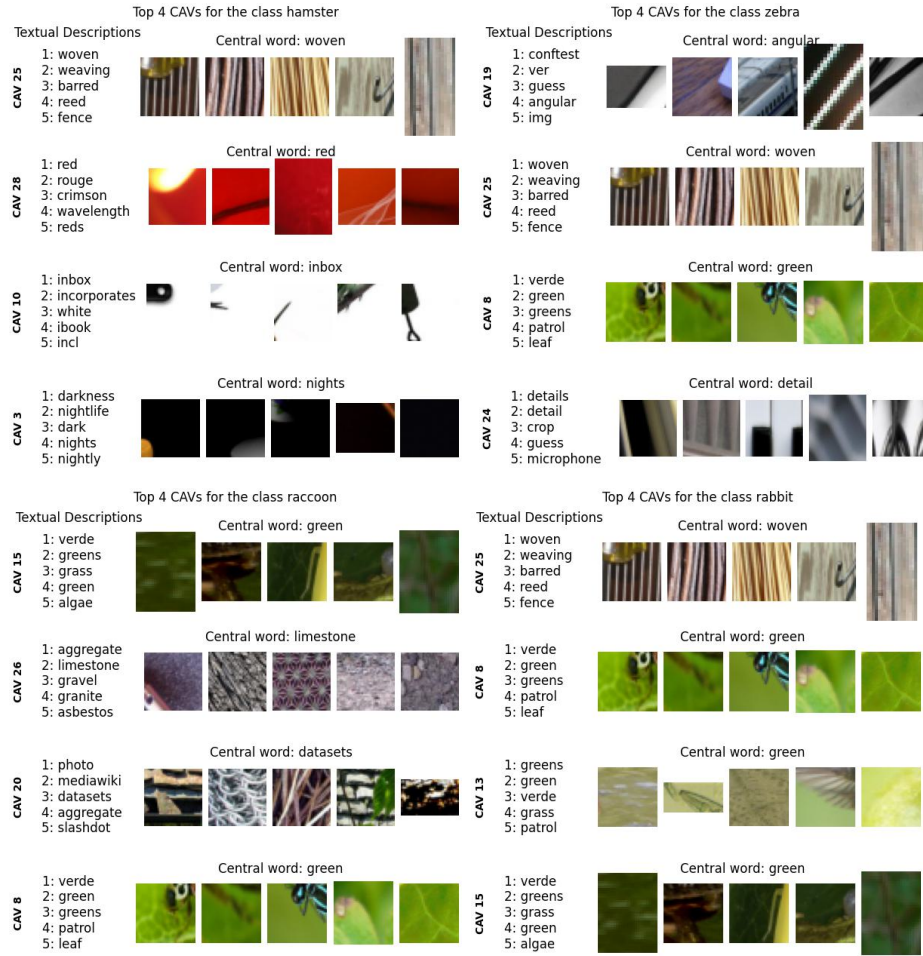


Fig. 2: Description of the classes “hamster”, “zebra”, “raccoon” and “rabbit” according to a set of CAVs. For each class, the textual descriptions and the most activated receptive fields of the CAVs with the strongest influence are shown. The image set was selected by $F_{mean \rightarrow max}$. The set of CAVs describes the hidden representation after the first residual block of a ResNet50 [16] finetuned on AwA2 [32]. The probing dataset is the validation set from ImageNet [6] and the concept set is google20k[13]

specifications: We used cross-entropy as the loss function and Adam as the optimizer, with a learning rate of 0.001 and a weight decay of 0.0005. During training, we implemented early stopping. We monitored validation loss with a patience of 10 epochs and training accuracy with a patience of 15 epochs. The maximum number of epochs was set to 1000.