POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Concept-Based Explanations for Image Classifiers Using Textual Prompts

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Daniele Di Santi

Advisor: Prof. Marco Brambilla

Co-advisor: Riccardo Campi, Matteo Bianchi, Antonio De Santis

Academic year: 2024-2025

1. Introduction

Today, AI models used for computer vision tasks have gained huge popularity and, for this reason, we can find them in many applications, such as medical diagnosis or automotive systems. This is due to recent progress in developing new architectures, especially new *Convolutional Neural Networks (CNNs)* models that perform very well in image classification tasks. However, these models are often treated as black-box systems, making them hard to interpret. Indeed, the internal representation of acquired knowledge and its use remains opaque to developers. Moreover, these models may inadvertently encode and propagate biases present in the training data. The *Explainable AI (XAI)* field aims to enhance the the transparency of complex models by facilitating debugging processes and improving model reliability, especially in high-stakes applications. In the context of CNNs used for classification, a state-of-the-art XAI method is the *Testing with Concept Activation Vectors (TCAV)* [2], along with its extension, *Visual-TCAV* [5]. TCAV is a concept-based method that aims to compute the attribution score of a specific concept for a specific model's prediction (i.e., it computes how much a concept is relevant for the model in returning a target class as a prediction). It is based on *Concept Activation Vectors (CAVs)*, which are encoded representations of human-interpretable concepts within the model's internal embedding space. However, extracting CAVs requires the analysts to collect sets of images representing the concepts, a process that is both labor-intensive and time-consuming. For this reason, these methods do not scale well in scenarios where a large number of concepts must be evaluated. A solution to this problem could be developed based on multi-modal models such as *CLIP* [4]. It is trained to associate images with their captions and produces embeddings from different modal sources in a unique latent space. Once encoded, embedding can be extracted and compared through vector similarity metrics. We can leverage *CLIP*'s text encoder to produce an embedding simply from a textual input describing the concept. Then, we can align this to CNN's latent space, producing a CAV so that we can use it in TCAV and *Visual-TCAV*. This approach enhances the scalability of CAV-based methods by enabling the computation of CAVs directly from textual descriptions, thereby eliminating the need for manually curated image sets.

Moreover, the paradigm of text-derived CAVs can also be applied to object localization. Since CAVs reveal how a CNN understands concepts, these vectors can be exploited to localize objects that the model has not been explicitly trained to recognize, enabling zero-shot localization capabilities.

2. Related Works

2.1. Explainability

The methods for explainability can be divided into ante-hoc and post hoc. On the one hand, the aim is to build AI models interpretable by design, using for instance linear relationships or rule-based algorithms. This leads to the creation of more transparent models, with the drawback of having lower performances w.r.t. other types of architectures. On the other hand, post-hoc XAI refers to explaining black-box models with greater performances after their training phase. Concerning image classification, we have feature-based and concept-based methods. Feature-based methods aim to explain the relationship between the features and the predictions. Techniques such as *Grad-CAM* [6] exemplify this approach by generating visual maps that highlight the regions contributing most significantly to the prediction. Similarly, *Integrated Gradients (IG)* [7] identifies the most relevant pixels contributing to the output. Concept-based methods aim instead to explain models using concepts, visual entities belonging to human knowledge ranging from geometric patterns and colors to body parts, background elements, or entire objects. We can decompose the class prediction as linear sum of concepts or test different layers of our CNN to understand how they recognize those concepts, and how much they use this information in the final prediction. *TCAV* [2] produces a sensitivity score that represents how much the model's prediction depends on that concept. It is both a concept and attribution-based method that can measure how relevant a Concept is for the prediction of a specific class.

2.2. Visual-TCAV

Visual-TCAV [5] extends *TCAV* producing a concept map in addition to the attribution scores. This method can provide an explanation by identifying the region where the model detects the concept, referred to as the concept map, and assigning a relevance score between 0 and 1, known as the concept attribution, which quantifies the concept's importance for the prediction. The concept map is obtained in a way similar to *Grad-CAM* [6], by calculating a weighted sum of the feature maps where the weights correspond to the values of the CAV. To normalize the output within the [0, 1] range, the map is rescaled using a factor derived from the average of the maximum activation values observed during the CAV extraction phase. concept attributions are then calculated by combining the *IG* of the feature maps with the CAV values and the corresponding concept map. Each explanation is derived from the CAV extracted at a specific layer of the CNN, enabling us to analyze how deeper layers function in comparison to shallower ones. To extract a CAV for a particular concept at a given layer, we require two sets of images: one representing the concept and another consisting of random images. At the chosen layer, we compute the feature map activations for both sets and perform mean pooling on each activation to obtain a set of vectors. We then calculate the centroids of the concept and random clusters, referred to as the positive and negative centroids, respectively. The CAV direction is determined by computing the vector difference between the positive and negative centroids.

2.3. Concepts from Text

In *Text-to-Concept* [3], the authors establish a connection between the latent space of *CLIP* and that of a generic CNN, enabling comparison between the visual features extracted by the CNN and the corresponding text embeddings. This is achieved using a linear mapping model, referred to as the *linear aligner*, which is trained to project vectors from one latent space into the other. This simple linear model maps representations in both directions, from *CLIP* to CNN and vice versa, with higher performances for the *CNN-to-CLIP* mapping. Since the *CLIP* text encoder produces a single vector, while a CNN outputs feature maps, the feature maps must first be pooled into a vector to allow for alignment. *CLIP* generates embeddings for both images and text using a shared latent space and is trained to associate images with their corresponding textual descriptions, so embeddings of related visual and textual inputs are close in that space, typically yielding a high cosine similarity. Later, the introduction of *Synthetic CAVs* was proposed as a solution to the scalability limitations of concept-based XAI methods that rely on manually curated image datasets [1]. The idea is to produce synthetic concept images with text-to-image generation models such as Stable Diffusion from a descriptive prompt and use them to train a CAV in the space of a pre-trained CNN. Our approach builds upon *Text-to-Concept*, with several key adaptations. Instead of using CNN-to-*CLIP* mapping, we align from *CLIP-to-CNN*. Our goal is to derive CAVs from text, not just generate concept embeddings. Additionally, we apply the system for model explainability and localization tasks, rather than for zero-shot classification.

3. Methodology

3.1. CAV Extraction

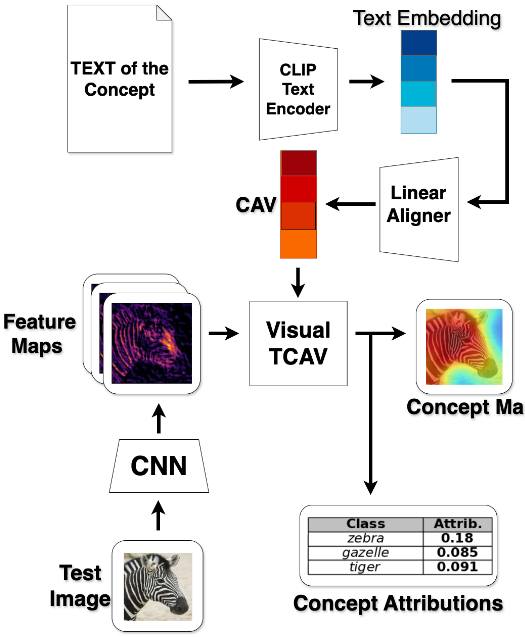
The first step of our solution is the CAV extraction phase. Inspired by *Text-to-Concept* work, we define a list of several templates T_i to better capture the target concept. Each template (e.g., "the image of {}") is completed with the word c describing the concept, resulting in a set of N prompts P_i that are fed in the *CLIP*'s text encoder. We obtain the set of text embeddings V_{P_i} from the *CLIP*'s text encoder, then we use the mean operation to have a single vector V_P that contains the information about the concept. However, this vector is not suitable for *Visual-TCAV* because it is not a proper CAV. Specifically, V_P encodes not only information about the concept but also includes bias introduced by the template's semantics. Since *Visual-TCAV* requires CAVs that represent only the concept information, we compute a refined vector V^c by leveraging the properties of *CLIP*'s text embedding space. In detail, we extract the text embeddings V_{T_i} from the templates T_i. Subsequently, we reduce the list of V_{T_i} into vector V_T with the mean operation. Eventually, we obtain V^c from the subtraction between V_P and V_T. After that, we have a vector that contains the information exclusively about the concept. However, V^c is still in the *CLIP*'s space, so it is needed to map it to the CNN's space. We use the linear aligner to map V^c to the space of a specific layer obtaining the vector p^c. A model with a linear architecture is enough to perform this alignment, but we need to train a specific aligner for each layer.

3.2. Explainability

After the CAV extraction phase, we obtain the CAV p^c, which can be used to produce explanations with *Visual-TCAV*. Each explanation is produced in a specific convolutional layer of the network, utilizing the corresponding feature maps. Given a fixed test image, we compute the concept map as the weighted sum of the feature maps using the values of the CAV as weights. Since we do not have access to the activations of the concept images (as the concept images themselves are unavailable), we cannot compute the rescale factor as done in standard *Visual-TCAV*. Instead, we rescale the concept map using its own maximum and minimum values. The concept attributions are computed in the same manner as in standard *Visual-TCAV*. Hence, we compute the *IG* of the feature maps, we multiply this result by the values of p^c, and we multiply this result by the concept map. This process enables us to generate local explanations, both visual and quantitative, for each layer of the CNN and for each test image, using only a textual representation of the concept. Additionally, we can derive global explanations by analyzing multiple test images and examining the distribution of their concept attributions. For example, we can evaluate how informative the concept "striped" is for predicting the class zebra.

3.3. Zero-Shot Localization

The potential of the CAVs extracted from text extends beyond the explainability field. We can extend the technical meaning of concept from visual patterns to entire objects. Leveraging the concept maps produced by *Visual-TCAV*, we can localize complete objects using bounding boxes. To achieve this, we refine the concept map by applying a threshold (e.g., 0.5), selecting only the most activated regions to delineate the bounding box. Since concepts can be defined through text alone, we are not limited to objects the CNN was explicitly trained to recognize. We use the knowledge learned by the CNN during its training to extract the feature maps from an image, then we use the CAV corresponding to the desired object to refine these feature maps in order to have only the information regarding the object. This technique grants us greater control over the CNN, enabling it to function as an image localizer even for objects outside its original classification set. In this way, we can perform zero-shot localization tasks.



Class	Attrib.
zebra	0.18
gazelle	0.085
tiger	0.091

Figure 1: Schema of CAV extraction with textual inputs and functioning of *Visual-TCAV* with the aligned CAV.

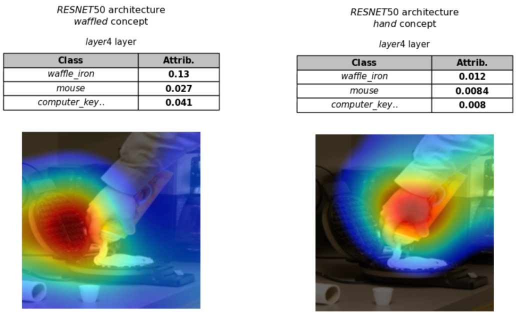
4. Experiments and Results

4.1. Experiment Setup

Regarding our experiments, we used the models *Resnet50* pretrained on *ImageNet-1k* and *Resnet18* pretrained on *CUB-200-2011* as CNNs, we selected the implementation “ViT-B/16” for *CLIP*, and an architecture of Linear Regression for the aligners. We trained the aligners with a subset of ImageNet-1k (25000 images) for the aligners working with Resnet50, and with the entire CUB-200-2011 for the aligners working with Resent18. In both cases, we have greater errors for the aligners working with the deepest layers. We used “layer1”, “layer2”, “layer3”, “layer4” as test layers, corresponding to the last convolutional blocks with channels, respectively, 256, 512, 1024, 2048. We performed our tests with three different datasets: ImageNet-1k, CUB-200-2011, and *COCO*. We worked with *Python* code and *Torch* libraries; therefore, *Visual-TCAV* was reimplemented in *Torch* to maintain a consistent environment with both the reused and newly implemented modules.

4.2. Experiments with ImageNet-1k

In this set of experiments, we used Resnet50 model pretrained on ImageNet-1k. To assess the consistency of our approach, we selected four classes of ImageNet-1 (zebra, waffle iron, honeycomb, and crossword puzzle) characterized by distinctive visual patterns. We computed the concept attributions for various concepts across different layers, and we examined both the resulting concept maps and the distributions of the concept attributions. We obtained consistent concept maps, also for concepts that were not directly relevant to the model’s prediction. Moreover, in the distributions of the concept attributions the characteristic concept was always in the top positions (e.g., concept “striped” for class zebra).



*RESNET*50 architecture
waffled concept

*layer*4 layer

Class	Attrib.
waffle_iron	0.13
mouse	0.027
computer_key..	0.041

(a) Concept Waffled

*RESNET*50 architecture
hand concept

*layer*4 layer

Class	Attrib.
waffle_iron	0.012
mouse	0.0084
computer_key..	0.008

(b) Concept Hand

Figure 2: Explanations with *Visual-TCAV* and CAV from text.

We were able to localize concepts background concepts (e.g. sky, water, and ground), body parts (e.g. legs, head hair, and hand), and entire objects (e.g. pen, bee and person). We then compared the distributions of the concept attributions computed with CAV extracted from concept images (standard *Visual-TCAV*) and the concept attributions computed with CAV extracted from Text (Text *Visual-TCAV*). Since the two methods have different ways to rescale the concept map, affecting the resulting concept attributions, we set to 1 all the values greater than zero in the concept map. While this adjustment slightly increases the typical values of the concept attributions, our focus here is on comparing the relative distributions rather than interpreting the absolute magnitudes. We observed a good similarity between each distribution, suggesting how CAVs from text and CAVs from images can be interchangeable.

4.3. Experiments with CUB-200-2011

We used a ResNet-18 model pretrained on the CUB-200-2011 dataset to evaluate our method in more complex scenarios. This dataset contains 200 bird species, with annotated attributes for each image that can be used as semantic concepts. We initially selected four bird classes characterized by a dominant color and analyzed the distribution of concept attributions related to different color attributes. The results showed that the dominant color consistently ranked among the top concepts, and the corresponding concept maps focused on the body regions exhibiting that color.
We then extended the experiment to explore more complex concepts beyond primary color. Specifically, we selected four bird species distinguished by features such as unique beak shapes, wing colors, and neck shapes. For each class, we defined a list of relevant and irrelevant concepts and computed the attributions for each. By analyzing the attribution distributions, particularly in the deepest layer, we found that relevant concepts consistently received higher scores than irrelevant ones. Furthermore, our results suggest that it is possible to extract concept activation vectors (CAVs) representing more composite concepts (e.g., black and white body) using only textual input.

4.4. Experiments with COCO

Concerning this set of experiments, we used “layer3” and “layer4” of Resnet50 pretrained with ImageNet-1k. To evaluate the performance of our method on a zero-shot Localization task, we selected the COCO dataset, which provides ground-truth bounding boxes for each image. This allows for a quantitative assessment of localization accuracy. We specifically chose a subset of objects that are not among the target classes of ImageNet-1K, in order to test the zero-shot capabilities of our approach. Two types of experiments were conducted: the Pointing Game and the Intersection over Union (IoU) evaluation. We implemented the Pointing Game to assess how accurately our model can indicate the position of objects in the images. We extract a bounding box with 0.9 as threshold and check if the box is fully inside the ground truth box. We then calculated the accuracy based on how often the predicted boxes fell within the ground-truth annotations. In the second experiment, we evaluated the model’s ability to generate bounding boxes that cover the full extent of the object, rather than just indicating its position. This was done by computing the Intersection over Union (IoU) between the predicted and ground-truth boxes. We tested with thresholds of 0.5 and 0.1 to explore how performance varied with different levels of refinement. The results showed a good accuracy (values from 0.6 to 0.8) in most cases for the Pointing Game and average scores (values around 0.5) for the IoU score.

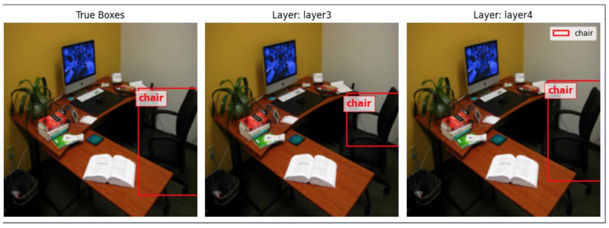


Figure 3: Zero-shot localization with object chair.

5. Results and Discussions

The results of our experiments demonstrate that CAVs extracted from text can be effectively used in *Visual-TCAV* method, producing consistent explanations with respect to both concept maps and concept attributions. The relevant concepts receive high attribution scores, while the irrelevant ones present lower values. In most cases, the concept maps focus on the correct region occupied by the concept, even when the concept is not directly relevant for the prediction. The explanations derived from text-based CAVs are similar to those obtained from image-based CAVs. Our approach allows for the evaluation of a wide range of concepts on the CNN, including those not directly related to a specific prediction. Results on the CUB-200-2011 dataset further confirm the method’s ability to generate consistent CAVs for complex concepts and assign higher attributions to those that are relevant.

Regarding prompt engineering, our findings align with those reported in the Text-to-Concept framework. Specifically, the use of predefined templates from Text-to-Concept improves both the accuracy of the concept vectors and the quality of the resulting explanations. Our results also confirmed the established theory regarding the types of feature learned by the different CNN layers. In fact, the deeper layers were able to recognize more complex concepts compared to the shallow layers that instead produced noisy or missing concept maps for those complex concepts.

Furthermore, these CAVs can be used to simplify the performing of localization tasks, in fact, we can simply use an image classifier instead of having a model trained with an annotated dataset with bounding boxes. The localization is not limited to objects that are targets of the classifier, but we can localize any object that has features observed by the CNN during its training. However, we still have some limitations for this novel approach. The concept map does not have a robust rescaling method, and this may lead to less precise maps and localizations. A linear architecture for the aligner may be a limitation, because it may lead to a partial mapping or to a high error, where some concepts are wrongly mapped or not mapped at all. We think that our work may be a solution to make *Visual-TCAV* more scalable and to simplify the approach for the localization tasks. In future work, it is possible to provide a solution to the aforementioned limitations and to explore further potential applications of CAVs extracted from text.

6. Conclusions

We have proposed a solution that can produce post-hoc and concept-based explanations for image classifiers using only a textual input. Our method extracts CAVs from text by leveraging *CLIP*, aligning them with any layer-specific latent space of a CNN. This solution makes the phase of CAV extraction scalable, which in turn enhances the scalability of the entire *Visual-TCAV* method. Furthermore, we proposed another use of CAVs extracted from text: zero-shot localization. We demonstrate that a CNN can implicitly localize objects outside its target class set, and that a CAV can guide this localization by filtering relevant information within the extracted feature maps.

References

[1] R. Campi, S. Borrego, A. De Santis, M. Bianchi, A. Tocchetti, and M. Brambilla. Towards synthetic concept activation vectors via generative models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, pages 2720–2728, June 2025.

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

[3] M. Moayeri, K. Rezaei, M. Sanjabi, and S. Feizi. Text-to-concept (and back) via cross-model alignment. *arXiv preprint arXiv:2305.06386*, 2023.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[5] A. D. Santis, R. Campi, M. Bianchi, and M. Brambilla. Visual-tcav: Concept-based attribution and saliency maps for post-hoc explainability in image classification, 2025.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[7] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017.