Visual-TCAV: Explainability of Image Classification through Concept-based Saliency Maps

Anonymous Author(s)
Affiliation
Address
email

Abstract

Convolutional Neural Networks (CNNs) have seen significant performance improvements in recent years. However, due to their size and complexity, their decision-making process remains a black-box, leading to opacity and trust issues. State-of-the-art saliency methods can generate local explanations that highlight the area in the input image where a class is identified but do not explain how different features contribute to the prediction. On the other hand, concept-based methods, such as TCAV (Testing with Concept Activation Vectors), provide global explainability, but cannot compute the attribution of a concept in a specific prediction nor show the locations where the network detects these concepts. This paper introduces a novel explainability framework, Visual-TCAV, which aims to bridge the gap between these methods. Visual-TCAV uses Concept Activation Vectors (CAVs) to generate saliency maps that show where concepts are recognized by the network. Moreover, it can estimate the attribution of these concepts to the output of any class using a generalization of Integrated Gradients. Visual-TCAV can provide both local and global explanations for any CNN-based image classification model without requiring any modifications. This framework is evaluated on widely used CNNs and its validity is further confirmed through experiments where a ground truth for explanations is known.

1 Introduction

Recent advancements in Deep Neural Networks (DNNs) have revolutionized the field of Artificial Intelligence, and Convolutional Neural Networks (CNNs) have emerged as the state-of-the-art for image classification due to their ability to learn complex patterns and features within images. However, as the performance of these models has grown significantly over recent years, their complexity has also increased. Consequently, it became a challenge to understand how these models produce their classifications. This led to the widespread use of the term *black-box* to describe these models, as only their inputs and outputs are known, while their internal mechanisms remain too complex for humans to comprehend. The black-box problem results in a lack of transparency [29], which can undermine trust in AI-based systems [12]. Indeed, blindly trusting AI poses serious ethical dilemmas, especially in critical fields such as healthcare or autonomous driving in which image classification systems are becoming increasingly employed [28, 3]. Additionally, debugging black-box models and identifying biases becomes difficult without comprehending the process they use to make predictions. To this end, the field of Explainable Artificial Intelligence (XAI) has made significant progress in developing techniques for producing explanations of AI decisions. However, comprehending the specific features or patterns that networks identify in an image and their precise impact on the prediction remains a challenge. State-of-the-art approaches for local explainability (i.e., for individual predictions) use saliency maps to locate where a class is identified in an input image, but they can’t explain which features led the model to its prediction. For instance, when analyzing an image of a golf ball, these saliency methods cannot determine whether the golf ball was recognized by the spherical shape, the dimples, or some other feature. Striving to cover this need, Kim et al. [11] introduced TCAV (Testing with Concept Activation Vectors), a concept-based method that can discern whether a user-defined concept (e.g., dimples, spherical) correlates positively with the output of a selected class. However, TCAV is designed exclusively for global explainability (i.e., for explaining the general behavior of a model) and therefore cannot measure the influence of a concept in a specific prediction or show the locations within the input images where the networks recognize these concepts.

In this article, we introduce a novel explainability framework, Visual-TCAV, which integrates the core principles of both saliency methods and concept-based approaches while aiming to overcome their respective limitations. Visual-TCAV can be applied to any layer of a CNN model whose output is a set of feature maps. Its main contributions are: (a) it provides visual explanations that show where the network identifies user-defined concepts; (b) it can estimate the importance of these concepts to the output of a selected class; (c) it can be used for both local and global explainability.

2 Related Works

In recent years, there has been a significant increase in the body of work exploring the explainability of black-box models. For CNN-based image classification, state-of-the-art methods primarily focus on providing explanations via saliency maps. These heatmaps highlight the most important regions of the input image and therefore can be used to gain insights into how a model makes its decisions. One approach for generating such visualizations involves studying the input-output relationship of the model by creating a set of perturbed versions of the input and analyzing how the output changes with each perturbation. Notable contributions to this approach include Local Interpretable Model-Agnostic Explanations (LIME) [17], which uses random perturbations, and SHapley Additive exPlanations (SHAP) [14], which estimates the importance of each pixel using Shapley values. A different approach that instead tries to access the internal workings of the model was originally proposed by Simonyan et al. [22] and consists of generating saliency maps based on the gradients of the model output w.r.t. the input images. This idea led many researchers [24, 23] to investigate how to exploit gradients to produce more accurate saliency maps. Selvaraju et al. [20] proposed a method named Gradient-weighted Class Activation Mapping (Grad-CAM) that extracts the gradients of the logits (i.e., raw pre-softmax predictions) w.r.t. the feature maps. It then uses a Global Average Pooling (GAP) operation to transform these gradients into class-specific weights for each feature map and performs a weighted sum of these feature maps to produce a class localization map, a saliency map that highlights where a class is identified. Grad-CAM has gained considerable attention and is extensively used for explaining convolutional networks. However, Sundararajan et al. [25] demonstrated that gradients can saturate, leading to an inaccurate assessment of feature importance. To address this issue, they introduced Integrated Gradients (IG), a method that calculates feature attribution by integrating the gradients along a path from a baseline (e.g., a black image) to the actual input image. Notable contributions of IG and its variants [10, 16, 30] include the ability to provide fine-grained saliency maps (i.e., each pixel has its attribution) and adherence to the axiom of completeness (i.e., the sum of the attributions of all pixels equals the logit value).

While saliency methods are effective and intuitive, they might not always provide a complete picture of why a model made a certain decision. This is because these methods perform class localization, but cannot explain which features led the model to recognize the highlighted class. Furthermore, these techniques rely on per-pixel importance which can't be generalized across multiple instances, as the position of these pixels is only meaningful for a specific input image. Consequently, they can only explain one image at a time, preventing them from providing global explanations. To overcome these limitations, Kim et al. [11] proposed Testing with Concept Activation Vectors (TCAV), a method that investigates the correlations between user-defined concepts and the network's predictions using a set of example images representing a concept. For instance, images of stripes can be used to determine whether the network is sensitive to the "striped" concept for predicting the "zebra" class. This is accomplished by calculating a Concept Activation Vector (CAV), which is a vector orthogonal to the decision boundary of a linear classifier, typically Support Vector Machines (SVMs), trained to differentiate between the feature maps of concept examples and random images. From this, a TCAV score for any concept and model's layer can be computed using the signs of the dot products between the CAV and the gradients of the loss w.r.t. the feature maps produced by images of a selected class. TCAV is effective in detecting specific biases in neural networks (e.g., ethnicity-related) and can be

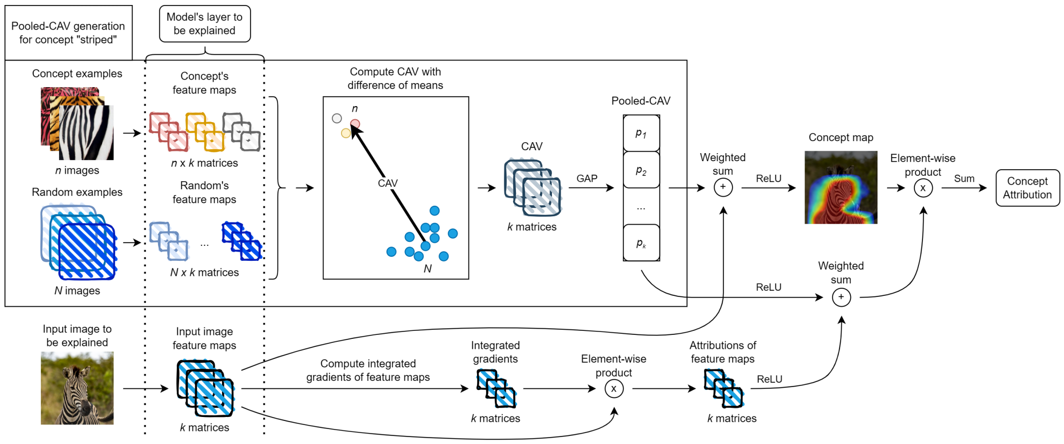


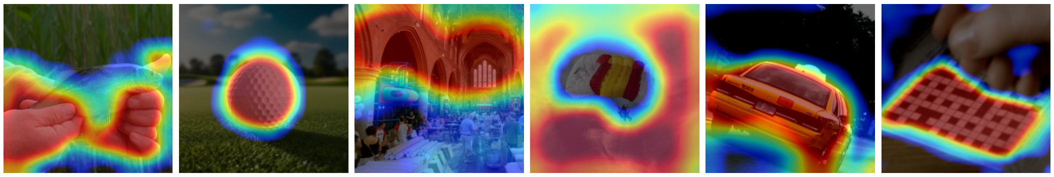
Figure 1: The Visual-TCAV process for generating local explanations. A Pooled-CAV is computed using the feature maps of user-defined concept examples and random images. A concept map is then produced through a weighted sum of the Pooled-CAV and the image's feature maps. Finally, a concept attribution is obtained by extracting the IG attributions of the neurons that the concept activates using the Pooled-CAV and the concept map, which is used as a spatial mask.

considered complementary to saliency methods. Indeed, while saliency methods apply exclusively to individual predictions, TCAV can only provide global explanations. However, TCAV does not provide any information about the locations where concepts are identified within the input images. This makes it challenging to assess whether a high score can truly be attributed to the intended concept and not to a related one. Moreover, TCAV computes the network's sensitivity to a concept, but not the magnitude of its importance in the prediction as the score only depends on the signs of the directional derivatives. For instance, "white" and "dimples" concepts might have identical TCAV scores for the "golf ball" class, even if one contributes substantially more to the prediction.

TCAV has received attention within the XAI community, leading to various extensions [5, 8] and applications [13, 2]. While our study focuses on user-defined concepts, unsupervised approaches have also been proposed. Ghorbani et al. [7] introduced Automatic Concept Extraction (ACE), a method that automatically extracts concepts from images for applying TCAV. This is accomplished by segmenting input images and subsequently clustering their activations. Building upon ACE, Zhang et al. [31] proposed Invertible Concept-based Explanations (ICE). This extension uses non-negative CAVs derived from non-negative matrix factorization and can also be used to explain locally by associating extracted concepts with a relevant area in the input image. Later, Bianchi et al. [1] proposed an unsupervised method for visualizing the entire feature extraction process of CNNs. They perform layer-wise clustering of similar feature maps to extract a set of concepts for each layer to which they assign a descriptive label through crowdsourcing. This approach provides local and global explanations, but the reliance on crowdsourcing can pose a practical challenge. Furthermore, these unsupervised approaches may provide opaque explanations. This is because, when the extracted image regions contain overlapping concepts (e.g., dimples, spherical, and white in a golf ball), it remains unclear which concepts the network has learned to recognize or considers more important.

3 Visual-TCAV

This section presents the methodology of our framework, Visual-TCAV, which is designed to explain the outputs of image classification CNNs using user-defined concepts. Local explanations can be generated considering any layer and consist of two key components. The first is the *Concept Map*, a saliency map that serves as a visual representation of the areas where the network has recognized the selected concept in the input image. The second is the *Concept Attribution*, a numerical value that estimates the importance of the concept for the output of a selected class. Figure 1 illustrates the pipeline for generating a local explanation. For global explanations, the process is replicated across multiple input images. The concept attributions for each image are then averaged to quantify how the concept influences the network's decisions across a wide range of inputs.



(a) Hands (b) Dimples (c) Arches (d) Sky (e) Car (f) Chequered

Figure 2: Examples of class-independent concept maps for various input images and concepts.

3.1 CAV Generation and Spatial Pooling

Similarly to the TCAV framework, the initial step of our method consists of computing a Concept Activation Vector (CAV) from a set of example images representing a user-defined concept, and a set of negative examples (e.g., random images). Specifically, we use the *Difference of Means* method, proposed by Martin and Weller [15], to compute the CAV. They demonstrated that this approach produces CAVs that are more resilient to perturbation and consistent than logistic classifiers or SVMs. As the name suggests, this method uses the arithmetic mean to determine the centroids of both the concept's activations and the activations of random images. Subsequently, it directly computes the CAV as the difference between these centroids.

Since we are interested in identifying which feature maps are activated by the concept, irrespective of its location within the example images, we apply a Global Average Pooling (GAP) operation on the obtained CAV. The result is a vector of scalar values whose length is equal to the number of feature maps of the layer under consideration. Each vector element is associated with a feature map, and its raw value approximates the degree of correlation between that feature map and the concept. Moving forward, we will refer to this vector as the *Pooled-CAV*.

3.2 Concept Map

From the Pooled-CAV, we can construct a concept map that locates a concept (c) within any input image to be explained. This is achieved by performing a weighted sum of the feature maps ($fmaps_k$) of the input image, with the weights being the Pooled-CAV values (p_k^c). Equation (1) shows how to compute a raw concept map (M_{raw}^c). We also apply a ReLU function after the weighted sum because we are only interested in the image regions that positively correlate with the concept. The computation is similar to Grad-CAM's equation, with the difference that we use the elements of the Pooled-CAV as weights instead of the global-average-pooled gradients.

$$M^{c,raw} = ReLU\Big(\sum_k p_k^c \cdot fmaps_k\Big) \tag{1}$$

We refer to this concept map as *raw* due to the absence of a scale factor (i.e., a maximum value) that would allow us to compare the degree of activation of the concept map across different concepts, input images, and model layers. To this end, we derive a concept map's scale factor from the example images the user provided, which represent an ideal concept. Formally, we use Equation (2) to calculate the scale factor (s_c) as the maximum value of a hypothetical concept map, computed using the centroid (C^c), derived from the mean of the feature maps of the example images for a concept (c). Subsequently, we normalize the raw concept map by dividing it by the scale factor (s_c) and limiting the values to a unitary maximum, as shown in Equation (3). An epsilon (ε) is added to the denominator to prevent division by zero.

$$s_c = \max\Big(ReLU\Big(\sum_k p_k^c \cdot C_k^c\Big)\Big) \tag{2}$$

$$M_{ij}^c = \min\Big(1,\ \frac{M_{ij}^{c,raw}}{s_c + \varepsilon}\Big) \quad \forall i, j \tag{3}$$

By overlaying the *normalized* concept map (M^c) on the input image, we can generate a class-independent visualization (examples are shown in Figure 2) that highlights the region of the image where the network recognized the concept. This allows us to know, for any input image, the concept's location and its degree of activation w.r.t an ideal concept defined by the user. Additionally, the concept map can provide a direct validation for the learned CAV, without requiring activation maximization techniques or sorting images based on their similarity to the CAV.

3.3 Concept Attribution

Once we acquire a set of concepts, we can gain insights into the network's decision-making process by measuring the attribution of these user-defined concepts towards the raw predictions, also known as the logits. For instance, if the "church" class is predicted with a certain logit, we aim to quantify how much of this value is attributable to the "pews" concept, the "fresco" concept, and so on. More specifically, given an input image and a layer, we compute the attributions of the activations (i.e., the values of the feature maps) to the logit of a specific target class. Subsequently, we utilize the Pooled-CAV to approximate which activations are attributable to a certain concept, and then we extract and sum these attributions. The attributions of a layer's activations can be computed through a generalized variant of the IG approach which computes the integrated gradients of a target class's logit w.r.t. the feature maps, instead of the input image. Specifically, we calculate the gradients along a straight-line path from zero-filled matrices to the actual feature maps and then approximate the integral using the Riemann trapezoidal rule. In our experiments, we consistently used 300 steps, which are sufficient to approximate the integral within a 5% error margin, as shown by Sundararajan et al. [25]. We then calculate the raw attributions by multiplying the integrated gradients with the feature maps, as shown in Figure 1. Since IG respects the completeness axiom regardless of which layer is considered as input, the attributions add up to the logit value of the target class, within the approximation error. A ReLU is then applied to extract positive attributions. These attributions are on the same scale as the raw logits, which can make their interpretation difficult. To obtain a comprehensible unitary scale, we normalize the attributions so that their sum equals a normalized logit, not the raw one. These normalized logits are obtained by applying a ReLU, followed by [0,1] rescaling to retain their relative ratios.

To estimate the attribution of a concept (c), we can utilize the Pooled-CAV to perform a weighted sum of the normalized attributions ($A^{t,norm}$). Before this summation, we apply a ReLU and [0,1] rescaling to the Pooled-CAV (p^c) so that we extract gradually less attribution for feature maps that are less correlated with the concept. The rationale behind using the ReLU is to discard the attribution of feature maps that show a negative correlation with the concept. In other words, if a certain feature map is activated by other non-correlated features, we discard its attribution. Finally, as shown in Equation (4), we obtain the $ConceptAttribution$ for a concept (c) and a target class (t) by summing all values of an element-wise multiplication of the weighted attributions and the concept map (M^c), which is used as a spatial mask. This enables us to discard the attributions of activations related to the regions within the input image where the concept is not present or was not recognized.

$$ConceptAttribution_{c,t} = \sum_{i,j} M^c_{ij} \cdot \Big(\sum_k ReLU(p_k^{c,norm}) \cdot A_k^{t,norm} \Big)_{ij} \tag{4}$$

The concept attribution is a per-concept metric of importance, meaning that two concepts can have significantly different attributions even if they are recognized in the same location of the input image, resulting in similar concept maps. For instance, considering the "zebra" class, the attribution of the "striped" concept could be significantly different from the attribution of the "fur" concept. This distinction is achieved by focusing not on per-pixel attributions but on the attributions of the activations produced by the neurons responsible for recognizing these two concepts. Moreover, since the attribution of a concept is independent of its location, we can average it across multiple input images to provide a quantitative measure of the overall importance of that concept for that particular class, thus providing a global explanation. For instance, we can calculate a global attribution of the "striped" concept for the "zebra" target class by averaging the attribution of "striped" across a large number (e.g., 200) of images containing zebras.

4 Experiments and Results

In this section, we present the results of applying Visual-TCAV to the following convolutional networks pre-trained on the ImageNet [6] dataset: GoogLeNet [26], InceptionV3 [27], VGG16 [21], and ResNet50V2 [9]. Examples of "striped", "zigzagged", "waffled", and "chequered" concepts are

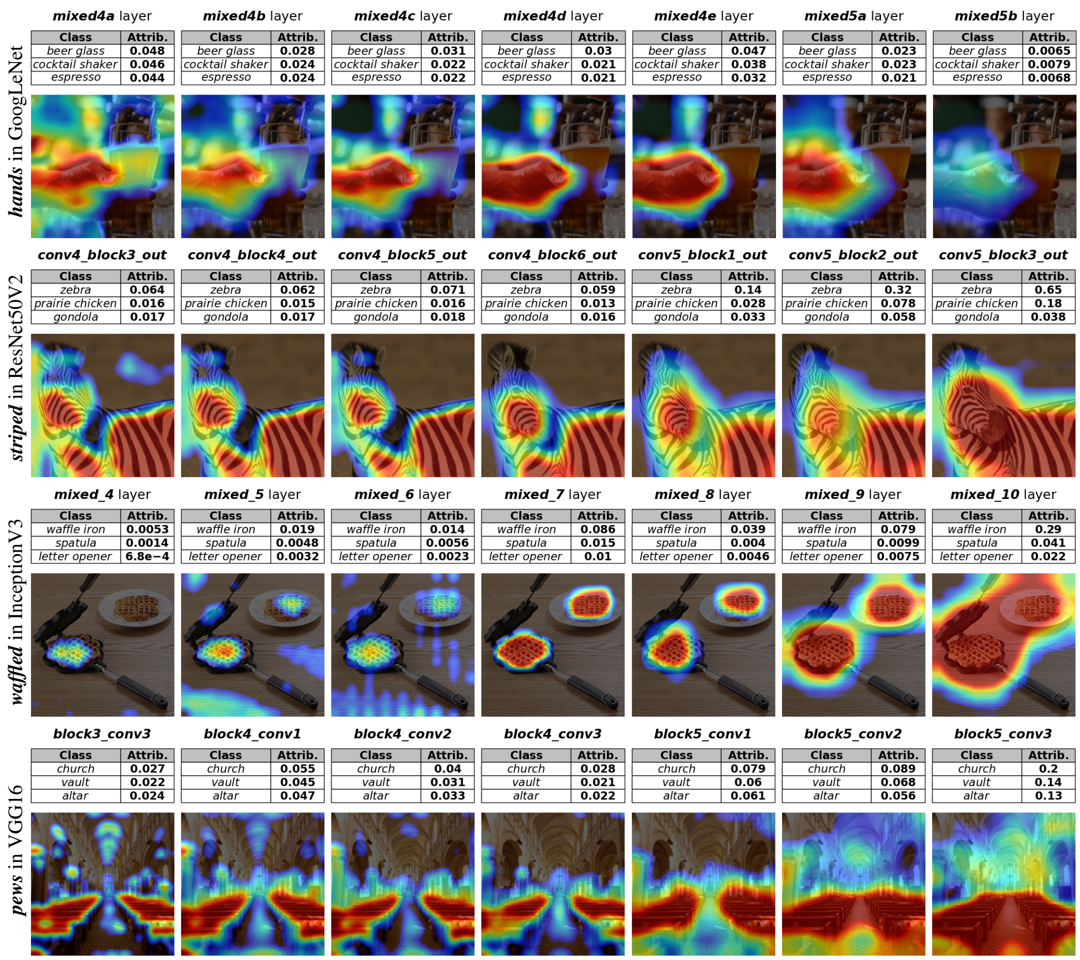


Figure 3: Examples of layer-wise local explanations for various concepts and networks. We compute the attribution of each concept for the top three predicted classes and the last seven layers.

sourced from the Describable Textures Dataset (DTD) [4], while "pews" and "fresco" are generated through Stable Diffusion v1.5 [18] (more on this in Appendix E). Other concepts are obtained from popular image search engines. Similarly to TCAV, we use a minimum of 30 example images per concept and 500 random images as negative examples, as suggested by Martin and Weller [15].

Our experiments are conducted on an Intel i7 13700k with an Nvidia RTX 4060Ti 16GB, and 32 GB of DDR5 RAM. The software runs on TensorFlow 2.15.1, CUDA 12.2, and Python 3.11.5. Local explanations, with 300 steps and seven layers, take less than a minute, while global explanations with 200 class images, 300 steps, and seven layers, can take anywhere from 5 to 20 minutes, depending on the model. For global explanations, the computation time remains nearly constant regardless of the number of concepts processed simultaneously. The official implementation is available in our GitHub repository: *removed for anonymity, see supplemental material .zip file*.

4.1 Local Explanations

In Figure 3, we provide local explanations for various concepts. While concept maps are class-independent, the attribution of each concept depends on the class considered. We examine the top three predicted classes in our examples and apply Visual-TCAV to a subset of the CNNs' layers. On one hand, we can observe a substantial increase in attributions in deeper layers, reaching a peak in the final layer, which holds the most information about the importance of each concept for a specific class, given its proximity to the output. On the other hand, the most accurate concept maps are typically found in slightly earlier layers due to their neurons having smaller receptive fields.

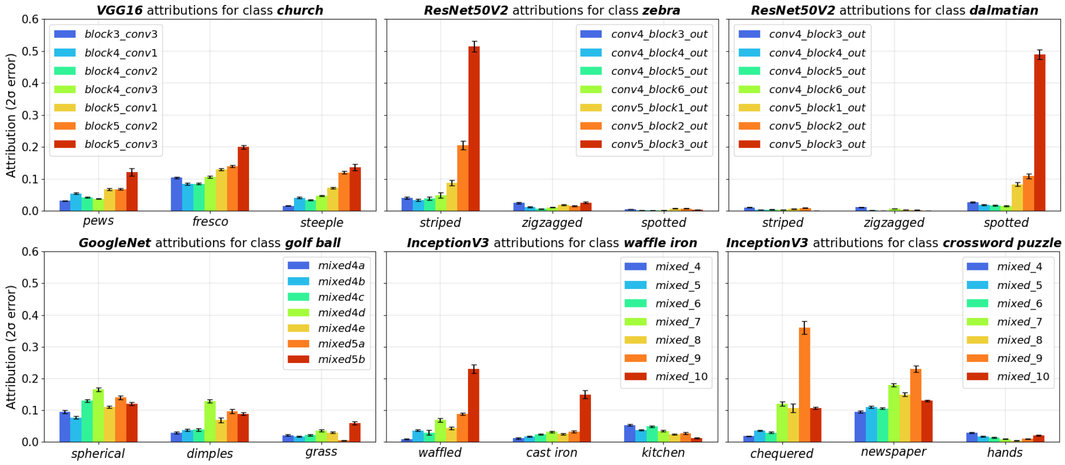


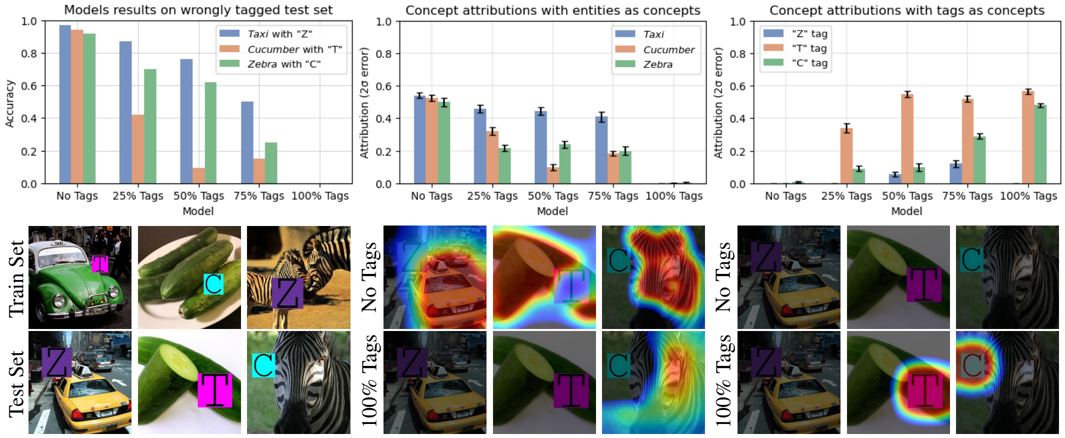
Figure 4: Results of global explanations for a variety of concepts, classes, and networks. Each bar chart reports the attributions of three concepts for a given class, throughout the last seven layers of each network. The attributions of each concept are computed across 200 images of the selected class. Although the theoretical limit of concept attributions is 1.0, the scale in our charts only extends to 0.6. This is based on our empirical observations, which rarely identified concepts with a global attribution exceeding this value.

Furthermore, these layer-wise explanations enable us to identify when specific concepts are recognized within the network. For instance, the "waffled" concept does not significantly activate the initial layers of InceptionV3, but it is recognized by deeper layers with a considerable attribution in the final one. We also observe that the "hands" concept is detected mainly by earlier layers and contributes only marginally to the score of the top classes for the analyzed image. This observation aligns with the common intuition that "hands" are not class-discriminative in this particular case for the classes "beer glass", "cocktail shaker", and "espresso". In contrast, the "striped" and "pews" concepts significantly activate the final layer and substantially contribute to the predictions, although with different magnitudes of importance. In the case of the "zebra" image, for instance, the network's decision is largely influenced by the "striped" concept, which accounts for more than half the logit value of the "zebra" class. This concept also has a notable impact on the "prairie chicken" class and a marginal one on the "gondola" class, probably since gondoliers usually wear striped t-shirts. More examples of local explanations can be found in Appendix C.

4.2 Global Explanations

The concept attribution is a per-concept metric of importance, hence we can derive global explanations by aggregating this attribution across a wide range of input images of a selected class. In our experiments, we utilize 200 images per class for each global explanation. For concepts that are inherently part of the class (e.g., "striped" for "zebra" or "dimples" for "golf ball"), we can directly use any image representing that class. On the other hand, for concepts that appear sporadically, we only use images where the concept is present. For instance, we only use images of church interiors for "pews" and "fresco" concepts, and images of church exteriors for the "steeple" concept. This ensures that the explanations are independent of the frequency of the concept's appearance in the class images.

The results are shown in Figure 4. The attributions match our intuitive expectations, considering, for instance, the importance of the "striped" concept for "zebra" or "spotted" for "dalmatian". Moreover, the final layer typically provides the highest attribution, which is expected for class discriminative concepts. However, there are instances, such as "chequered" and "newspaper" for "crossword puzzle", where concepts recognized in the earlier layers have a greater impact on the network's prediction. We observe a more gradual increase in attribution in VGG16 and GoogleNet, compared to InceptionV3 and ResNet50V2. This could be attributed to the depth of the latter networks, which means they perform more convolution operations that could potentially lead to a more complex feature extraction between the analyzed layers. More examples of global explanations are provided in Appendix D.



(a) Models results (b) Visual-TCAV for entities (c) Visual-TCAV for tags

Figure 5: The results of the validation experiment. The upper section of the figure shows the test results and the concept attributions for both entities and tags across all models. The lower section provides examples of tagged images and concept maps for the no tags model and 100% tags model.

4.3 Validation Experiment with Ground Truth

We conduct a validation experiment to evaluate the effectiveness of Visual-TCAV. In this experiment, we train convolutional networks in a controlled setting, where ground truth is known, and assess whether the Visual-TCAV attributions match this ground truth. For this purpose, we create a dataset of three classes – cucumber, taxi, and zebra – which are the same classes used in the TCAV paper. We then create multiple versions of this dataset by altering a percentage of the images with a tag, represented by a letter enclosed in a randomly sized square and added in a random location of the image (examples are shown in Figure 5a). Specifically, zebra images are tagged with a "Z" in a purple square, taxi images with a "T" in a magenta square, and cucumber images with a "C" in a cyan square. From these tagged images, we create five datasets: one of images without tags, and four others with 25%, 50%, 75%, and 100% of tagged images, respectively. Each dataset is then used to train a different model, each including six convolutional layers and a GAP layer. Depending on the dataset used for training, each model may learn to recognize either the entities (i.e., cucumbers, taxis, and zebras), the tags, or both and will decide which ones to give more importance. To obtain an approximated ground truth assessing which concept – entity or tag – is more important, we ask the models to classify a set of 200 incorrectly tagged test images per class. In this test set, taxis are tagged with the "Z", cucumbers are tagged with the "T" and zebras are tagged with the "C". If the network correctly classifies most of the images, it indicates that the entity is more important than the tag, and thus, its attribution should be higher. On the other hand, if the performance deteriorates on these wrongly tagged images, it indicates that the tag is more important than the entity, and thus its attribution should be higher. We obtain the CAVs for entities using images of each class as concept examples and random images as negative examples. For tags, we use random images containing that tag as concept examples and images of cucumbers, taxis, and zebras containing the other two tags as negative examples. We use the same incorrectly tagged test set to compute the concept attributions for both entities and tags across the last convolutional layer of all models.

The results are shown in Figure 5. As expected, an increase in the percentage of tagged images correlates with a decrease in accuracy. In particular, for the "cucumber" class the accuracy declines much faster compared to other classes, with the majority of the images being incorrectly classified as taxis. This suggests that even the models trained on a small fraction of tagged images tend to overfit on the "T" tag. The concept attributions for both the "cucumber" entity and the "T" tag closely mirror this ground truth. The "zebra" entity and the "C" tag are also consistent with the ground truth: the attributions for "zebra" show a positive correlation with accuracy, whereas the attributions for the "C" tag demonstrate a clear inverse correlation. Notably, the networks did not pay much attention to the "Z" tag, focusing instead on the absence of the other two tags to classify zebras. Indeed, the model trained with 100% of images tagged classifies any image without a "C" or a "T" tag as "zebra", regardless of whether the "Z" tag is present or not. This is confirmed by

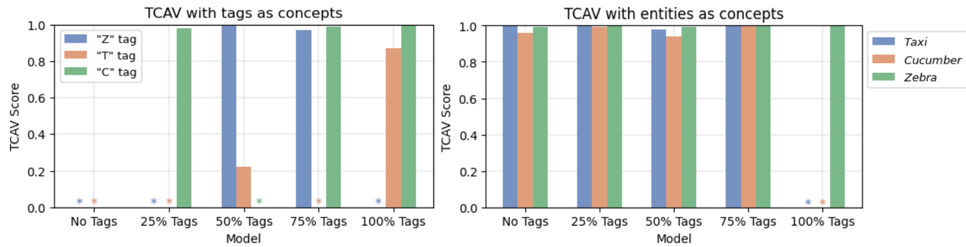


Figure 6: TCAV scores for tags and entities across each validation model. Results marked with an asterisk ("*") have been excluded due to statistical insignificance (p-value > 0.05).

our method, which assigns an attribution of nearly zero to both the "Z" tag and the "taxi" entity for the aforementioned model. We tested other saliency methods, such as Grad-CAM and IG, to further validate these findings. These methods do not highlight the "Z" tag either, but rather the entire image, in search of the "zebra" class (see Appendix B). For models trained with less than 100% of tags, the accuracy for "taxi" remains high, implying that these models are indeed capable of recognizing the "taxi" entity. The concept attribution for the "taxi" entity aligns with this observation. In Figures 5b and 5c, we provide examples of concept maps for the model trained without tags and the model trained with 100% of tagged images. The former recognizes the entities but not the tags, while the latter struggles to recognize the entities but effectively identifies the "T" and "C" tags.

Comparison with the TCAV Score. The primary difference between our concept attribution and the TCAV score is that the former considers not only the direction of gradients but also their magnitude. This allows us to measure the concept's impact on the predictions, beyond just the network's sensitivity to it. To demonstrate this, we compute the TCAV scores for tags and entities across each validation model (see Figure 6). On one hand, TCAV scores match the ground truth in showing that the network trained without tags exhibits high sensitivity to the entities and no sensitivity to the tags. Furthermore, TCAV aligns with the concept attribution in showing that the 100% tags model is sensitive to the "T" and "C" tags but not to the "Z". On the other hand, TCAV struggles to capture the variations in the concept's importance defined by ground truth. In fact, all models except the 100% tags show very similar TCAV scores for the entity concepts, even though their importance varies significantly across these models. This is attributable to most of the networks being sensitive to the entities. Indeed, on images without tags, the models' accuracies are 96.5%, 96.2%, 96.2%, 95.2%, and 36.2% respectively. Similarly, the "C" tag has almost the same TCAV score for the models trained with 25%, 75%, and 100% tags, which is inconsistent with the decline in accuracy for the "C" tagged zebras.

5 Conclusion

In this article, we introduced a novel method, Visual-TCAV, to explain the outputs of image classification models. This framework is capable of providing both local and global explanations based on high-level concepts, by estimating their attribution to the network's predictions. Additionally, Visual-TCAV generates saliency maps to show where concepts are identified by the network, thereby assuring the user that the attributions correspond to the intended concepts. The effectiveness of this method was demonstrated across a range of widely used CNNs and through a validation experiment, where Visual-TCAV successfully identified the most important concept in each examined model.

Limitations and Future Work. Visual-TCAV provides a novel approach for concept-based explainability, but it has some limitations. Our current implementation only considers positive attributions for classes with positive logit values. However, since a concept may negatively impact the output, in future implementations we aim to include negative values, which would improve explanations and also extend the applicability of Visual-TCAV beyond classification tasks. Another limitation arises from the accumulation of noise along the IG linear path, which may sometimes result in slightly underestimated attributions. Future studies could investigate how to mitigate this using alternative IG variants to compute the attributions of feature maps. Additionally, future research could explore generative approaches such as DreamBooth [19] to generate a large number of concept images starting from a small set of examples, leading to more robust CAVs and reducing workload for analysts. Finally, future works could study interconnections between concepts to determine how the activation of a concept might influence not only the output but also the activation of other concepts.

References

[1] Matteo Bianchi, Antonio De Santis, Andrea Tocchetti, and Marco Brambilla. Interpretable network visualizations: A human-in-the-loop approach for post-hoc explainability of cnn-based image classification, 2024. URL https://doi.org/10.48550/arXiv.2405.03301.

[2] Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, page 258–26. Association for Computing Machinery, 2019. ISBN 9781450362726. doi: 10.1145/3301275.3302289. URL https://doi.org/10.1145/3301275.3302289.

[3] Lei Cai, Jingyang Gao, and Di Zhao. A review of the application of deep learning in medical image classification and segmentation. *Annals of Translational Medicine*, 8(11), 2020. ISSN 2305-5847. URL https://atm.amegroups.com/article/view/36944.

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: A generalized framework for concept-based explanations. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/11a7f429d75f9f8c6e9c630aeb6524b5-Abstract-Conference.html.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[7] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/77d2afcb31f6493e350fca61764efb9a-Paper.pdf.

[8] Mara Graziani, Vincent Andrearczyk, and Henning Müller. Regression concept vectors for bidirectional explanations in histopathology. In Danail Stoyanov, Zeike Taylor, Seyed Mostafa Kia, Ipek Oguz, Mauricio Reyes, Anne Martel, Lena Maier-Hein, Andre F. Marquand, Edouard Duchesnay, Tommy Löfstedt, Bennett Landman, M. Jorge Cardoso, Carlos A. Silva, Sergio Pereira, and Raphael Meier, editors, *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pages 124–132, Cham, 2018. Springer International Publishing. ISBN 978-3-030-02628-8.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

[10] Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: an adaptive path method for removing noise. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5048–5056, 2021. doi: 10.1109/CVPR46437.2021.00501.

[11] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2668–2677. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

[12] Zachary Lipton. The mythos of model interpretability. *Communications of the ACM*, 61, 10 2016. doi: 10.1145/3233231.

[13] Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. pages 1–10, 07 2020. doi: 10.1109/IJCNN48605.2020.9206946.

[14] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`.

[15] Tyler Martin and Adrian Weller. *Interpretable Machine Learning*. M.Phil. diss., Dept. of Engineering, University of Cambridge, August 2019. URL `https://www.mlmi.eng.cam.ac.uk/files/tam_final_reduced.pdf`.

[16] Deng Pan, Xin Li, and Dongxiao Zhu. Explaining deep neural network models with adversarial gradient integration. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2876–2883. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/396. URL `https://doi.org/10.24963/ijcai.2021/396`. Main Track.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16. ACM, August 2016. doi: 10.1145/2939672.2939778. URL `http://dx.doi.org/10.1145/2939672.2939778`.

[18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042`.

[19] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.02155. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.02155`.

[20] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017. doi: 10.1109/iccv.2017.74. URL `http://dx.doi.org/10.1109/ICCV.2017.74`.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[22] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

[23] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL `http://arxiv.org/abs/1706.03825`.

[24] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[25] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 3319–3328. JMLR.org, 2017.

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

[28] Tolga Turay and Tanya Vladimirova. Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey. *IEEE Access*, 10:14076–14119, 2022. doi: 10.1109/ACCESS.2022.3147495.

[29] Warren von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34, 12 2021. doi: 10.1007/s13347-021-00477-0.

[30] Chase Walker, Sumit Jha, Kenny Chen, and Rickard Ewetz. Integrated decision gradients: Compute your attributions where the model makes its decision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38:5289–5297, 03 2024. doi: 10.1609/aaai.v38i6.28336.

[31] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A. Ehinger, and Benjamin I. P. Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11682–11690, May 2021. doi: 10.1609/aaai.v35i13.17389. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17389`.

A Appendix Overview

In the appendix, we provide:

B. Saliency methods for 100% tags model

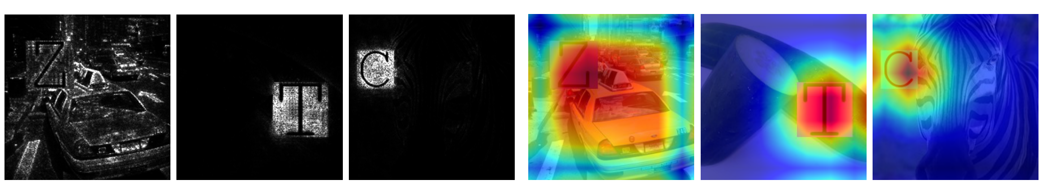
C. Additional results of Local Explanations

D. Additional results of Global Explanations

E. Example images for generated concepts

B Saliency methods for 100% tags model

We provide the results obtained by applying IG and Grad-CAM to the 100% tags model (see Figure 7). These methods align with Visual-TCAV in showing that this model does not pay attention to the "Z", but rather to the absence of the "T" and the "C" for predicting the "zebra" class.



(a) Integrated Gradients (b) Grad-CAM

Figure 7: Integrated Gradients and Grad-CAM for the model with 100% tags, searching respectively for the classes "zebra", "taxi", and "cucumber". Both methods highlight the "T" for class "taxi" and the "C" for class "cucumber", but fail to recognize the "Z" for class "zebra".

C Additional results of Local Explanations

Continuing from the results presented in Section 4.1, we further provide additional local explanations for more input images and concepts in Figure 8.

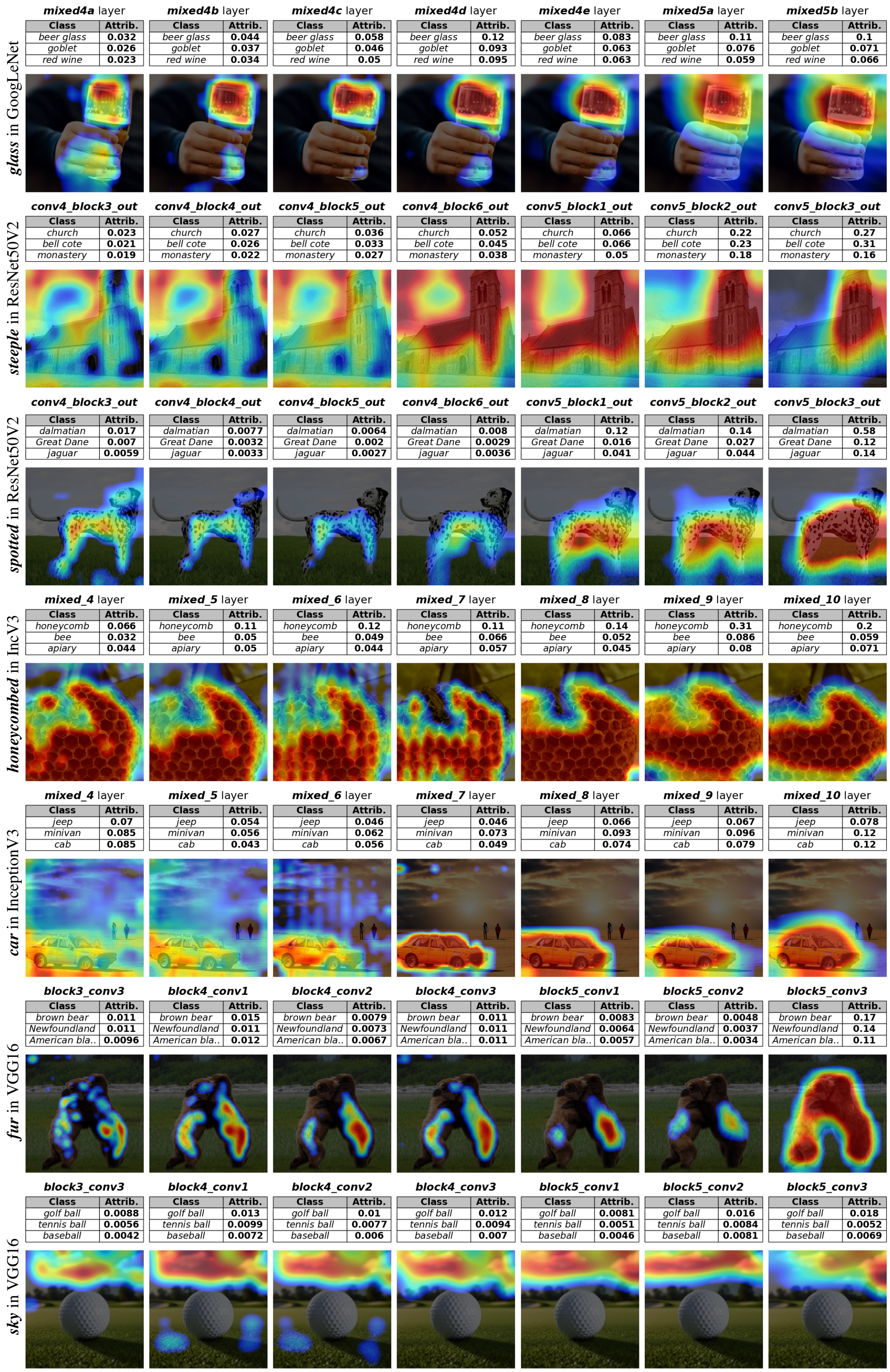


Figure 8: More examples of layer-wise local explanations for various concepts and networks.

D Additional results of Global Explanations

Building upon the results outlined in Section 4.2, we provide additional global explanations for various classes and concepts in Figure 9.

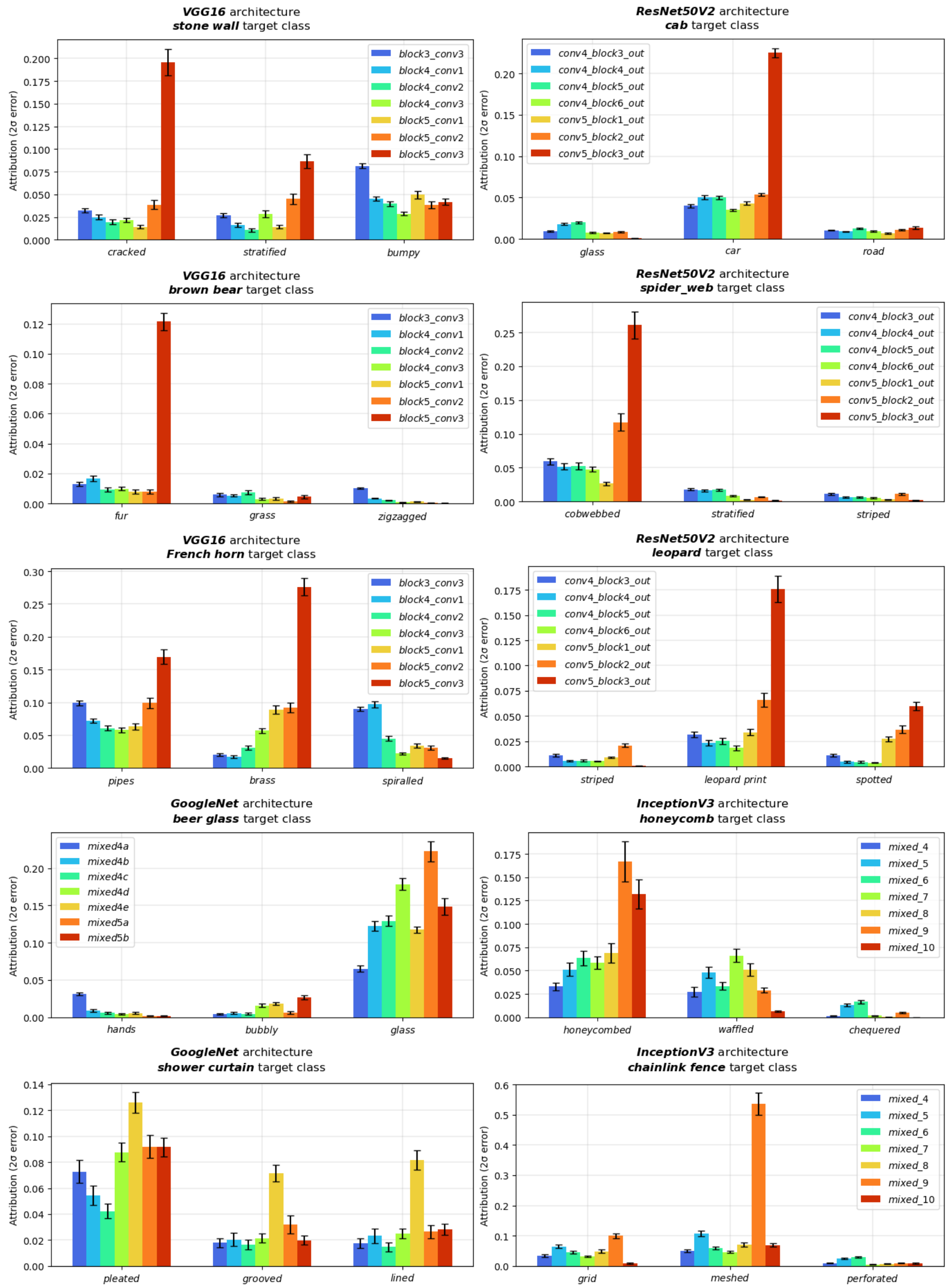
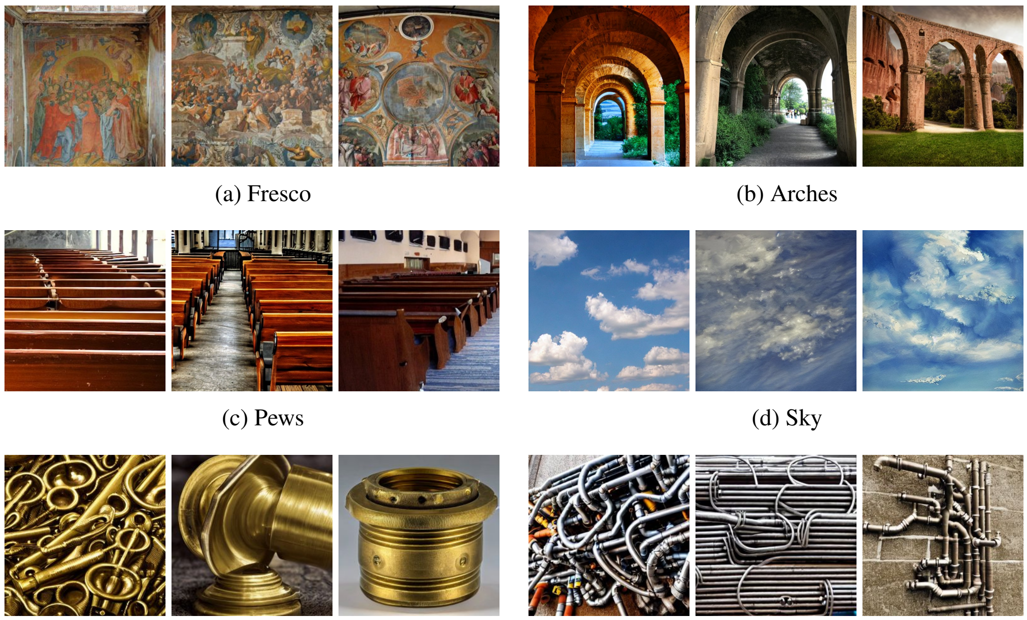


Figure 9: More examples of global explanations for various classes, concepts, and networks.

E Example images for generated concepts

Some of the concepts used in the paper were automatically generated using Stable Diffusion v1.5 [18] with default parameters. In particular, we generated the following concepts: "pews", "fresco", "arches", "sky", "pipes", and "brass". We used just the concept name as a prompt and generated 200 images per concept. A subsequent manual revision was still necessary to eliminate errors and strange artifacts. In Figure 10, we provide three example images for each generated concept.



(a) Fresco (b) Arches

(c) Pews (d) Sky

(e) Brass (f) Pipes

Figure 10: We provide three example images for each concept generated with Stable Diffusion v1.5.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that our method can provide visual explanations through saliency maps based on user-defined concepts, estimate the attributions of these concepts for a selected class, and provide both local and global explanations. These claims are all validated through the experimental results performed in the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our work has some limitations, we discuss them in Section 5.1.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include any new proof or theorem.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Every experimental result presented in the paper is fully reproducible using the provided code and data.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code, data, and instructions needed to reproduce every experiment both to reviewers and to the public through a GitHub repository (in case of publication).

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper describes in detail all the necessary steps to reproduce and understand the experiments. Furthermore, the code used is also available as supplementary material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In our bar plots we always report 2-sigma error bars.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We describe in detail the characteristics of the machine used to run all the experiments and the execution time.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and our research conforms with it.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In the introduction, we briefly discuss the problem of transparency in AI systems, particularly as Convolutional Neural Networks are being widely utilized in critical sectors such as healthcare and autonomous driving. Our work can have a positive societal impact by facilitating a trustworthy adoption of these systems. We are not aware of any negative impact our work could have.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not release any data or models that pose such risks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models and datasets used for the experiments are properly cited in the paper.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new assets.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.