Activation-Based Concept Extraction for Explainability in Image Classification

Matteo Bianchi[1*] matteo.bianchi@polimi.it
Riccardo Campi[1*] riccardo.campi@polimi.it
Antonio De Santis[1*] antonio.desantis@polimi.it
Sara Merengo[1*] sara.merengo@mail.polimi.it
Marco Brambilla[1] marco.brambilla@polimi.it
[1]Politecnico di Milano, DEIB, Italy

Abstract

Concept-based explainability aims to interpret deep learning models such as CNNs in terms of high-level, human-understandable concepts. However, the need to manually define concepts often limits these approaches, as it requires domain knowledge and manual effort to collect example images for each concept. Moreover, many automatic extractors rely on random cropping or activation-agnostic segmentation, which can produce many noisy or irrelevant concepts. To address this, we introduce Activation-Based Concepts (ABC), a post-hoc technique for automatically extracting visual concepts aligned with the model's internal activations, whose importance for a prediction can be computed using standard concept attribution methods. To validate the proposed approach, we perform a user study to measure concept understandability and coherence, and a concept-removal experiment to assess fidelity. We compare ABC with other state-of-the-art methods for post-hoc concept extraction, showing better understandability and concept coherence with comparable fidelity.

1. Introduction

In recent years, we have witnessed unprecedented progress in computer vision, automating a variety of tasks such as image classification and captioning, semantic segmentation, object detection, and many others. However, due to the size and complexity of models, AI decisions cannot be easily explained. We refer to such models as black boxes, as only their inputs and outputs are observable, while their internal functioning is too complex to be interpreted by the human mind. Without transparency, we face significant challenges in AI safety and trust, especially in tasks where predictions can have a significant impact on material goods, economic interests, or people's lives, such as industrial applications [17], medical diagnosis [8], and autonomous driving [15].

Additionally, the lack of interpretability can lead to undetected biases [6, 7], exacerbating existing inequalities.

In the context of computer vision, a popular approach to eXplainable AI (XAI) is using methods such as Grad-CAM [26] or Integrated Gradients [29] to produce *saliency maps* showing which regions of the image contribute most to a certain prediction [9]. While this allows understanding *where* the network is looking, it does not answer the question of *what* the network is seeing there. On the other hand, concept-based explainability methods focus on the idea of providing explanations based on high-level, human-understandable concepts. This approach has been introduced with TCAV [18] and later extended by Visual-TCAV [10], allowing users to define any concept through example images and evaluate its impact on the model's predictions. Methods were also developed to automatically discover learned concepts [11, 14, 33] by performing pixel segmentation or random cropping and clustering the resulting patches. However, a limitation of these methods is that they ignore where the model is paying attention when deciding where to crop, which can lead to extracting meaningless concepts or breaking them apart. They also require users to define a fixed number of concepts per class, which can over- or underestimate the actual number of concepts learned by the model.

In this study, we address these limitations by proposing Activation-Based Concepts (ABC). In contrast with previous methods, we integrate the model's internal state in the image segmentation step to extract only patches that the network is paying attention to, reducing the risk of computing irrelevant concepts, as areas that are not used in the predictions are ignored. The need for human supervision is also reduced by automatically determining the number of concepts from a range of possible values. A refinement step is also added to remove noisy concepts and merge too similar ones. The extracted concepts are represented by a set of

*These authors contributed equally.
Correspondence to: matteo.bianchi@polimi.it.

cropped image patches and are used to generate local and global explanations through Visual-TCAV.

2. Related Work

The field of XAI focuses on developing methods that can provide human-understandable explanations of the inner workings and predictions of AI models. In this paper, we focus on *post-hoc* explainability (i.e., explaining models after training) of Convolutional Neural Networks (CNNs) trained for image classification. One of the earliest approaches to explain computer vision models is to provide explanations via *saliency maps*, using gradients to highlight which areas of the input image contributed most to a prediction. Widely used methods in this category are CAM [34], Grad-CAM [26], and Integrated Gradients (IG) [29]. Since these methods operate on pixels, it can be difficult to understand which high-level features or patterns the network recognizes in an image. Additionally, studies have shown that these methods are fragile and vulnerable to confirmation bias [2, 13].

Beyond saliency methods, recent studies have proposed focusing on *concept-based* methods, which aim at providing explanations in terms of higher-level, human-understandable attributes or abstractions, referred to as *concepts*, to better align with the way humans reason and explain. A seminal work in this area is Testing with Concept Activation Vectors (TCAV) [18], which estimates a global concept importance score for user-defined concepts. The method asks the user to provide a set of images representing a concept c along with a set of negative examples, which are usually random images. Then, a Concept Activation Vector (CAV) is computed for c as the normal to a hyperplane separating feature map activations at layer l of example images from those of negative examples. The learned CAV can be used to derive the "conceptual sensitivity" of the model to a certain concept c, by computing the directional derivatives of the outputs with respect to changes in the inputs towards the direction of the concept (represented by the CAV). Later, De Santis et al. [10] proposed Visual-TCAV, an extension of TCAV that also provides local explanations composed of a *concept map*, which is a saliency map that localizes a concept in an image, and a *concept attribution*, which estimates the concept's contribution for a prediction, not just the model's sensitivity to it. The concept map is particularly useful to assess whether high importance scores can truly be attributed to the concept intended by the user. Unlike TCAV, Visual-TCAV applies a Global Average Pooling (GAP) operation to the CAV so that the position of the concept in the example images does not affect its representation. The concept map for a concept c is then obtained by computing a weighted sum of the input image's feature maps at layer l, using the CAV for c as the weights vector, while the concept attribution is computed by calculating the feature maps attributions using IG and combining them through a weighted sum, also using the CAV as the weights vector.

While these methods are effective in testing specific hypotheses about how the model recognizes a class, the manual selection of concepts can be critical in terms of scalability, required expertise, and confirmation bias [14, 31]. Modern big datasets can have hundreds of classes, and finding multiple meaningful concepts for each class, compiling a set of example images, and testing concept importance can be an effort-intensive task. Additionally, there is a growing trend of systems performing tasks requiring significantly more expertise than that of their users, who might not be able to define relevant concepts or find accurate example images. To address this issue, Automated Concept-based Explanation (ACE) [14] is among the earliest works trying to automatically extract concepts. Starting from a set of images of class k, they segment each image with SLIC pixel segmentation [1], using multiple resolutions to capture a diverse range of concepts at different sizes (e.g., textures, parts of objects, complete objects). Then, cropped patches are passed through the CNN, and the resulting activations are used as representations to cluster them into concepts using K-means. However, clustering methods can be inefficient for concept extraction, as they might find multiple concepts with very similar directions. For this reason, Invertible Concept-based Explanation (ICE) [33] improves on ACE by instead applying Non-Negative Matrix Factorization (NMF) to the patches representations and identifying a concept for each reduced direction. Each concept is then represented by the patches that maximally activate in this direction. While NMF has been shown to perform better than K-means clustering for concept extraction [12], a significant limitation of ICE is that, like ACE, it segments images using SLIC, which requires padding each segment with a default value, potentially introducing bias and artifacts [19, 28]. To address this limitation, Concept Recursive Activation FacTorization for Explainability (CRAFT) [11] identifies concepts as randomly cropped square patches, so that no padding is necessary. Similarly to ICE, they use NMF to identify concept directions. Overall, both SLIC segmentation methods (such as ACE and ICE) and random cropping techniques (such as CRAFT) have limitations, as neither approach considers what the network is truly focusing on, which can result in fragmenting concepts the network recognizes as a whole or focusing on irrelevant image regions.

3. Methodology

We propose Activation-Based Concepts (ABC), a post-hoc automatic concept extraction method that aims to overcome the limitations of state-of-the-art approaches by aligning cropped patches with the network's internal processing of the image. To achieve this, we guide segmentation using clusters of similar feature maps, extracting only rele-

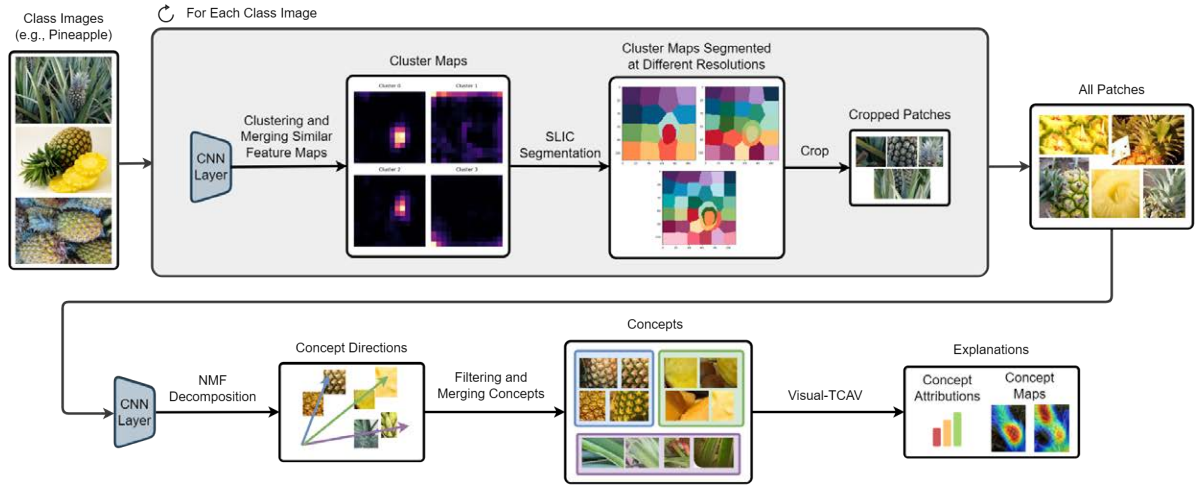


Figure 1. Overview of ABC methodology. Feature maps from images of a class are extracted, clustered, and segmented into patches. Then, patch activations are decomposed via NMF to extract meaningful concepts. Finally, explanations are provided by localizing these concepts and computing their importance for the decision.

vant segments and disregarding areas that are not used in the prediction. We then apply NMF for concept extraction, followed by an additional refinement step to remove noisy concepts and merge similar ones. Concepts extracted with ABC are represented as a set of image patches, which can be used to generate local and global explanations leveraging methods that compute concept importance through example images, such as TCAV and Visual-TCAV. This way, we show not only which image regions the network is paying attention to, but also which concepts the model recognizes in those regions, represented by the example images, and how much they contribute to each class.

ABC requires two steps. The first is *Image Segmentation*, during which we compute feature maps for a set of test images and use them in combination with segmentation to extract relevant patches from these images. The second is *Concept Extraction*, in which concept directions are computed by applying NMF to the set of extracted patches. A high-level overview of ABC is shown in Figure 1.

3.1. Image Segmentation

We start by selecting a set D_k of images of class k and compute their feature maps at layer l, discarding the ones for which activations are zero. Since many feature maps often look at the same image region, extracting a cropped patch for each feature map may lead to many redundant patches. To address this, we cluster the feature maps for each image using the approach proposed by Bianchi et al. [5]. This approach involves clustering similar feature maps using agglomerative clustering, with the optimal number of clusters chosen based on the average silhouette score. Then it computes a set of cluster maps by averaging all feature maps belonging to a cluster. We use these condensed feature maps, which are 3-8 per image, for segmentation and cropping so that we crop only 3-8 patches for each image based on what the model was paying attention to.

To perform segmentation, we try to improve on intuitions from previous methods. Similar to ACE and ICE, we perform segmentation with three different levels of resolution to capture a diverse range of concepts at varying levels of granularity (e.g., objects, parts of objects, and textures). This is in contrast with CRAFT, which instead uses a fixed, square cropping size. However, while ACE and ICE apply SLIC directly on the image itself, we instead use it to segment the cluster map, ensuring that the segmentation aligns with the model's internal processing of the image. Additionally, rather than selecting all segments, we choose a single segment for each cluster map, aiming to extract the one that best represents the concept activating the map. To balance capturing as much as possible of the high-activation areas while avoiding including inactive regions, we use the 95-percentile of the cluster map to split high- and low-activation areas into a binary mask. We then apply this mask to select the segment capturing the largest active area and the highest ratio of active area to total area. Finally, we use the bounding box of the selected segment to crop a patch from the original image. Similarly to CRAFT, we use rectangular patches instead of irregular-shaped segments to avoid having to pad them with a default value before passing them to the CNN, which could introduce unwanted bias and artifacts in the concept extraction phase. Our approach, however, differs in the fact that, unlike CRAFT, which uses square patches, we do not place any constraints on the aspect ratio of extracted patches to capture concepts with a diverse range of shapes.

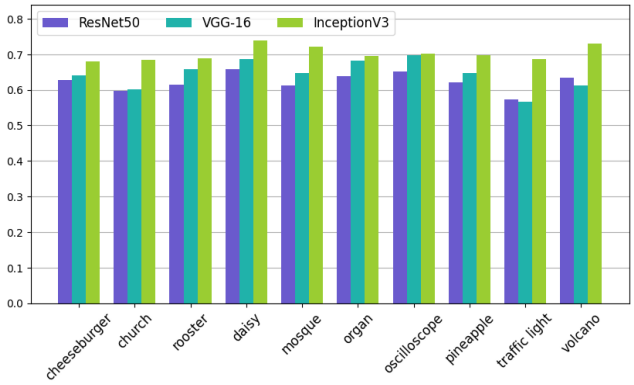
3.2. Concept Extraction

This step consists of extracting relevant concepts from the set of patches at the previous step. First, we compute the patches' representations in the CNN's latent space, which allows us to evaluate their visual similarity according to what the model has learned [32]. Unlike previous methods, however, we use the fully convolutional [21] version of the CNN, as it can process inputs of any size, thus avoiding the introduction of artifacts from resizing patches to a square size (e.g., vertical images ending up in a different cluster than horizontal images of the same concept). The computed feature maps are then averaged with a GAP operation, resulting in each patch p being represented by a vector $a_p \in \mathbb{R}^{\mathrm{d}}$, with d being the number of channels at layer l. This is done because the spatial location of the concept within the patch is irrelevant at this stage.

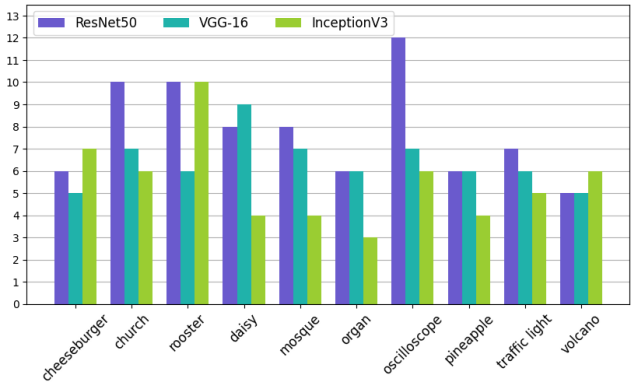
To extract concept directions, we can now think of the set of vector representations as a matrix $A \in \mathbb{R}^{\mathrm{n}\times\mathrm{d}}$, where n is the number of patches, and apply NMF to obtain a new representation. NMF decomposes non-negative matrix A into a product of two non-negative matrices $W \in \mathbb{R}^{\mathrm{n}\times\mathrm{r}}$ and $H \in \mathbb{R}^{\mathrm{d}\times\mathrm{r}}$ by solving:

$$(W, H) = \arg\min_{W\geq 0, H\geq 0} \frac{1}{2}\|A - WH^T\|_F^2 \qquad (1)$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm. We can interpret matrix H as containing the concept directions in the activation space, and matrix W as redefining each data point a_p according to the new basis, transforming it into a new vector $w_p \in \mathbb{R}^{\mathrm{r}}$. The number r is a hyperparameter determining the dimensionality of the new representations, i.e., the number of concepts to extract. For each concept c corresponding to the i-th vector in H, we can interpret the corresponding i-th value of w_p as representing how much patch p activates concept c. Thus, we can use this value to represent each concept c as a set P_c of the N patches having the highest activations for c in the new representation W. NMF has been widely used for concept extraction and has been shown to perform better than K-means [12]. It works under the assumption that the representations matrix A is non-negative, which is reasonable as recent CNN models use the ReLU function as their activation function [33]. Choosing the right number r of concept directions to extract is also important. Extracting just a few directions might lead to missing important concepts, while a high r could result in concepts that are overall noisier and less coherent. This is because computing new directions with NMF does not leave previous directions unaltered, meaning that changing r can result in completely different concept directions. ICE and CRAFT solve this issue by having the user manually choose the number of concepts for each class, relying on visual inspection to assess the quality of concepts. This, however, may not scale effectively for large datasets. Since the optimal number of concepts to extract cannot be known beforehand, we iterate over different r values, choosing the one that maximizes the internal coherence of concepts. Specifically, for each concept c, we estimate its internal coherence by computing the average cosine similarity of the patches in P_c. We then choose the set of concepts yielding the maximum average internal coherence. After the extraction, some concepts might still be noisy (i.e., with low internal coherence). Therefore, if the internal similarity is lower than 0.5, we discard the concept.



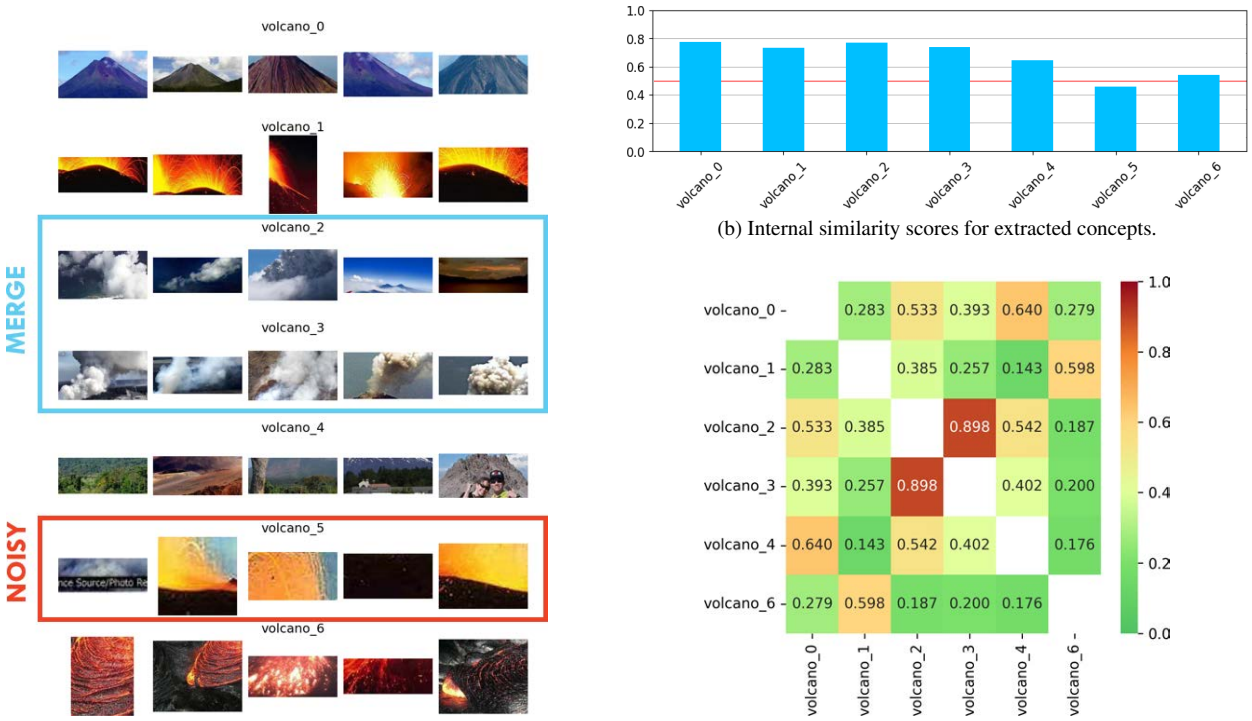
(a) Average internal similarity of concepts obtained for each analyzed class and model architecture.



(b) Number of concepts obtained for each combination of tested class and examined architecture.

Figure 2. Comparison across classes and models of (a) avg. internal similarity and (b) no. of concepts.

While ICE and CRAFT use the concept directions directly from H, our approach is more similar to TCAV and Visual-TCAV, which proposed a comparison with a set of negative examples E_{neg} in the form of random images. Specifically, for compatibility with Visual-TCAV, we obtain the CAV of a concept c using the *Difference of Means* method proposed by Martin and Weller [22]. We compute feature map activations for both the set P_c of patches belonging to c and a set of negative examples E_{neg}, apply a GAP operation, and then obtain the CAV as the differ-



(a) Concepts from "volcano" with VGG-16.

(b) Internal similarity scores for extracted concepts.

(c) CAV similarity matrix for the extracted concepts.

Figure 3. Concepts extracted for class "volcano" with VGG-16 before the processes of noise removal and merging of similar concepts, with the modifications proposed by the algorithm highlighted.

ence between the centroids of the two groups. We use this method so that the importance of extracted and user-defined concepts could be directly comparable (i.e., the same concept would yield the same explanation, whether extracted automatically or collected manually). To reduce redundancies in explanations, we also include the option to merge concepts with high cosine similarity (i.e., > 0.85) between their respective CAVs.

4. Experiments and Results

In this section, we describe the experimental setup and present the results. We provide both a qualitative evaluation with exemplary explanations and a quantitative evaluation. The latter includes a user study evaluating the understandability and coherence of the extracted concepts, and a concept-removal experiment to assess the method's fidelity. Our code can be found at https://github.com/DataSciencePolimi/Activation-Based-Concepts.

4.1. Experiment Setup

As models to explain, we select three widely used architectures: VGG-16, ResNet50, and InceptionV3 [16, 27, 30]. All are pre-trained on ImageNet-1k [25]. To extract concepts, we focus on the last convolutional layer of these models, as it has greater capacity for complex abstractions compared to earlier layers [4, 5, 10, 18] and provides more accurate estimated importance due to its proximity to the output [3, 24, 33]. Next, we select the probing dataset D containing the images from which to extract concepts. Ideally, the probing dataset should contain enough examples of all relevant class concepts. However, since the frequency of relevant concepts is unknown, we build the dataset by randomly sampling from ImageNet, assuming that a sufficiently large subset will capture the most relevant concepts. To this end, we select 500 images per tested class, assuming that very infrequent concepts within such a set are unlikely to be particularly relevant.

While a relatively small number of example images (i.e., 5 to 10) is sufficient for a human user to understand a concept, other tasks, such as computing CAV directions and internal cluster similarities, require more patches per concept. For instance, prior literature commonly selects 30 example images per concept [10, 18]. However, they are incentivized to keep this number as low as possible since the images must be manually selected by the user. On the other hand, setting N too high may lead to capturing patches that are not closely related to the concept. Therefore, for our experiments, we select $N = 50$ patches for each concept. In line with common choices in the literature, the negative set E_{neg} consists of 500 samples from random ImageNet classes [10, 22]. Furthermore, for each class, we select be-

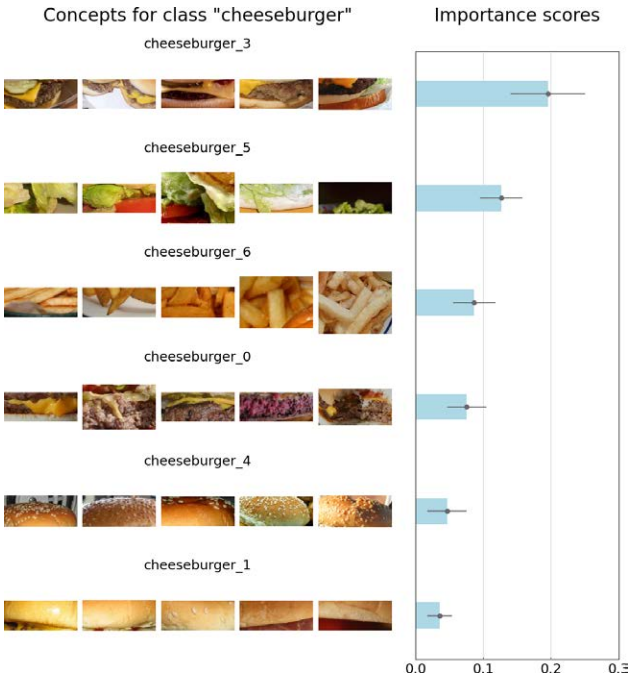
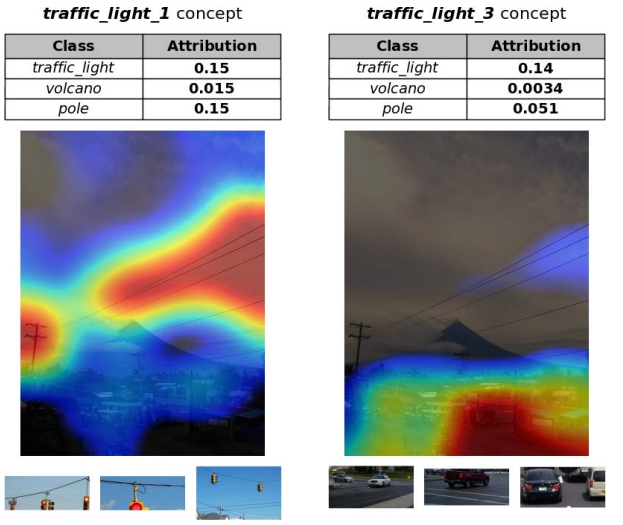


Figure 4. Concepts extracted from the class "cheeseburger". The global explanation shows the Visual-TCAV importance score towards the class with ResNet50 and iterating over a set of 100 images of class "cheeseburger" taken from ImageNet.



(a) Most important concept in local explanation.

(b) Second most important concept in explanation.

Figure 5. Local explanation of an image of class "volcano" wrongly predicted as "traffic light", generated with Visual-TCAV using concepts extracted for class "traffic light" with VGG-16. We show the two concepts with the highest importance towards "traffic light" for this specific input image. The three classes are ranked by their output probability.

tween 7 and 20 concepts, filtering out those with internal similarity lower than 0.5, and merging those with pairwise CAV similarity greater than 0.85. Figure 2a shows the average internal similarity for concepts belonging to different classes and computed with different architectures. After these operations, we obtained 4 to 12 concepts per class, as shown in Figure 2b, which shows the concept distribution for each class and architecture.

We run experiments on an Intel Xeon Gold 5118, 48 vCores, 376 GB RAM, and an NVIDIA P100 with 16 GB VRAM. Each class takes approximately 20 to 30 minutes to be processed, depending on the model.

4.2. Qualitative Evaluation

This section presents and discusses some examples of concepts extracted using our method, illustrating how automatically extracted concepts can help to understand the examined CNN's functioning through local and global explanations. We generate these explanations using Visual-TCAV. For each concept, we use the top N patches returned by our pipeline as example images. From these examples, Visual-TCAV uses the *Difference of Means* method to learn a CAV. The concept map for an input is then obtained by a weighted sum of the model's feature maps, using the CAV elements as weights. The concept attribution is computed by taking the integrated gradients of the target class logit with respect to the same feature maps and weighting them by the CAV. [10] describe this process in more detail. In Figure 3b, we show the concepts extracted for class "volcano" in VGG-16, highlighting the changes proposed by the noise removal and concept merging algorithms. We can see that concept "volcano_5" has been marked as noisy: indeed, it appears less coherent than the other concepts, and it is difficult to understand what it represents. Additionally, concepts "volcano_2" and "volcano_3" have been merged, as they both contain pictures of clouds and smoke. Figure 3c shows that they have the highest CAV similarity among all pairs of concepts, sitting above the 0.85 threshold.

An example of a global explanation computed using our extracted concepts is provided in Figure 4. It shows an explanation for class "cheeseburger" with ResNet50, obtained by averaging importance scores for each previously extracted concept over a set of 100 images of class "cheeseburger", randomly selected from ImageNet. We can observe, for example, that the concept highlighting the cheese is the most important for the predictions. Notably, concepts such as salad or fries are also significantly important, even though they are not part of the semantics of the class.

Extracted concepts can be used to generate local explanations showing how each concept contributed to the prediction in a certain instance. In Figure 5, we show a local explanation for an image of class "volcano" incorrectly predicted by VGG-16 as class "traffic light". The most im-

portant concept for the prediction is "traffic_light_1", which shows traffic lights suspended by cables. The concept is identified instead with the electricity cables in the image. The second most important concept is "traffic_light_3", which shows cars on the road and is identified at the bottom of the image. These concepts, particularly "traffic_light_1", also contributed to the "pole" class, which is the third top-predicted class for this image. Such explanations allow us to gain information about the model's rationale behind mispredictions in a human-understandable way. Furthermore, they provide actionable insights, suggesting that the training set may not contain enough images of volcanoes in city settings, while the network learned to heavily associate roads and electricity cables with traffic lights. More explanations are available in the Appendix.

4.3. Comparative Study with Human Subjects

We perform a user study to compare the understandability and coherence of our extracted concepts with those obtained using CRAFT. The experiment is performed by human participants using a gamified approach and consists of a concept-matching game. This includes a series of 40 questions, with 20 containing concepts extracted with our method and 20 with CRAFT, in a randomized order to avoid bias. For each question, the user is shown an example image and is asked to choose which among three groups of five images best represents the same concept as the example image. An example question is available in Fig. 6, while additional ones are provided in the Appendix. To evaluate answers, we assign 1 point if the user chooses only the correct answer, 0.5 points if the user chooses two answers of which one is the correct one (in this case, we interpret the correct answer as having a confidence of 50%), and 0 points in all other cases. The questionnaire was answered by 217 participants, one of whom was excluded due to a score worse than random guessing. Participants scored an average accuracy of 94.91% on ABC and 69.33% on CRAFT, showing that ABC achieves statistically better results (Welch t-test, p-value $<= 0.001$). Regarding participants' demographics, they were mostly university students of a variety of nationalities, their median age was 24, and the gender split was 76.6% male/23.4% female. We also asked them whether they were experts in Machine Learning, but no statistically significant difference emerged from this distinction, as shown in Tab. 1.

Reasons for CRAFT's worse performance could be related to the fact that CRAFT uses a random cropping strategy to generate image patches, leading to a higher probability of finding noisy or confusing images within the concepts. Furthermore, another explanation for the results might be a concept redundancy issue. For many of the tested classes, computing the default number of concepts (i.e., 10) results in some overlapping concepts. When a user is faced with the choice between two similar concepts, they might choose both or focus on minor details that lead them to choose the wrong answer. On the other hand, our method typically extracts fewer concepts per class (see Fig. 2b). Because of the steps taken to ensure concept coherency and to avoid redundancy, concepts found with our method tend to be more distinct and internally coherent, resulting in better accuracy overall.

Table 1. Comparison of scores from ABC and CRAFT in the concept matching game.

	ABC (Ours)	CRAFT
ML expert (n=142)	**95.32** ±8.29%	69.35 ±26.11%
Not ML expert (n=74)	**94.12** ±8.83%	69.29 ±32.32%
Overall (n=216)	**94.91** ±8.65%	69.33 ±28.39%

4.4. Fidelity Analysis

We run a fidelity experiment to evaluate whether the explanations produced by our method are a valid proxy of the models we are aiming to explain. We use the *c-deletion* metric proposed by Fel et al. [12], which evaluates the importance of the extracted concepts towards a given class while progressively removing (i.e., zeroing) them in the input space according to the assigned importance and concept map, and studying the change in the class logit (i.e., the pre-softmax score) at inference time. We perform the same experiment with CRAFT for comparison. Both methods provide a concept saliency map from which we derive the mask used to perturb the inputs. For deciding which parts of the input to mask, we consider the activation values of the concept maps that exceed the 95th percentile. We run two different experiments on 5000 images randomly selected from ImageNet and belonging to the 10 classes, and progressively remove the extracted concepts in decreasing order of importance. In the first experiment, we remove concepts cumulatively within the input images, while in the second, we remove them independently. The results are shown in Fig. 7. Both the curves obtained with ABC and CRAFT decrease in the standard c-deletion experiment, demonstrating that the extracted concepts are critical for predicting the class from which they were derived. The drop also decreases with diminishing returns, aligning with the order of importance proposed by the two methods. We can see a similar behavior in the second experiment, where the concept removal is applied non-cumulatively. The concept that the two methods deem most important causes the largest drop, and the remaining ones also cause a drop that is proportional to their estimated importance. The only difference with CRAFT seems to be in the fact that in ABC the most important concept on average produces a larger drop than the one proposed by CRAFT. This may be because ABC's top concepts tend to cover the concept more distinctly (see

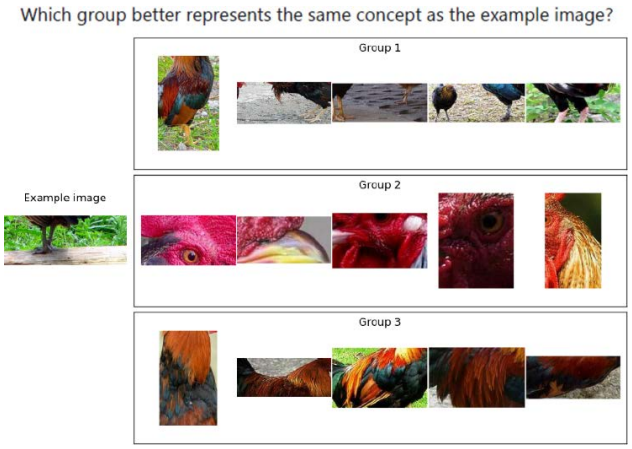


Figure 6. Example question from the concept matching game, showing a choice for class "rooster".

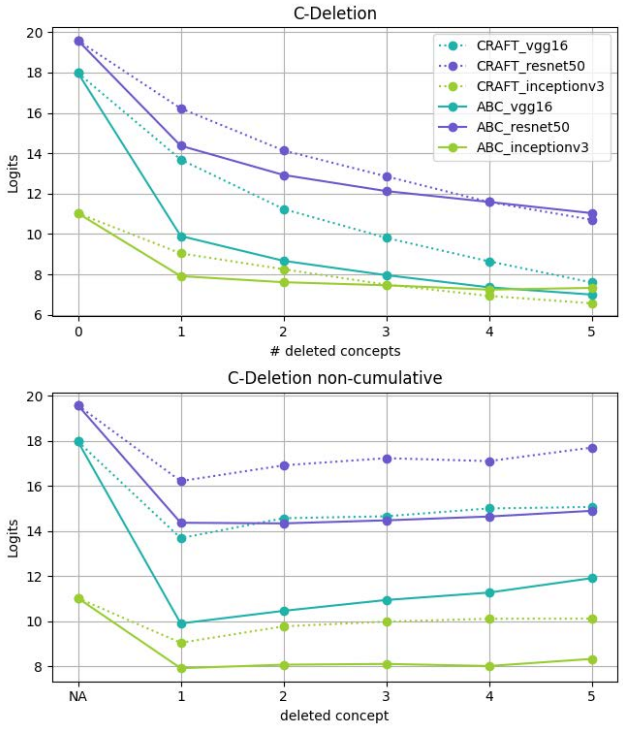


Figure 7. Standard (left) and non-cumulative (right) c-deletion experiments. We observe that removing more important concepts results in a larger decrease in logit scores, with our method producing a more pronounced drop in the first concept than CRAFT.

Fig. 5a), so ablating it disrupts the model's decision boundary to a greater extent than ablating CRAFT's top-ranked concept.

5. Conclusions

We presented ABC, a post hoc method for extracting human-interpretable concepts from CNNs. Our approach improves on the state-of-the-art by extracting patches from images that align with the model's attention and by automatically choosing the number of concepts to extract. It also includes a refinement step to filter out noisy concepts and merge similar ones. We have also shown that extracted concepts can be used to provide both global and local explanations, revealing insights into what the model has learned as well as for identifying the root cause of a misprediction.

5.1. Limitations and Future Work

As limitations, we still lack a systematic way to automatically select thresholds for noise filtering and concept merging, which are currently set heuristically based on qualitative assessment. Another limitation is that currently, ABC is only applicable to model layers whose output is a set of feature maps, such as CNNs, and not Vision Transformers (ViTs). Extending ABC to ViTs is in our future work. Other interesting research directions may be using methods like Text2Concept [23] for automatic concept naming and exploring the use of extracted concepts to train models interpretable-by-design, such as Concept Bottleneck Models [20].

References

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

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 9525–9536, Red Hook, NY, USA, 2018. Curran Associates Inc. 2

[3] Rana Ali Amjad, Kairen Liu, and Bernhard C. Geiger. Understanding neural networks and individual neuron importance via information-ordered cumulative ablation. *IEEE Transactions on Neural Networks and Learning Systems*, 33 (12):7842–7852, 2022. 5

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, 2017. 5

[5] 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. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, page 3715–3723. International Joint Conferences on Artificial Intelligence Organization, 2024. 3, 5

[6] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 4356–4364, Red Hook, NY, USA, 2016. Curran Associates Inc. 1

[7] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR, 2018. 1

[8] Riccardo Campi, Antonio De Santis, Paolo Colombo, Paolo Scarpazza, and Marco Masseroli. Machine learning-based forecast of helmet-cpap therapy failure in acute respiratory distress syndrome patients. *Computer Methods and Programs in Biomedicine*, 260:108574, 2025. 1

[9] Antonio De Santis, Riccardo Campi, Matteo Bianchi, Andrea Tocchetti, and Marco Brambilla. 2 - foundational approaches to post-hoc explainability for image classification. In *Bi-directionality in Human-AI Collaborative Systems*, pages 23–54. Academic Press, 2025. 1

[10] Antonio De Santis, Riccardo Campi, Matteo Bianchi, and Marco Brambilla. Visual-TCAV: Concept-based attribution and saliency maps for post-hoc explainability in image classification. *Transactions on Machine Learning Research*, 2026. https://openreview.net/forum?id=SLh00W5rhu. 1, 2, 5, 6

[11] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Remi Cadenc, and Thomas Serre. CRAFT: Concept Recursive Activation FacTorization for Explainability . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2711–2721, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 1, 2

[12] Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 2, 4, 7

[13] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688, 2019. 2

[14] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. *Towards automatic concept-based explanations*, pages 9277–9286. Curran Associates Inc., Red Hook, NY, USA, 2019. 1, 2

[15] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[17] Rahima Khanam, Muhammad Hussain, Richard Hill, and Paul Allen. A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*, 12:94250–94295, 2024. 1

[18] 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 *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018. 1, 2, 5

[19] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. *The (Un)reliability of Saliency Methods*, pages 267–280. Springer International Publishing, Cham, 2019. 2

[20] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020. 8

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation . In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, Los Alamitos, CA, USA, 2015. IEEE Computer Society. 4

[22] Tyler Martin and Adrian Weller. *Interpretable Machine Learning*. M.Phil. diss., Dept. of Engineering, University of Cambridge, 2019. 4, 5

[23] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023. 8

[24] Ari S. Morcos, David G.T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *International Conference on Learning Representations*, 2018. 5

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 5

[26] 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)*, pages 618–626, 2017. 1, 2

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 5

[28] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 2020. https://distill.pub/2020/attribution-baselines. 2

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 1, 2

[30] 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. 5

[31] Schrasing Tong and Lalana Kagal. Investigating bias in image classification using model explanations. *CoRR*, abs/2012.05463, 2020. 2

[32] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, Los Alamitos, CA, USA, 2018. IEEE Computer Society. 4

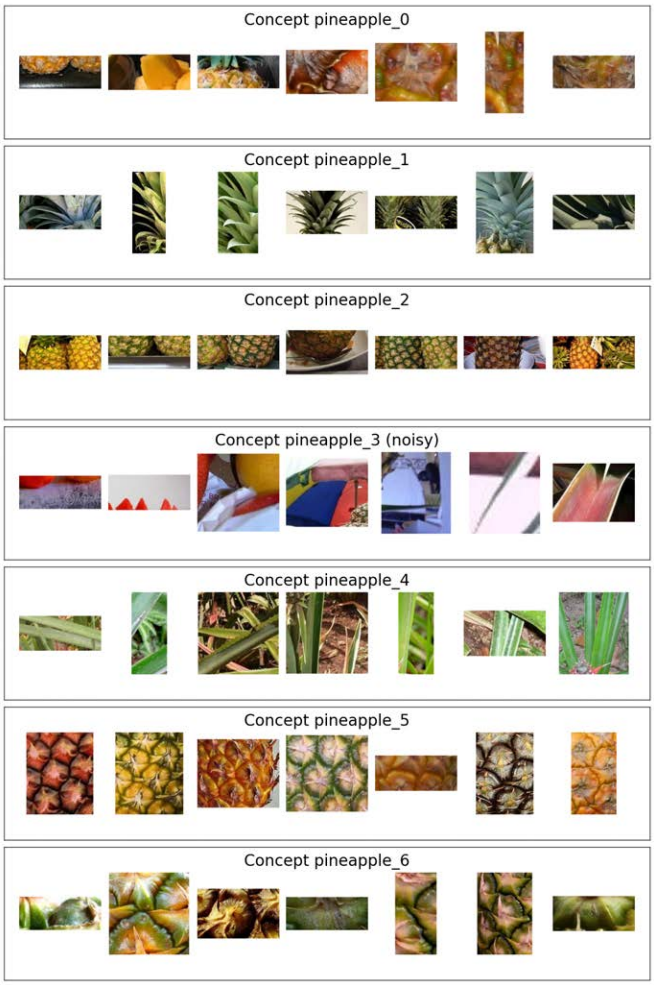
[33] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista Ehinger, and Benjamin Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:11682–11690, 2021. 1, 2, 4, 5

[34] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization . In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, Los Alamitos, CA, USA, 2016. IEEE Computer Society. 2

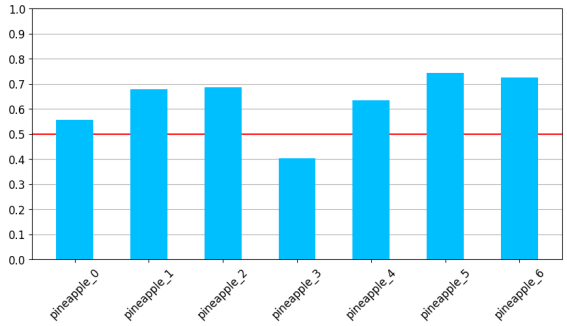
Activation-Based Concept Extraction for Explainability in Image Classification

Supplementary Material

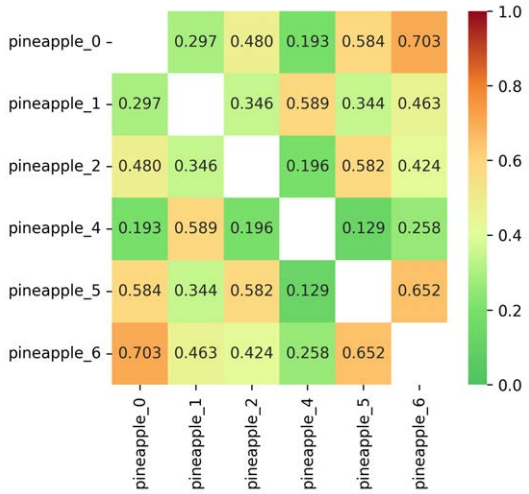
6. Examples of Concept Filtering and Merging



(a) Concepts

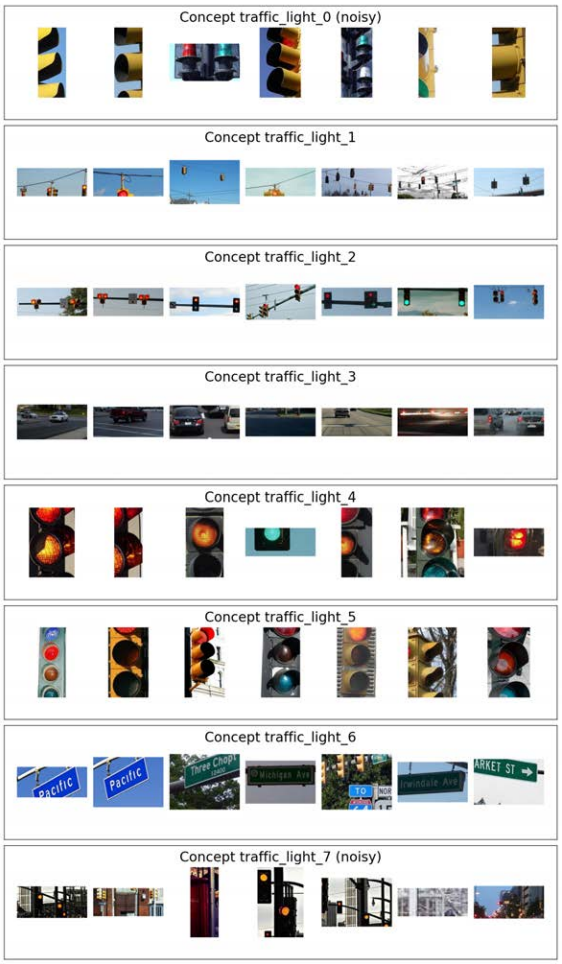


(b) Internal similarity scores for all extracted concepts.

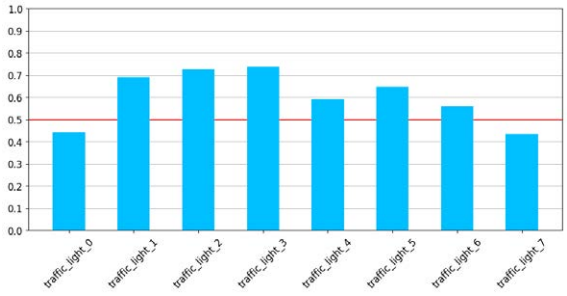


(c) Similarity matrix between CAV directions for non-noise concepts.

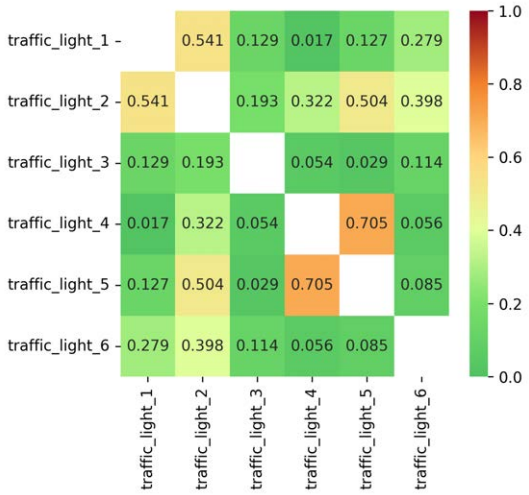
Figure 8. Concepts extracted for class “pineapple” with ResNet50, along with the internal similarity score for each extracted concept and the similarity matrix showing similarity scores between each pair of distinct concepts. Concept “pineapple_3” is the only one falling below the noise threshold, which is coherent with our perception as it is hard to find a common concept between its images. We can also see that the similarity matrix aligns with our perception, for example marking concepts “pineapple_0”, “pineapple_5” and “pineapple_6” as similar. On the other hand concepts “pineapple_1” and “pineapple_4”, which contain images of different types of leaves, are distinct from concepts representing pineapple peel but similar to each other.



(a) Concepts



(b) Internal similarity scores for all extracted concepts.



(c) Similarity matrix between CAV directions for non-noise concepts.

Figure 9. Concepts extracted for class "traffic light" with VGG-16, along with the internal similarity score for each extracted concept and the similarity matrix showing similarity scores between each pair of distinct concepts. We can see that concepts "traffic_light_0" and "traffic_light_6" are not very cohesive, which is reflected in their internal similarity scores as they are the noisiest ones. Additionally, concepts "traffic_light_4" and "traffic_light_5" are marked as the most similar pair, which aligns with our perception as they both represent traffic light bodies, although at different levels of zoom.

7. Examples of Concept Extraction

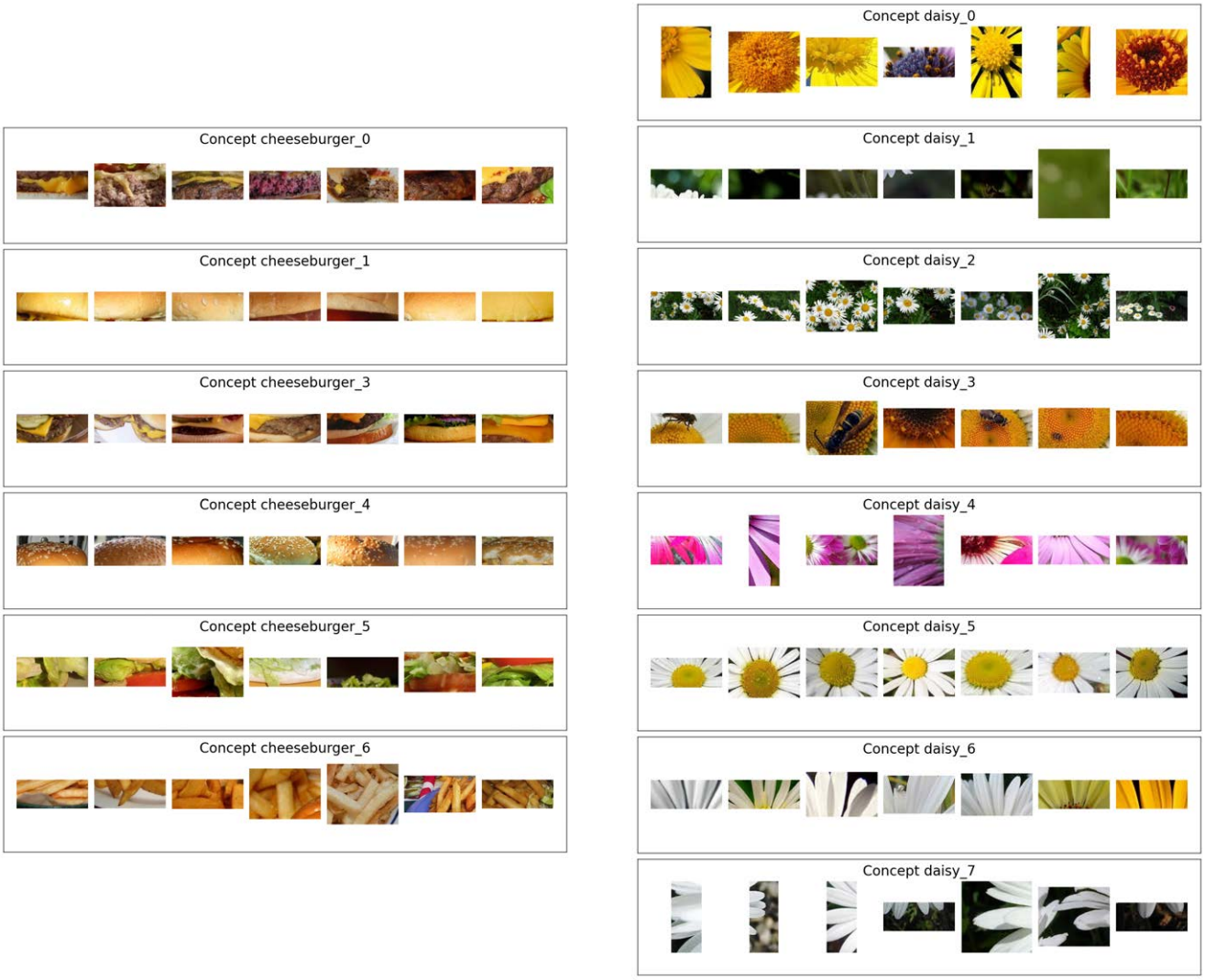


Figure 10. Concepts extracted for class “cheeseburger” and “daisy” with ResNet50.

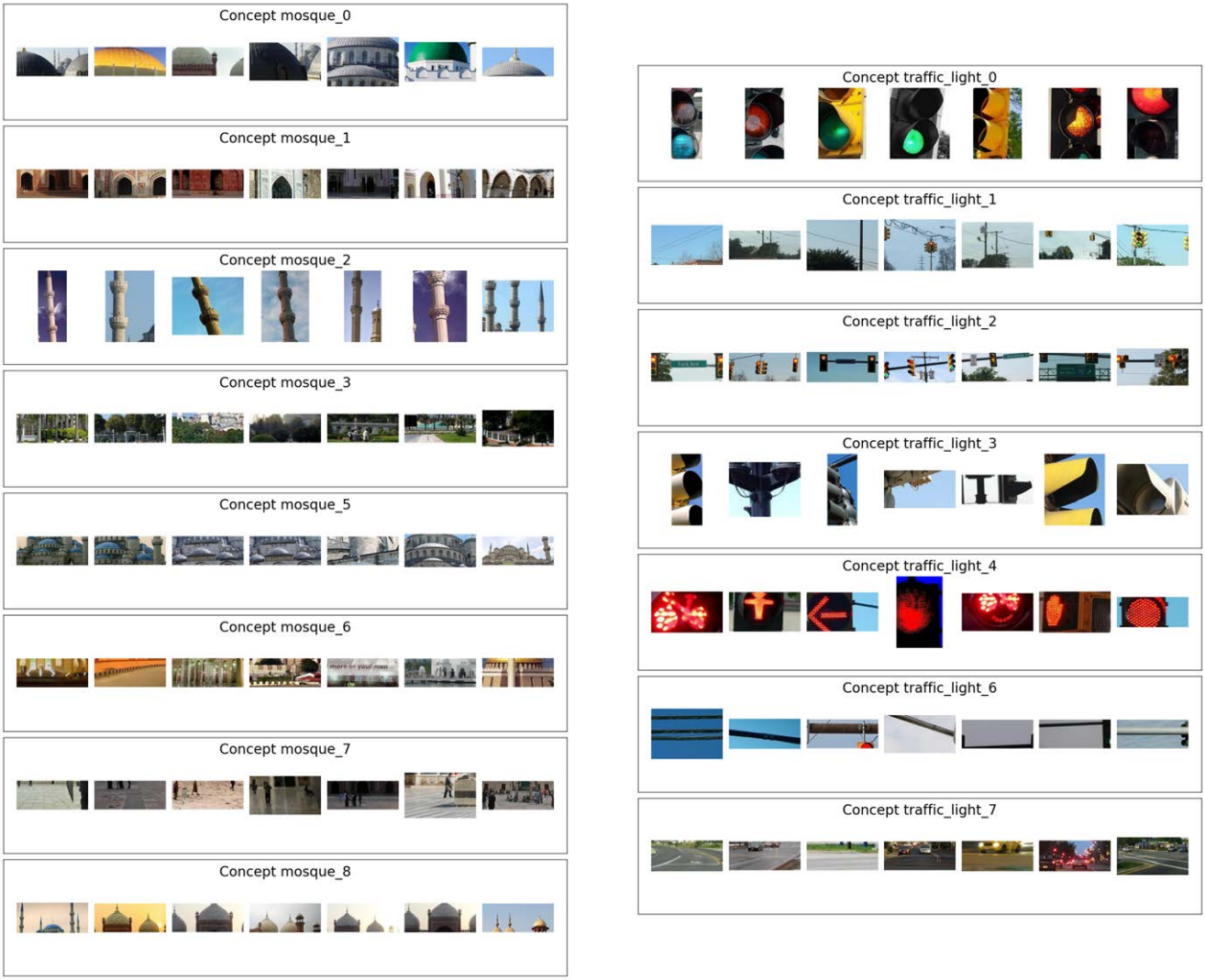


Figure 11. Concepts extracted for class “mosque” and “traffic light” with ResNet50.

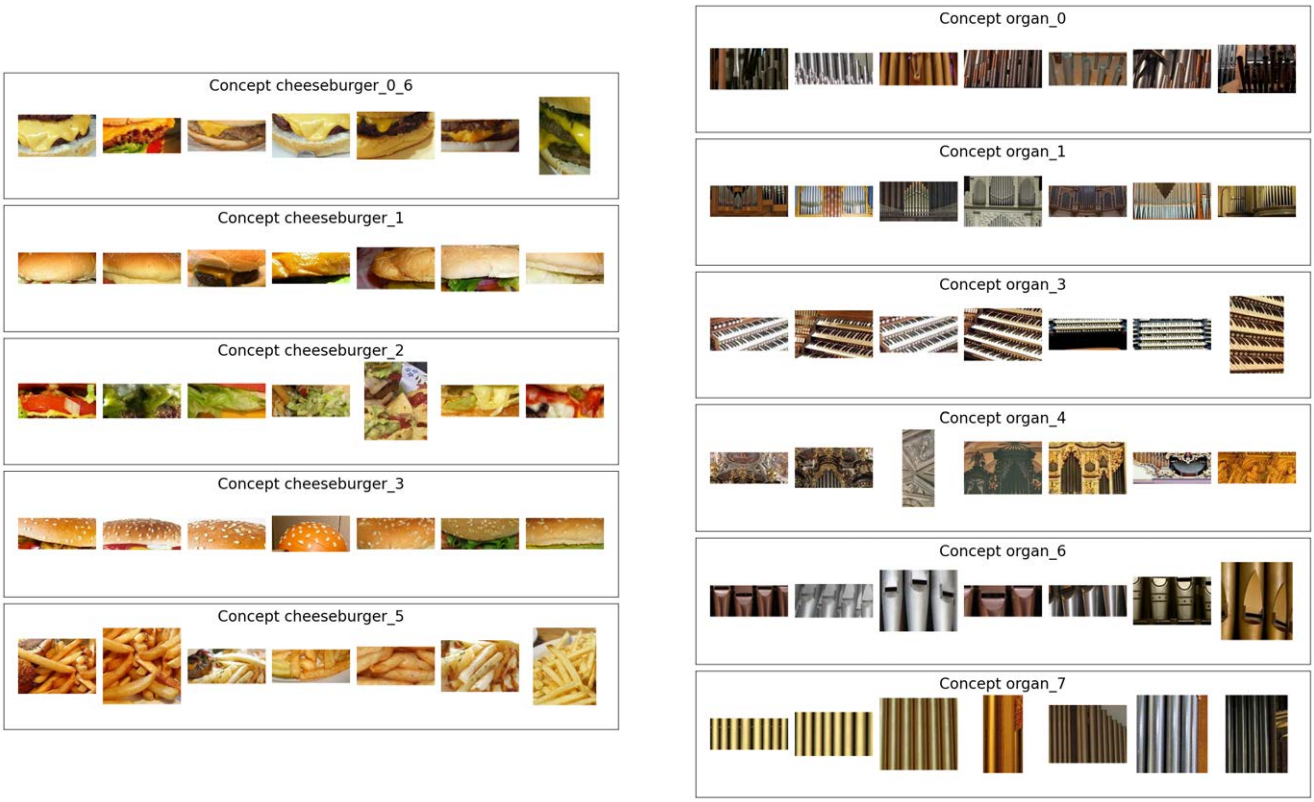


Figure 12. Concepts extracted for class “cheeseburger” and “organ” with VGG-16.

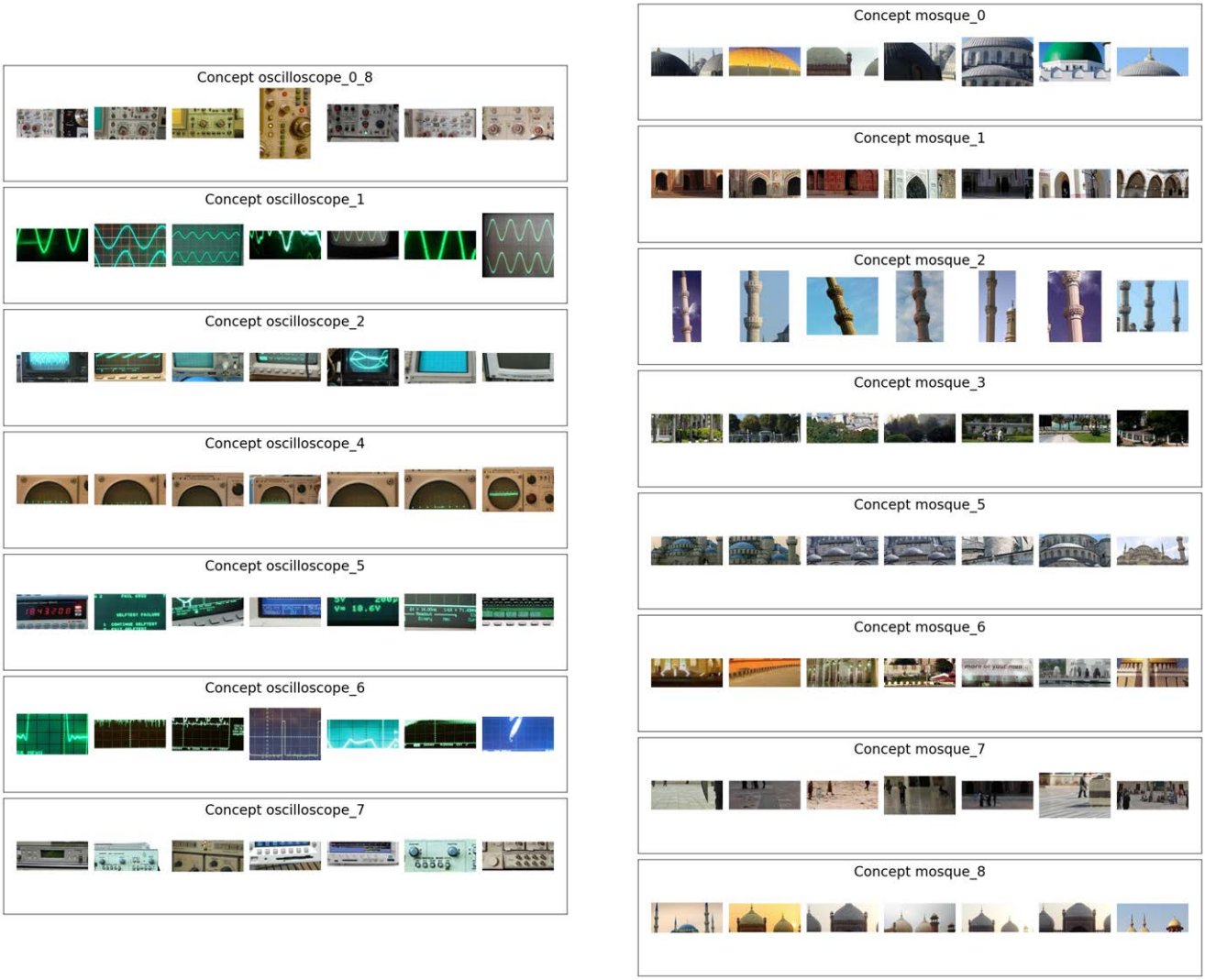


Figure 13. Concepts extracted for class “oscilloscope” and “mosque” with VGG-16.

8. Examples of global explanations

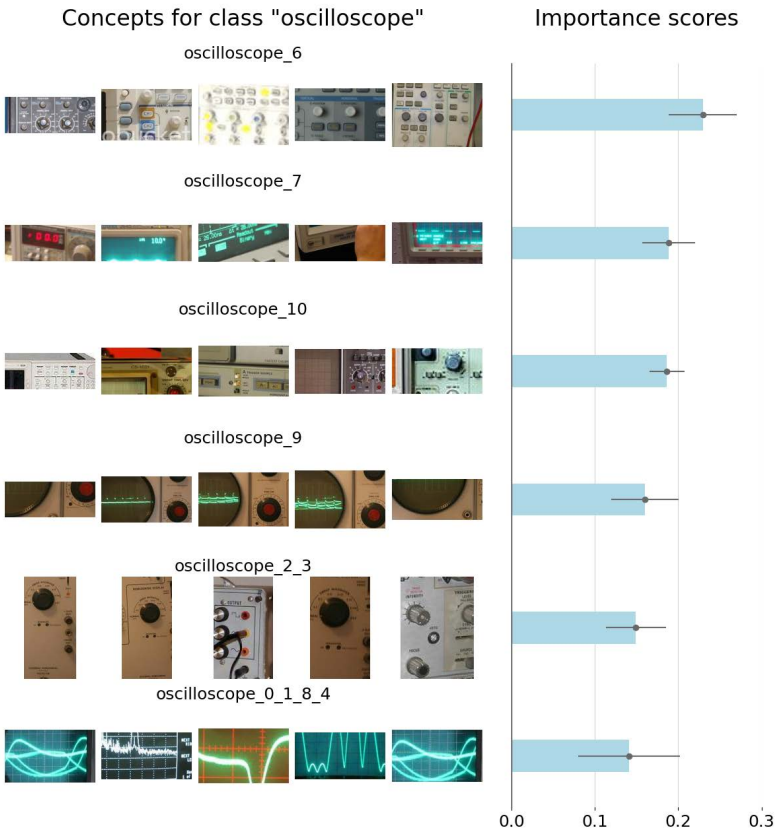


Figure 14. Global explanation for class “oscilloscope” with InceptionV3. The explanation was generated using Visual-TCAV and iterating over a set of 100 “oscilloscope” images from ImageNet.

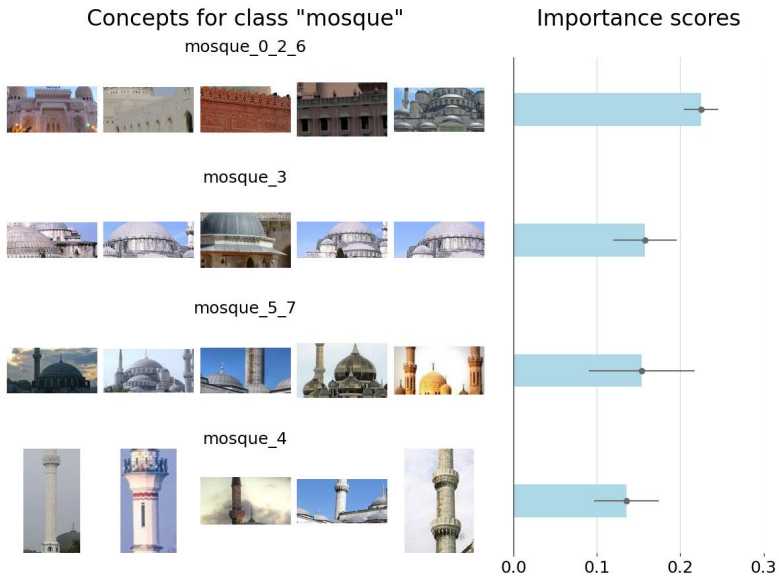


Figure 15. Global explanation for class “mosque” with InceptionV3. The explanation was generated using Visual-TCAV and iterating over a set of 100 “mosque” images from ImageNet.

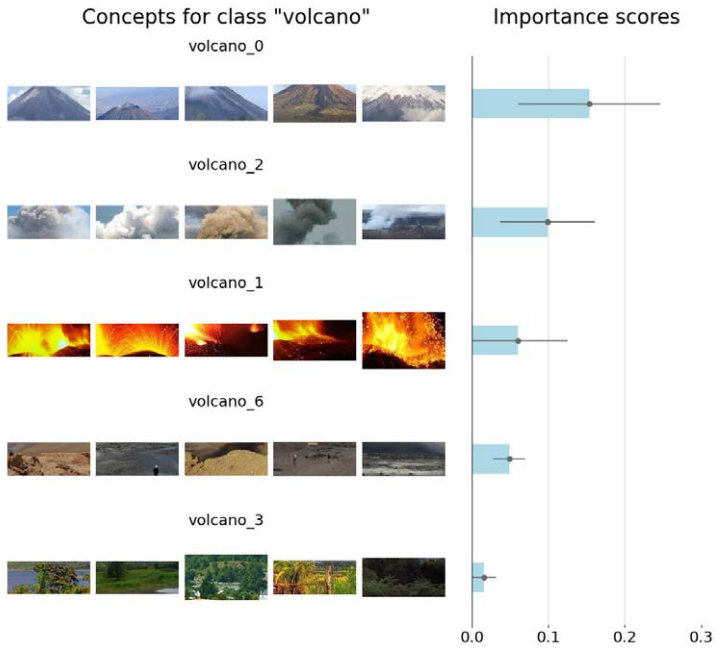


Figure 16. Global explanation for class “volcano” with ResNet50. The explanation was generated using Visual-TCAV and iterating over a set of 100 “volcano” images from ImageNet.

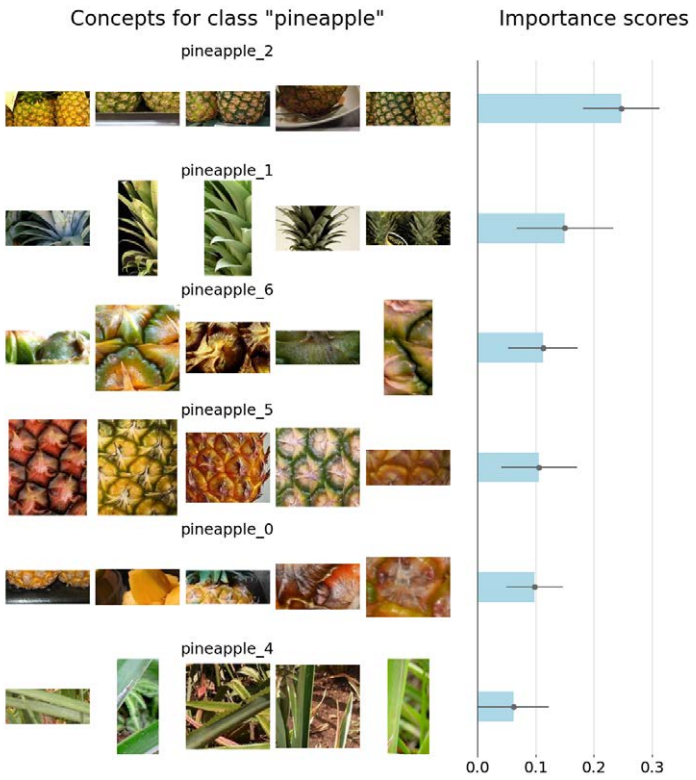


Figure 17. Global explanation for class “pineapple” with ResNet50. The explanation was generated using Visual-TCAV and iterating over a set of 100 “pineapple” images from ImageNet.

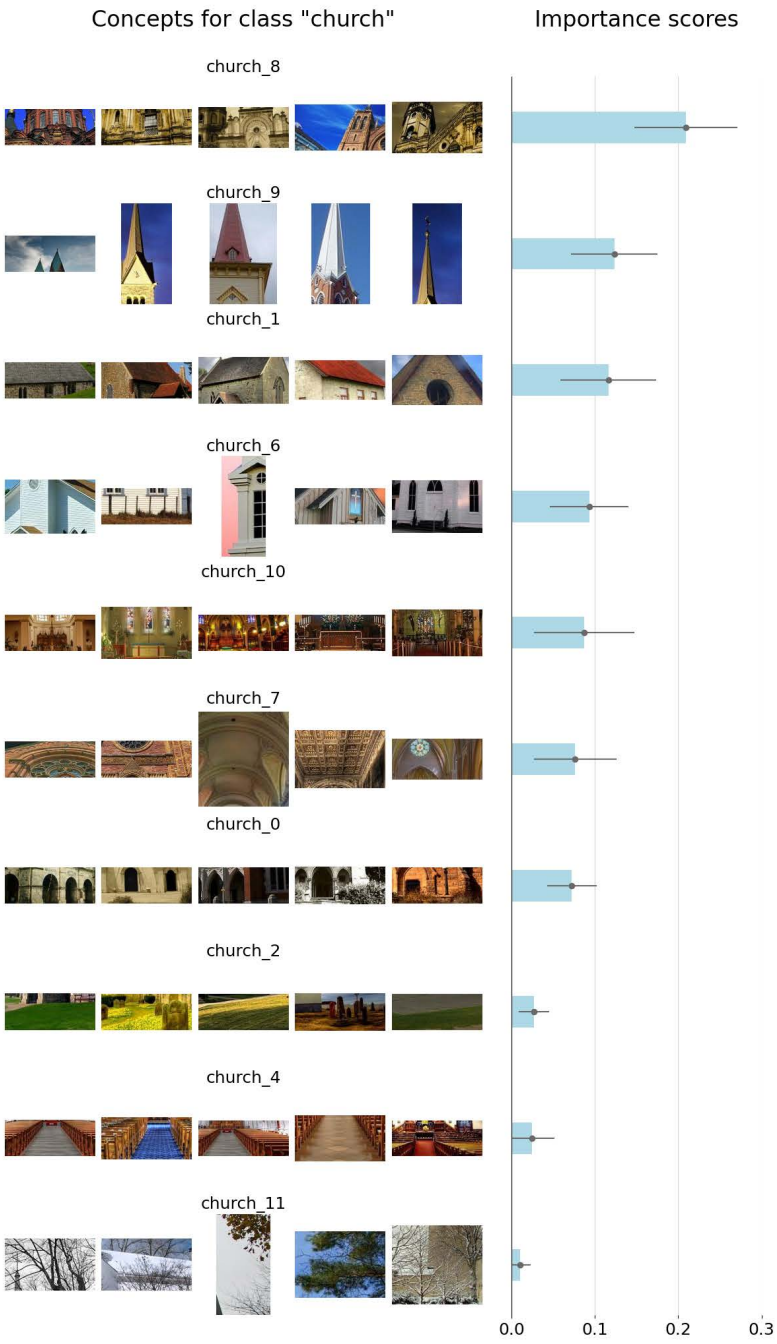


Figure 18. Global explanation for class “church” with ResNet50. The explanation was generated using Visual-TCAV and iterating over a set of 100 “church” images from ImageNet.

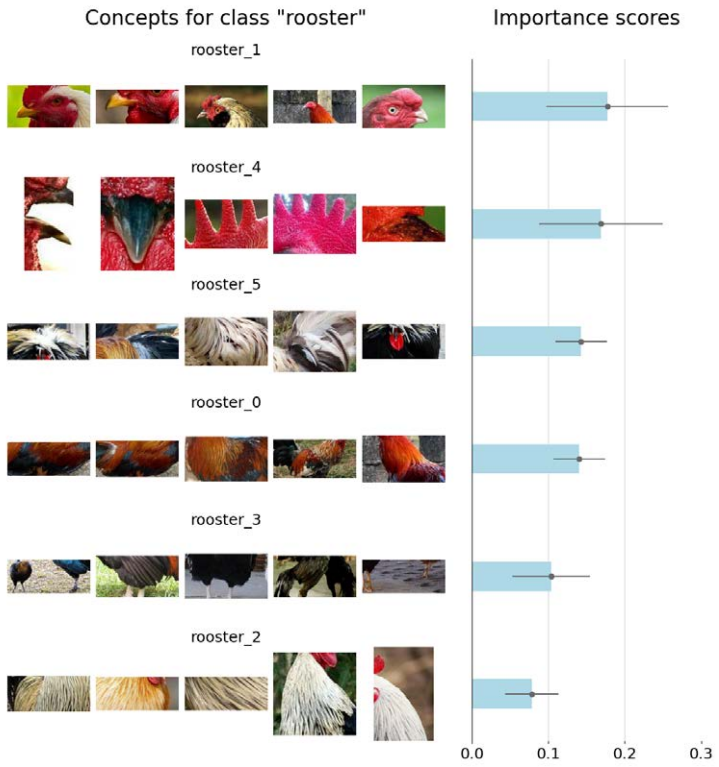


Figure 19. Global explanation for class “rooster” with VGG-16. The explanation was generated using Visual-TCAV and iterating over a set of 100 “rooster” images from ImageNet.

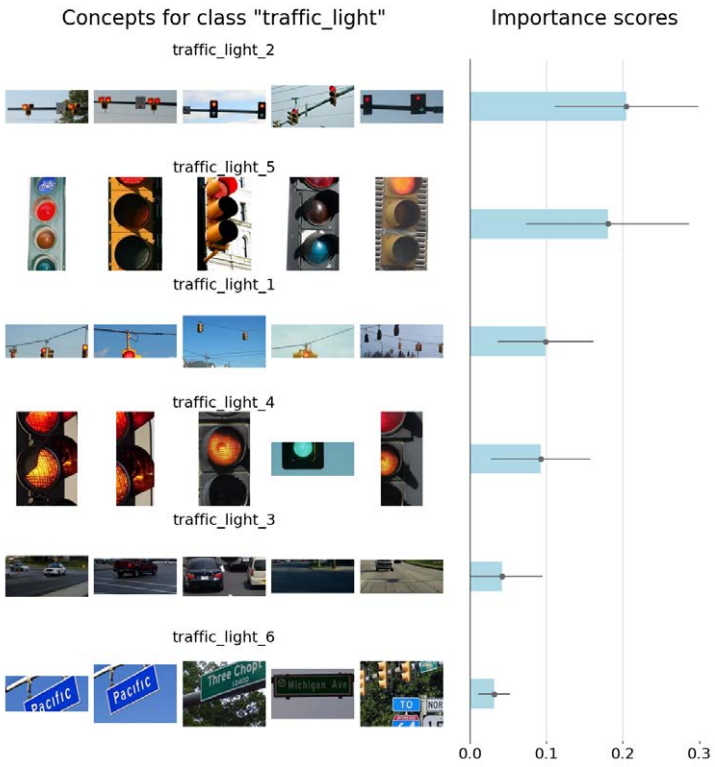


Figure 20. Global explanation for class “traffic light” with VGG-16. The explanation was generated using Visual-TCAV and iterating over a set of 100 “traffic light” images from ImageNet.

9. Examples of Local Explanations

pineapple_2 concept

Class	Attrib.
pineapple	**0.27**
ear	**0.058**
banana	**0.077**

pineapple_1 concept

Class	Attrib.
pineapple	**0.22**
ear	**0.069**
banana	**0.045**

pineapple_4 concept

Class	Attrib.
pineapple	**0.16**
ear	**0.11**
banana	**0.052**

(a) Image of class "pineapple" from ImageNet.

(b) Most important concept in local explanation.

(c) Second most important concept in local explanation.

(d) Third most important concept in local explanation.

Figure 21. Local explanation of an image of class "pineapple", generated with Visual-TCAV using the concepts extracted for class "pineapple" with ResNet50.

traffic_light_2 concept

Class	Attrib.
traffic_light	**0.18**
street_sign	**0.076**
trolleybus	**0.04**

traffic_light_7 concept

Class	Attrib.
traffic_light	**0.14**
street_sign	**0.12**
trolleybus	**0.14**

traffic_light_1 concept

Class	Attrib.
traffic_light	**0.11**
street_sign	**0.068**
trolleybus	**0.071**

(a) Image of class "traffic light" from ImageNet.

(b) Most important concept in local explanation.

(c) Second most important concept in local explanation.

(d) Third most important concept in local explanation.

Figure 22. Local explanation of an image of class "traffic light", generated with Visual-TCAV using the concepts extracted for class "traffic light" with ResNet50.

church_5 concept

Class	Attrib.
church	**0.26**
bell_cote	**0.18**
monastery	**0.24**

church_1 concept

Class	Attrib.
church	**0.23**
bell_cote	**0.11**
monastery	**0.13**

church_6 concept

Class	Attrib.
church	**0.19**
bell_cote	**0.13**
monastery	**0.14**

(a) Image of class “church” from ImageNet.

(b) Most important concept in local explanation.

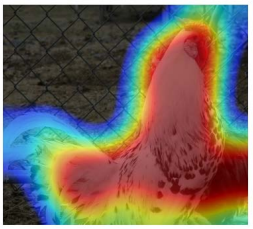
(c) Second most important concept in local explanation.

(d) Third most important concept in local explanation.

Figure 23. Local explanation of an image of class “church”, generated with Visual-TCAV using the concepts extracted for class “church” with VGG-16.

rooster_5 concept

Class	Attrib.
rooster	**0.16**
hen	**0.13**
partridge	**0.092**

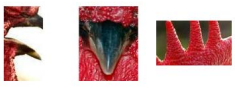


rooster_1 concept

Class	Attrib.
rooster	**0.14**
hen	**0.12**
partridge	**0.067**

rooster_4 concept

Class	Attrib.
rooster	**0.13**
hen	**0.12**
partridge	**0.058**



(a) Image of class “rooster” from ImageNet.

(b) Most important concept in local explanation.

(c) Second most important concept in local explanation.

(d) Third most important concept in local explanation.

Figure 24. Local explanation of an image of class “rooster”, generated with Visual-TCAV using the concepts extracted for class “rooster” with VGG-16.

10. Examples Questions from the Concept Matching Game

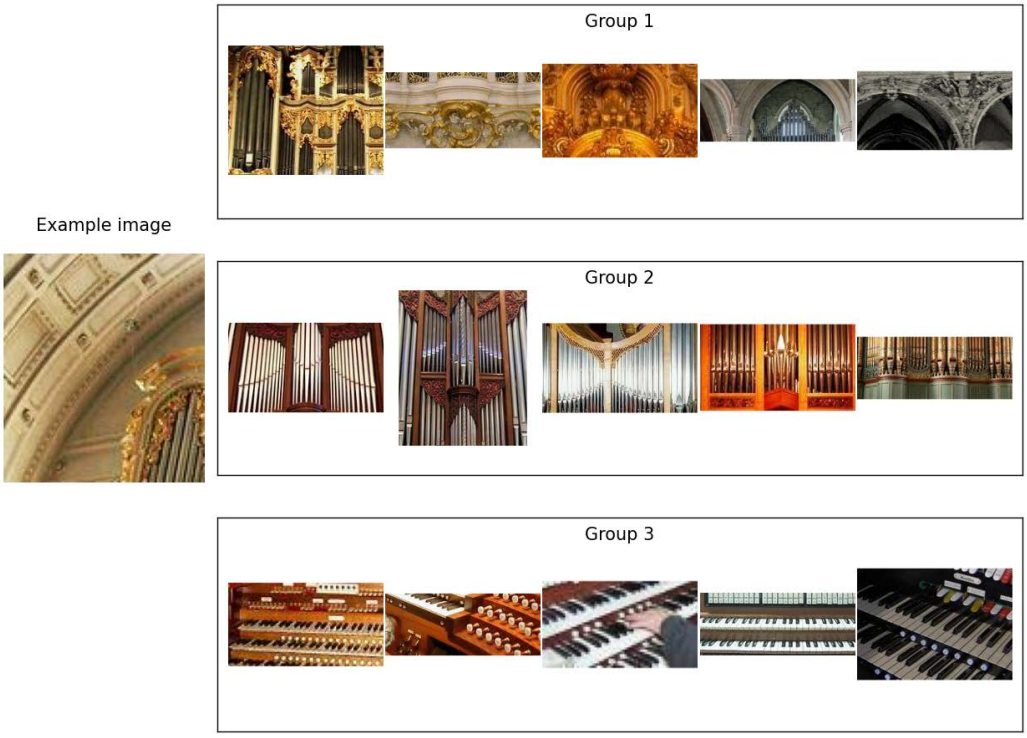


Figure 25. A sample question from the concept matching game, presenting an option for the class “organ”.

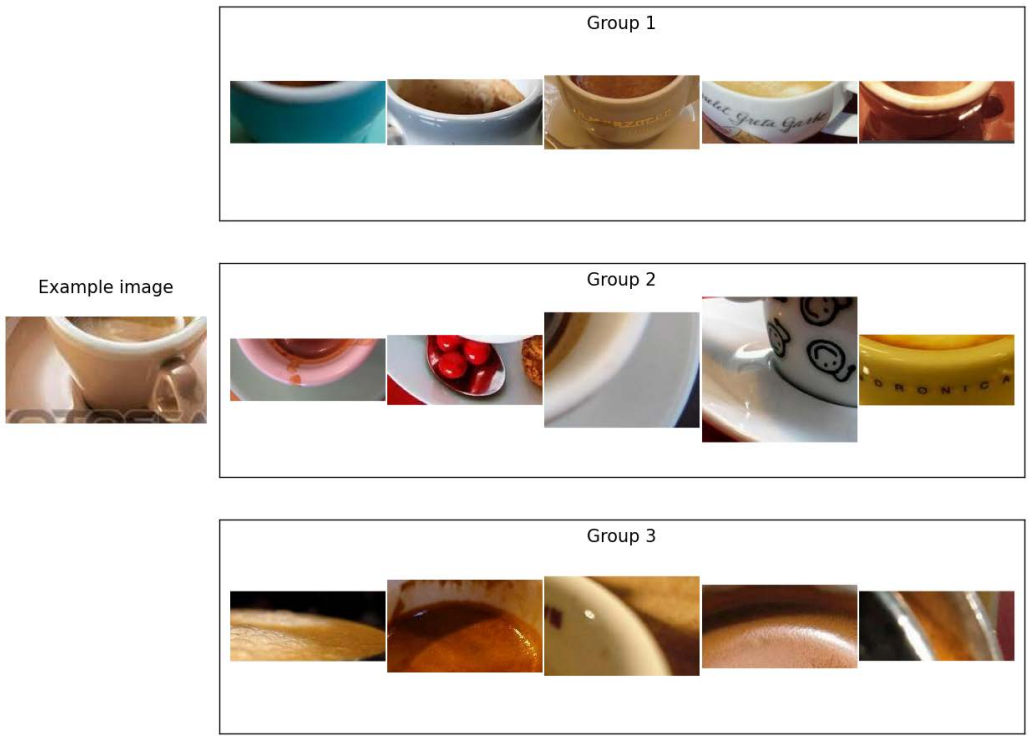
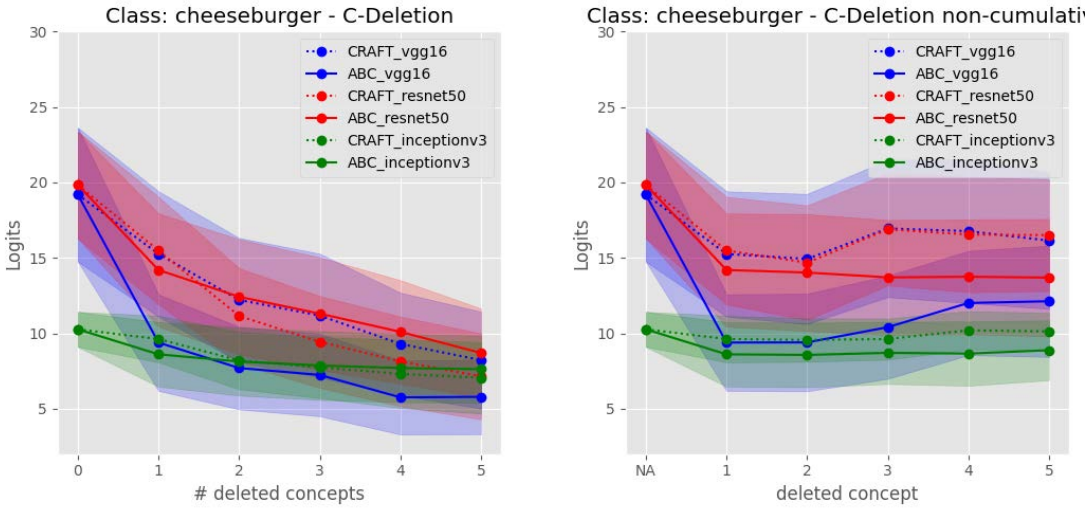


Figure 26. An example question from the concept matching game, showing a choice for class “espresso”.

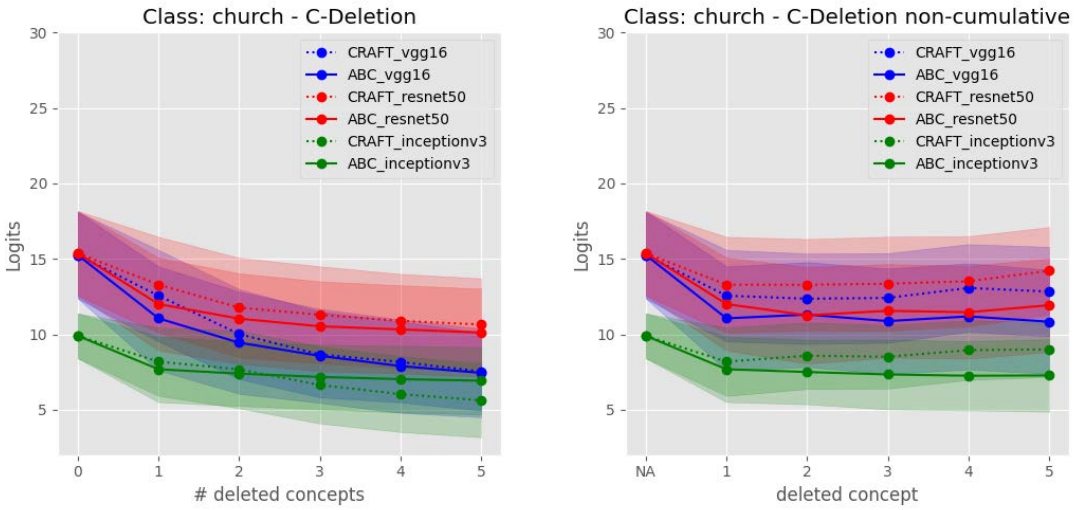
11. Fidelity Study's C-Deletion Plots



(a) Cumulative c-deletion plot.

(b) Non-cumulative c-deletion plot.

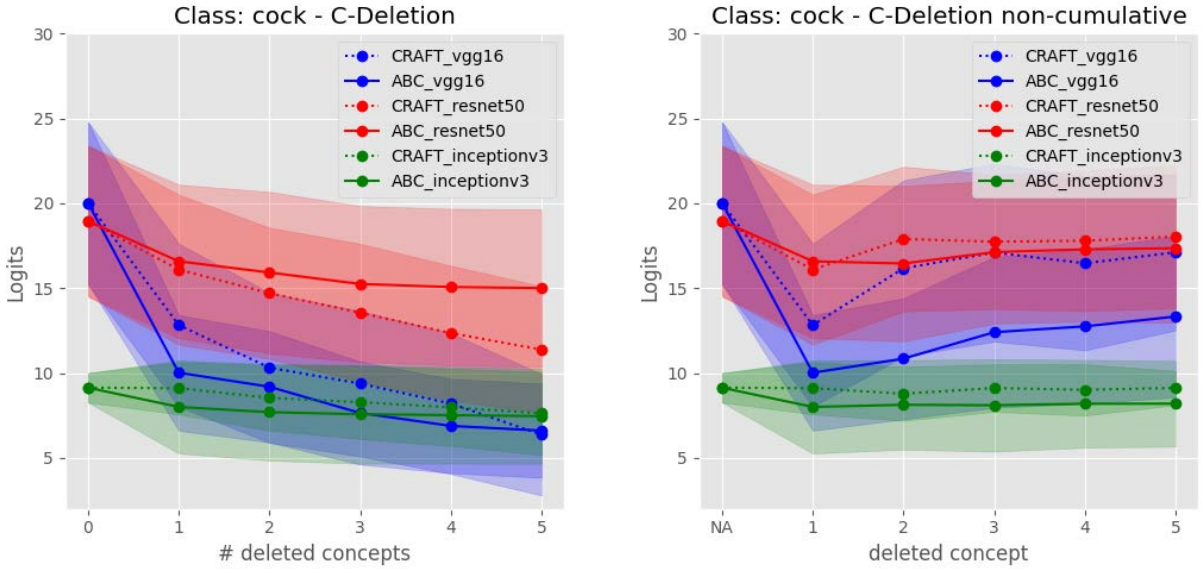
Figure 27. C-deletion results comparing ABC and CRAFT for class "cheeseburger" using 500 images.



(a) Cumulative c-deletion plot.

(b) Non-cumulative c-deletion plot.

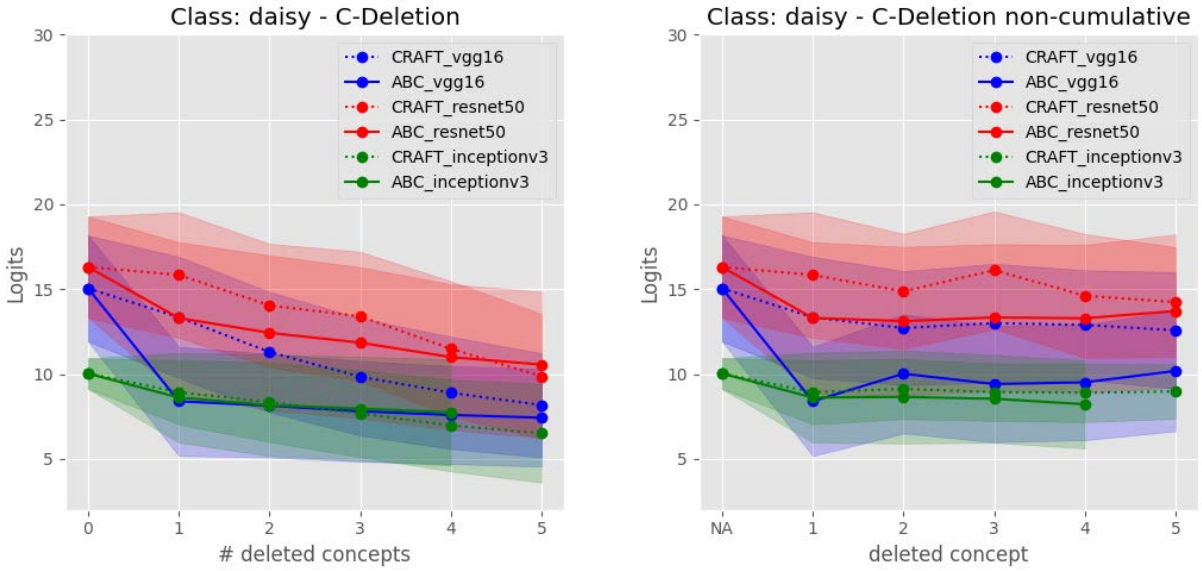
Figure 28. C-deletion results comparing ABC and CRAFT for class "church" using 500 images.



(a) Cumulative c-deletion plot.

(b) Non-cumulative c-deletion plot.

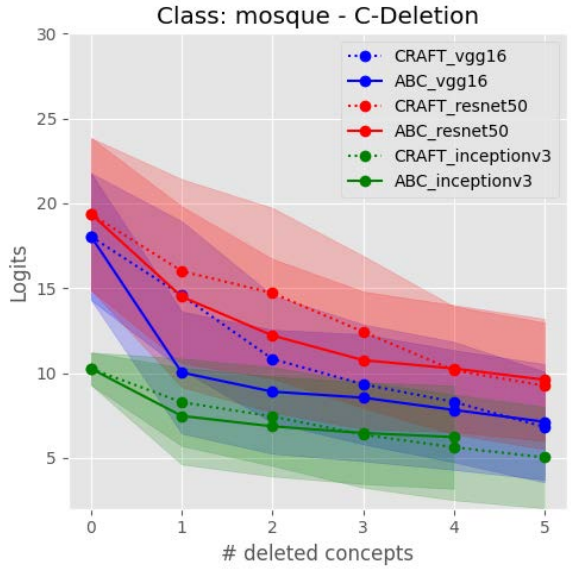
Figure 29. C-deletion results comparing ABC and CRAFT for class “cock” using 500 images.



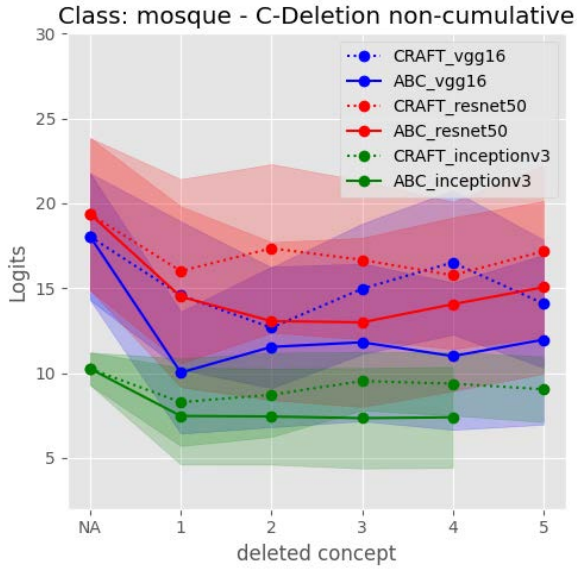
(a) Cumulative c-deletion plot.

(b) Non-cumulative c-deletion plot.

Figure 30. C-deletion results comparing ABC and CRAFT for class “daisy” using 500 images.

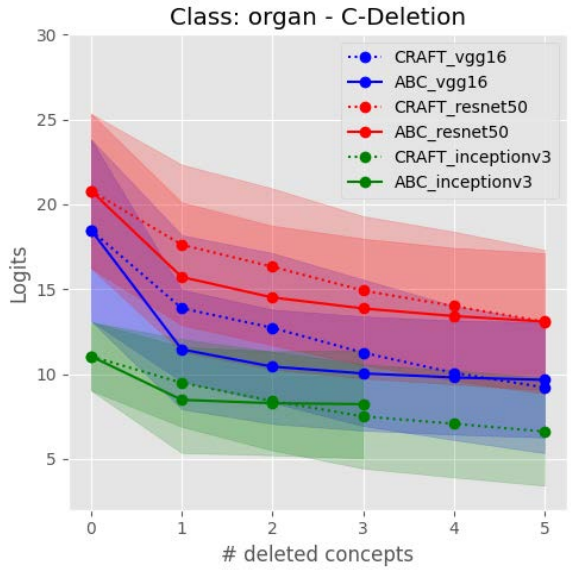


(a) Cumulative c-deletion plot.

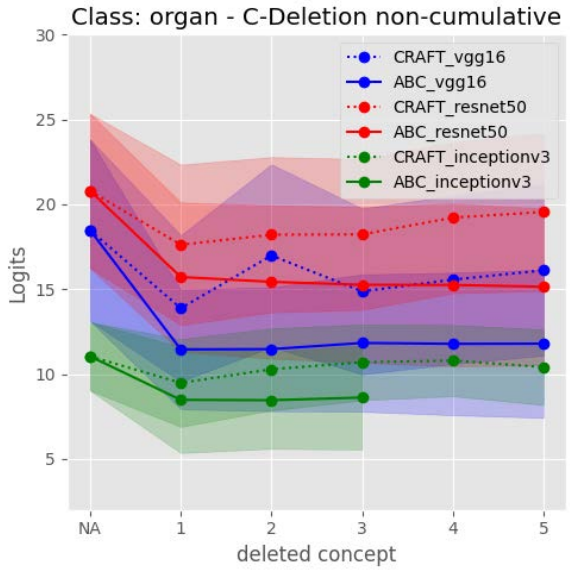


(b) Non-cumulative c-deletion plot.

Figure 31. C-deletion results comparing ABC and CRAFT for class “mosque” using 500 images.

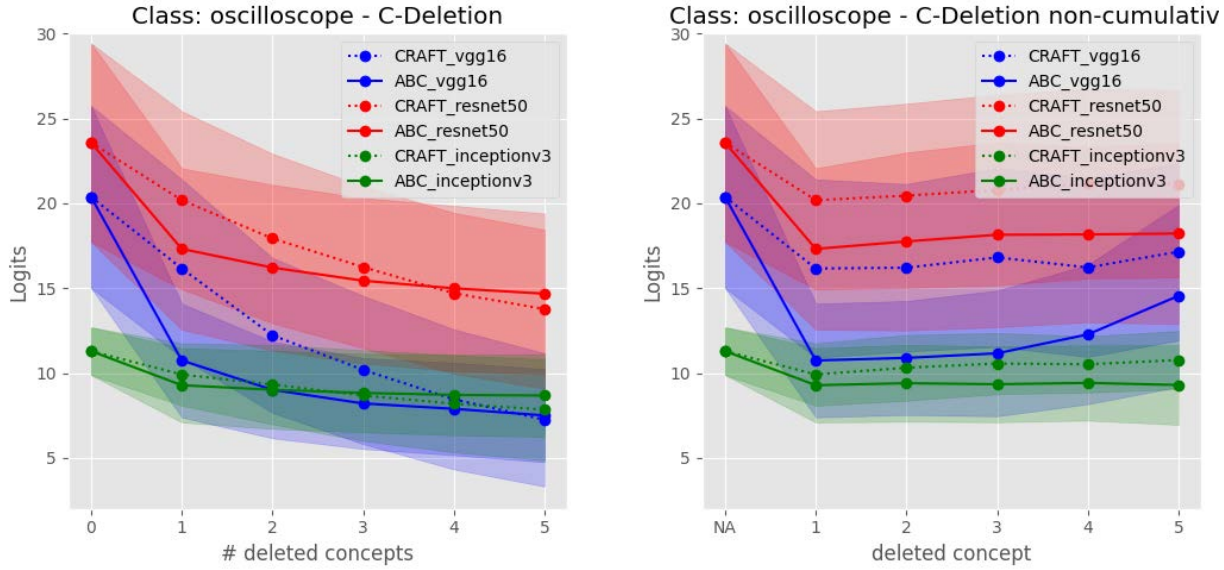


(a) Cumulative c-deletion plot.



(b) Non-cumulative c-deletion plot.

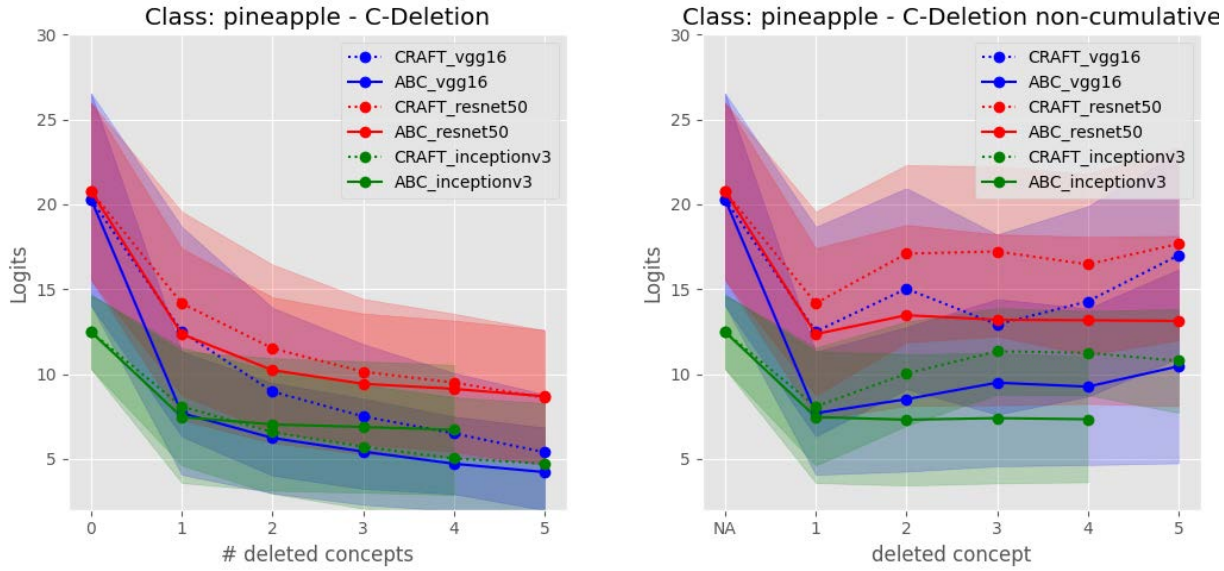
Figure 32. C-deletion results comparing ABC and CRAFT for class “organ” using 500 images.



(a) Cumulative c-deletion plot.

(b) Non-cumulative c-deletion plot.

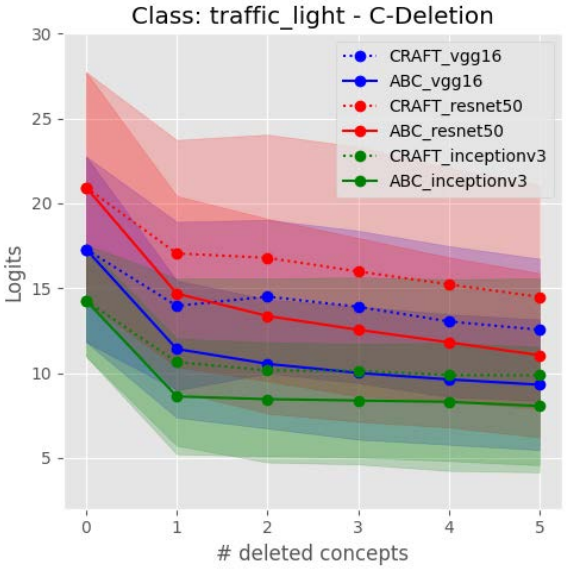
Figure 33. C-deletion results comparing ABC and CRAFT for class “oscilloscope” using 500 images.



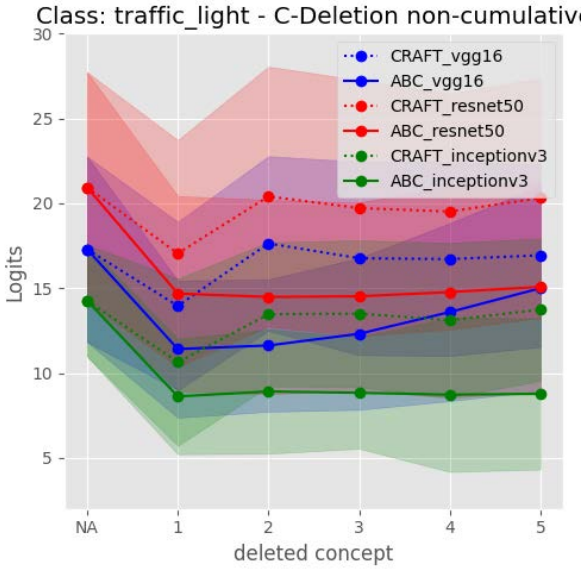
(a) Cumulative c-deletion plot.

(b) Non-cumulative c-deletion plot.

Figure 34. C-deletion results comparing ABC and CRAFT for class “pineapple” using 500 images.

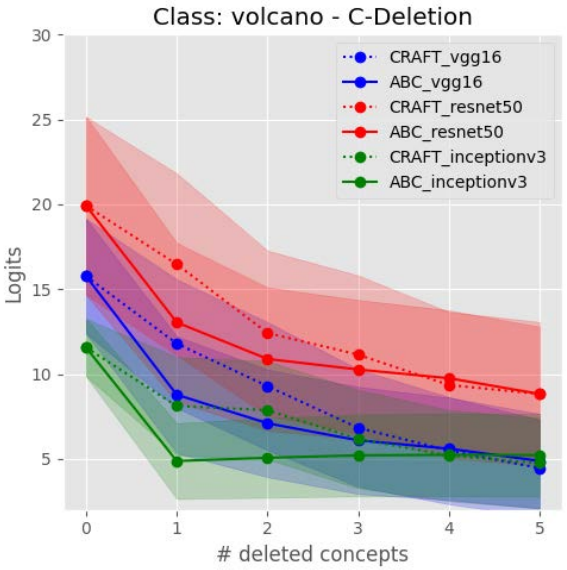


(a) Cumulative c-deletion plot.

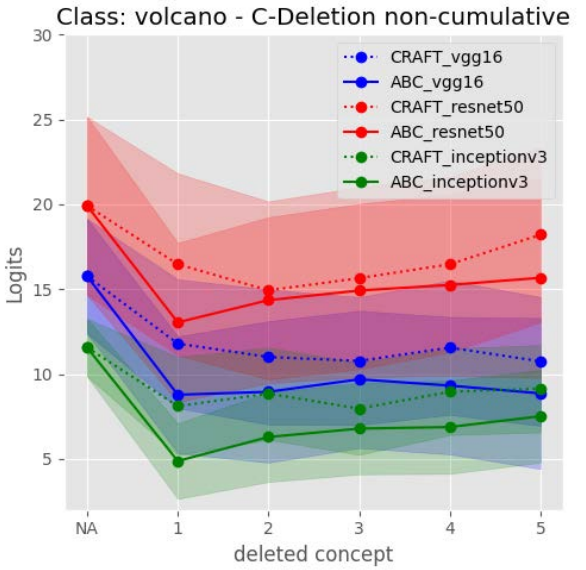


(b) Non-cumulative c-deletion plot.

Figure 35. C-deletion results comparing ABC and CRAFT for class "traffic_light" using 500 images.



(a) Cumulative c-deletion plot.



(b) Non-cumulative c-deletion plot.

Figure 36. C-deletion results comparing ABC and CRAFT for class "volcano" using 500 images.