Project Report: Group 10
Towards a Holistic Approach to Model Explainability

Aapo Kärki (s6235905), Niklas Magnusson (s6222048), and Verena Szojak (s6222064)

***Abstract*—Towards a holistic approach to Explainable Artificial Intelligence (XAI), this project investigates how the combination of various explainability methods can lead to a better understanding of deep learning models. Our objective is to study how different XAI techniques contribute to explaining a pre-trained Convolutional Neural Network and to assess its strengths and limitations. We aim to answer four research questions: (1) What parts of an image are important for the model? (2) What patterns and structures has the model learned? (3) What limitations does the model exhibit? (4) How can XAI techniques contribute to model improvement?**

We applied three distinct XAI methods: Invertible *Concept-based Explanations (ICE), Grad-CAM*, and *XRAI* on a pretrained VGG16 model trained on the Cifar10 dataset. These methods represent both feature-based and concept-based explainability strategies. Our analysis showed that the different methods highlight similar aspects of the model's behaviour, but also complementary aspects that can only be seen with specific methods. With ICE offering global concept-based explanations, Grad-CAM capturing local relevance via heatmaps and XRAI focusing on region-based segment attributions, we gathered a substantive analysis for the model.

We found that the model has learned about both the object and the background, as well as differences in color and some class-specific concepts. With all XAI-methods, the model showed weaknesses in classes *cat* and *dog*, suggesting potential areas for dataset improvements.

We provide a holistic interpretation of model reasoning and showcase the benefits of combining concept- and pixel-/region-level explanations. While our findings provide a glimpse into the model's behaviour, we recognize that further work is required to achieve a truly comprehensive and trustworthy model explainability.[1]

I. Introduction

Various Explainable AI (XAI) approaches have been introduced in the past that aim at shedding light on the black box of AI models [1]. These methods come with different advantages and drawbacks [2]–[4]. Feature-based techniques have been criticized as they are not as reliable, suffer from confirmation bias, and are not useful to humans for interpreting a model [2], [3]. Concept-based techniques are more intuitive to humans, as explanations are based on higher-level features, known as concepts [2], [5]. However, they also introduce problems such as the existence of an infinite number of concepts, lack of knowledge about suitable concepts, and a bias in concept selection [2], [4]. To overcome the issues of individual methods, we showcase the usefulness of a more holistic approach towards model explainability by combining three different explainability methods. We chose one concept-based XAI method (*Invertible Concept-based Explanations (ICE)* [6]) and two feature-based techniques (*Grad-CAM* [7] and *XRAI* [8]) and selected a pretrained CNN model as a use case. By aiming at gaining a deeper understanding of this model, we intend to answer the following research questions regarding the reasoning process of our model:

- What image parts are important for the model?
- What patterns and structures has the model learned?
- What limitations has the model?
- In what ways do the XAI techniques contribute to the model's improvement?

Explaining models using these approaches and applying several approaches on the same model has been done in the past. We present selected works: Foscarin et al. [9] create concept-based explanations for a composer classifier by deploying the supervised approach of Kim et al. [3] (*TCAV*) and the unsupervised approach of Zhang et al. [6] (*ICE*). Duvvuri et al. [10] apply *Grad-CAM* to a Diabetic Retinopathy classifier and Chen et al. [11] applied 6 feature-based XAI techniques, including *Grad-CAM* and *XRAI*, on a Tessaratoma papillosa pest detector to analyse and compare the results of the different methods. Similar to our approach, Paccotacya-Yanque et al. [12] compared 7 XAI techniques (4 feature-based and 3 concept-based, including *ICE* and *Grad-CAM*) and applied them to a model trained on a public skin-lesion dataset. Thus, we clearly state that our approach is not novel. The motivation of this project is to better understand these methods ourselves and to gain insights into the similarities and differences of the techniques' results for model explainability.

This project report is structured as follows: In Section II we introduce the 3 XAI techniques and in Section III we present the model and the dataset used for our experiments. This provides the baseline for Section IV where we describe our experiments, performed using the XAI methods, and illustrate the results for each of the methods. In Section V, we combine the findings to answer the research questions. Finally, Section VI critically reflects on general and XAI method-specific limitations that could reduce the relevance of the results and Section VII concludes the report.

[1]All code for the project is provided in a Github repository: https://github.com/aapokrki/t_xai_project.git

II. Explainablity Methods

In this section, we will provide the technical details about *ICE, Grad-CAM* and *XRAI* and state all used metrics as well as the visualization methods used to display important features and concepts.

A. Invertible Concept-based Explanations

Zhang et al. [6] introduce *ICE*, calculating *Concept Activation Vectors (CAVs)* [3] in an unsupervised fashion for CNNs. A trained CNN can be applied in two steps. They call these two parts feature extractor $A_l = E_l(I)$ and classifier $Y = C_l(A_l)$. The feature extractor E_l takes input images I, passes them through the trained model, and returns activation matrix A_l for layer l. A_l can be fed directly into the classifier C_l to continue the forward pass and return predictions. Alternatively, A_l serves as input for the *reducer* where CAVs and an approximation for A_l (A'_l) are calculated. A'_l serves as input for the classifier $C_l(A'_l)$ to return logits. We implemented the method from scratch according to the description in their paper.

1) Reducer: Using NMF, more generally called the *reducer* [6], explanations based on concepts for a given activation layer of a CNN are provided. The subscript l for the layer is omitted here for better readability. A ReLU-activated output of a convolutional layer $A \in \mathbb{R}^{(n,h,w,c)}$ (batch, height, width, channels) is the starting point. NMF requires a two-dimensional input [13]. Thus, the first three dimensions are collapsed into one $V \in \mathbb{R}^{(n \times h \times w, c)}$. V serves as input for NMF[2] which results in two matrices, "feature score $S \in \mathbb{R}^{(n \times h \times w, c')}$ and feature direction $P \in \mathbb{R}^{(c', c)}$" [6]. S indicates the presence of a concept while P defines how the concepts look and holds the CAVs. c' is the hyperparameter for the number of concepts that is used for NMF. An approximation of V is obtained, $V' \approx SP$. When expanding again the dimensions of V' to $A' \in \mathbb{R}^{(n,h,w,c)}$, the approximated activation layer can be used as input for the classifier C.

2) Explainer: The explainer uses NMF with fixed P and a different layer, A_e. Instead of fitting new CAVs, done in the reducer, the previously computed concepts P are reused. With V_e and P, a new feature score matrix S_e is calculated. It holds the information on where the previously computed concepts are in this particular activation layer.

3) Metrics: Throughout the experiments with *ICE*, we use the following metrics: **Average Concept Presence.** This metric calculates a score that relates to the strength of a concept in a sample [6], [9]. We take the feature score matrix $S \in \mathbb{R}^{(n \times h \times w, c')}$ and reshape it to $S' \in \mathbb{R}^{(n,h,w,c')}$. The average of height h and width w then leads to a scalar for each of the n samples in the batch and for each concept c'.

$$\text{ACP}_{n,c'} = \frac{1}{h \times w} \sum_h \sum_w S' \tag{1}$$

[2]We used https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html with default parameters and a fixed integer for `random_state`.

We use it in the *ICE* experiments to sort and filter prototypical and atypical samples for a concept.

Cosine Similarity. Using NMF for unsupervised feature extraction has the drawback of selecting the hyperparameter `number of concepts` [14]. As this number is unknown, we explore the cosine similarity as a means of determining the number of concepts for NMF. Previous works have applied this similarity measure to concepts [15]–[18] and use it to measure the distance of concepts and to determine how similar concepts are. As our concepts are in the same feature space [17], the cosine similarity appears to be a reasonable measure. It is defined for two concept vectors $(v^l_{c_1}, v^l_{c_2})$ of the lth activation layer as the normalized dot product between $v^l_{c_1}$ and $v^l_{c_2}$.

$$\text{Cosine Similarity}(v^l_{c_1}, v^l_{c_2}) = \frac{v^l_{c_1} \cdot v^l_{c_2}}{\|v^l_{c_1}\| \|v^l_{c_2}\|} \in [-1, 1] \tag{2}$$

A value of 1 corresponds to the vectors being maximally similar and pointing in the same direction, 0 indicates orthogonal vectors, and -1 implies the vectors pointing in opposing directions [15], [18]. When training NMF with a specific value for `number of concepts`, we use the cosine similarity to measure how similar the feature directions $P \in \mathbb{R}^{(c', c)}$ are, where each row corresponds to a CAV of length c.

Testing with Concept Activation Vectors. CAVs can be used to determine whether a specific concept is relevant for the prediction of a specific class. The procedure is introduced as *Testing with Concept Activation Vectors (TCAV)* by Kim et al. [3]. We implemented the metric according to Equations 3 and 4. To measure how sensitive a single image or a class is with respect to a specific concept, the directional derivative of the logit for a specific sample in the direction of the CAV is used. Whenever this derivative is positive, the concept is useful for classifying a sample as a specific class. Formally, the lth activation layer $f_l(x)$ for input images x is taken, previously denoted as A_l, and the directional derivative $S_{c,k,l}(x)$ with respect to a specific CAV v^l_c holding concept c in layer l is calculated as the difference between the logits $h_{l,k}(x)$ for the kth class using activation layer l in combination with the added CAV v^l_c and without it.

$$S_{c,k,l}(x) = \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(x) + \epsilon v^l_C) - h_{l,k}(f_l(x))}{\epsilon} \tag{3}$$

To finally obtain the TCAV score for a concept c, Kim et al. [3] simply calculate the fraction of directional derivatives $S_{c,k,l}$ for a set of images X_k where the CAV v^l_c has a positive influence.

$$\text{TCAV}_{c,k,l} = \frac{|\{x \in X_k : S_{c,k,l}(x) > 0\}|}{|X_k|} \in [0, 1] \tag{4}$$

A TCAV score of 1 indicates that the model uses a certain concept for making the predictions for all images from the set X_k and conversely, a score of 0 implies the concept is not present in any of the images. After exploring different values for the hyperparameter ϵ, we set it to `1e-6`.

4) Visualization of Concepts: Inspired by the visualization techniques of Zhang et al. [6] and their repository[3], we use the following technique for visualizing images and their concepts. The matrix $S \in \mathbb{R}^{(\text{n x h x w, c'})}$ is used, holding the information on where the concepts are. First, the dimensions of S are transposed in order to interpolate S to get back to the original image sizes: $S \in \mathbb{R}^{(n \times h \times w, c')} \rightarrow S \in \mathbb{R}^{(n, c', h, w)}$. Using the provided Pytorch `interpolate` function, S is interpolated and upscaled to $S \in \mathbb{R}^{(n, c', 32, 32)}$, where height and width correspond to the original image dimensions. For each image in the batch of size n and each concept c', visualizations are created by normalizing the corresponding matrix from S to the range $[0, 1]$. A threshold of 0.7 is then applied to filter the regions with a high concept presence. These are then plotted with an alpha value of 1 while regions with a lower presence are displayed with an alpha value of 0.7. With these specifications, the high-presence regions are highlighted.

B. Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a widely used XAI technique introduced by Selvaraju et al. [7] for visualizing the decision-making process of CNNs. It highlights the input that influences the prediction of the model the most for a given class by using the gradients of the target class flowing into the chosen convolutional layer. *Grad-CAM* produces heatmaps that are overlaid on top of the original image to show important areas of that image.

1) Method: For a selected CNN model and a specific class c, Grad-CAM computes the gradient of the class score y^c with respect to the activations A^k of a chosen convolutional layer, where A^k denotes the kth feature map at that layer. These gradients are global-average-pooled over the width and height dimensions (i, j) to obtain importance weights α_k^c for each feature map:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{5}$$

where Z is the total number of spatial locations in the feature map. The class activation map $L^c_{\text{Grad-CAM}}$ is then computed as a weighted combination of the forward activation maps, followed by a ReLU operation to retain only the features that have a positive influence on the class of interest:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \tag{6}$$

This map is upsampled to the size of the input image and overlaid as a heatmap to provide visual explanations of the model's decision.

C. XRAI

XRAI is a local region-based attribution method used to explain which regions in an image contribute the most to the prediction of a deep neural network [8]. *XRAI* uses larger segments of an image and calculates the total importance of a certain segment. The reason for doing this is that it is easier for humans to understand explanations regarding segments than just pixels [19]. The aim of *XRAI* as a method to explain neural networks is to find the smallest region possible that predicts the class of the image correctly and therefore help humans understand what features are important for the model [8]. The method of *XRAI* follows the higher-level steps in Algorithm 1 [8]. We implemented the method ourselves.

Algorithm 1 XRAI

0: **Given** image I, model f and attribution method g
0: Over-segment I to segments $s \in S$
0: Get attribution map $A = g(f, I)$
0: Let saliency mask $M = 0$, trajectory $T = []$
0: **while** $S \neq \emptyset$ **and** $area(M) < area(I)$ **do**
0: **for** $s \in S$ **do**
0: Compute gain[2]:

$$g_s = \sum_{i \in s \setminus M} \frac{A_i}{area(s \setminus M)}$$

0: **end for**
0: $\hat{s} = \arg\max_s g_s$
0: $S = S \setminus \hat{s}$
0: $M = M \cup \hat{s}$
0: Add M to list T
0: **end while**
0: **return** T =0

1) Segmentation: The *XRAI* method segments the input image using the graph-based algorithm [8]. In image segmentation, the algorithm computes the difference in color between each pair of neighboring pixels (v_i, v_j) to create a weight for each of these edges $(w(e) \in E)$ to include similar pixels in the same segment [20]. A low weight for an edge means that the pixels are similar. In the beginning, all pixels are individual segments (C). It starts by calculating the internal difference of a segment $\text{Int}(C)$ by finding the largest weight for an edge in the minimum spanning tree of the segment $\text{MST}(C, E)$:

$$\text{Int}(C) = \max_{e \in \text{MST}(C,E)} w(e) \tag{7}$$

The next step it to calculate the difference between two segments $Dif(C_1, C_2)$ by finding the lowest edge weight between two neighboring pixels, one from each segment:

$$\text{Dif}(C_1, C_2) = \min_{\substack{v_i \in C_1,\, v_j \in C_2 \\ (v_i, v_j) \in E}} w(v_i, v_j) \tag{8}$$

In the case of there not being an edge between C_1 and C_2, the difference is set to $+\infty$. A predicate, D, to justify the existence of a boundary between two segments is formulated in the following way:

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if Dif}(C_1, C_2) > \text{MInt}(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases} \tag{9}$$

[3]https://github.com/zhangrh93/InvertibleCE

where $\text{MInt}(C_1, C_2)$ is the minimum internal difference of the two segments. This is defined as:

$$\text{MInt}(C_1, C_2) = \min\left(\text{Int}(C_1) + \tau(C_1),\ \text{Int}(C_2) + \tau(C_2)\right) \tag{10}$$

The term τ is a threshold function

$$\tau(C) = \frac{k}{|C|} \tag{11}$$

including a pre-established constant k that sets a scale of observation. A larger k makes the method prefer creating larger segments.

The algorithm continues through the segments and merges them until the predicate $D(C_1, C_2)$ for a boundary is true for all adjacent segments.

The segments created by this process often exclude edges of objects in images because the edges might differ in color from the object. Therefore, the segments are dilated by five pixels to make sure that the edges are included in the segments [8].

2) Segment attribution using integrated gradients: Using integrated gradients in the input layer, the contribution of every pixel to the prediction can be calculated using the following equation [8].

$$IG_i(x) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i}\, d\alpha \tag{12}$$

Where $IG_i(x)$ is the attribution of each pixel, x_i and x_i' individual pixels and their baseline value respectively and F is the model. A positive attribution indicates that the pixel contributes to the prediction.

The attribution of each segment is then calculated using the gain function in Algorithm 1 that sums the individual pixels attribution A_i to a total contribution of the parts of of the segments that is not stored yet ($area(s \backslash M)$). The highest contributing segment is then stored in the variable M. For the next iteration, the highest contributing segment is used as input, and the algorithm goes through the segments again to find the segment that increases the gain the most when combined with the previous segment. This is done iteratively until the union of the segments covers the whole image. In every iteration, the best union of segments is stored, which gives the user the ability to choose their specific area threshold and proceed with their explanation. The chosen union of segments is then placed on top of the original image and makes only the pixels present in the segmentation visible for explanation. A common approach is to use the explanation that includes the least amount of pixels, but still gives the correct explanation.

3) Evaluation metrics: To evaluate the *XRAI* explanations, we ran the method until the model could predict the correct label and calculated the area of the image needed to do so. By doing it this way, we made sure that the explanation included segments contributing to the correct prediction without the risk of including segments that were not important. Since the images in the dataset are very different, the area needed for each is not the same for every data point. This meant that using a strict threshold of the amount of pixels in the explanation could cause some explanations to include unimportant segments. A large area needed for the model to predict the correct class could mean that the model is having difficulties extracting the most important features of the data point.

III. Experiment Setup

The following section introduces the image classifier and the dataset that we used in our experiments.

A. Dataset and Preprocessing

We chose the Cifar10 dataset[4] for our experiments. The dataset consists of 60000 RGB images of size 32x32. Train and test splits are already provided. The images can be divided into ten different classes (*airplane, automobile, bird, cat, deer, dog, frog, horse, ship,* and *truck*). All images were normalized before model inference as these values were used during model training, using means: $[0.4914, 0.4822, 0.4465]$ and standard deviations: $[0.2023, 0.1994, 0.201]$.

B. Model Architecture and Training

We used a publicly available pretrained VGG16 model[5]. It was trained for 193 epochs on the Cifar10 dataset using the mentioned means and standard deviations and we found no indications of additional data augmentation techniques in the Github repository. The model achieved an overall accuracy of 94.16% with highest accuracies for classes *automobile* 97.30% and *horse* 97% and lowest ones for *cat* 87.20% and *dog* 90.10%.

IV. Experiments and Results

This section describes the individual experiments we performed with the three XAI methods and the corresponding results by analysing heatmaps, segments and additional metrics.

A. Invertible Concept-based Explanations

We use the training set in our experiments for training the reducer and extracting the concepts. This is important for the reducer and aligns with the findings of Ramaswamy et al. [5] who emphasize to maintain the same dataset distribution for creating explanations and model training. In addition, we keep incorrectly classified samples to potentially detect learned model biases [21]. We decided using an intermediate layer because concept in later layers are too coarse [22] and lower layers tend to focus more on simpler features [23]. We decided on the activation layer in the VGG16 feature extractor used for extracting the concepts after conducting the following analysis: We looked into different model layers, activated with one unbalanced training batch (512 samples),

[4]Cifar10 dataset: https://www.cs.toronto.edu/~kriz/cifar.html

[5]Repository of the pretrained VGG16 model: https://github.com/chenyaofo/pytorch-cifar-models.git

and extracted ReLU-activated maps because NMF requires a non-negative matrix as input [13]. For each of these 4-dimensional activation maps, we started with 20 concepts and decreased this number as long as the cosine similarity of CAVs was above a certain threshold $t \in [0.1, 0.2, \ldots, 0.5]$, ensuring that the learned concepts are dissimilar to a certain degree. By reducing the number of concepts, we intended to make concepts understandable to humans [5] and adhere to the *Co-12 explanation quality property* of explanation compactness [24]. For the tested layers, the analysis resulted in either 1 or 20 concepts, except for layer 29, where we received an optimal number of 10 concepts with a threshold of 0.3. Thus, we fixed the layer for our explanations to layer 29 and extracted 10 concepts from the first training batch that are then applied to our test set. We set a seed and did not shuffle the datasets to ensure reproducibility of the results.

In a next step, we applied the explainer with fixed concepts to all test samples and calculated for each sample the average concept presence for each concept. Finally, we extracted the 5 prototypical samples for each concept having the highest ACP score and the 5 atypical samples with the lowest score. Onto these samples, we overlayed the concepts, as outlined in Section II-A4 and plotted the images.

Concept	Description	Top 5 Classes
1	body & face of bird	bird
2	car windows	automobile
3	car part between wheels	automobile
4	belly of horse (brown & grass)	horse
5	middle of ship (hull & mast)	ship
6	truck front with connection to trailer	truck
7	area close to frog eye (tympanum)	frog
8	face of deer, between antler	deer
9	cockpit, center of plane, part of wing	airplane
10	cat face (focus on nose and neck)	cat

TABLE I: Table describing the extracted concepts according to the highlighted regions in the top 5 samples from the test set for each concept and stating the classes of the top 5 images. One concept is learned for each class (exception for the *dog* class).

In Figure 9a, the prototypical examples for each concept are plotted. In Table I we provide a short description of what each concept could be. Generally, we see a tendency that one concept represents exactly one class and is strongly present in images of a specific class. The only exception is class *automobile* that is represented by two concepts and class *dog* that does not appear among any prototypical samples. From the highlighted parts, we can derive that the model looks primarily at some specific subparts of objects. For Concept 4, the background is also important as both, the horse's belly and the grass behind it are highlighted. In addition, we can provide a categorization of concepts into the following groups: *class-specific concepts, contrasts and transitions,* and *broader concepts*. We find four concepts (2, 6, 7, 8) that could define a unique characteristic of a specific class: Concept 2 highlights the region of the windows of automobiles which resembles a specific location and color. Concept 6 emphasizes the space between the front of the truck and the trailer. Concept 7 highlights the area around a frog's eye where frogs have their tympanum which is an oval shaped membrane [25] and Concept 8 highlights the face of a deer with the space between its head and its antler. In the second group, we find three concepts (3, 4, 5) that appear to look at transitions between objects or colors: Concept 3 highlights the region between the wheels and covers at least one wheel and parts of the lower half of an automobile. Concept 4 describes this contrast between horse belly and green grass in the background. Concept 5 looks at the middle part of a ship where we see a darker hull and a lighter upper part (potentially with a mast). In the last group, we place Concepts 1 and 10 which lack one defining feature: Concept 1 is present at different parts of a bird (body, throat, or face) and Concept 10 occurs either at a cat's face (the nose and its surrounding) or at a cat's neck.

To get a deeper understanding of what a concept comprises and about the presence of each concept in each class, we provide the TCAV scores for all test samples. We grouped the activation maps for each class together in one batch and calculated the TCAV score for all concepts, see Figure 9b. In general, TCAV scores provide mixed results. For some CAVs, the class of prototypical concepts coincides with the class where the concept is primarily present (see Concepts 2 and 7). However, we also see an indication that the concepts are rather general. They appear in several classes and for some concepts, the prototypical class does not coincide with the class providing the highest TCAV score for this particular concept. An example is Concept 1 where the prototypical samples are birds, however the highest TCAV score for this concept occurs in the *airplane* class or Concept 6 which is very typical for trucks, but is not the most prominent concept in the *truck* class. Besides, we see that concepts represent some classes better than others as classes *bird, dog, horse* and *ship* have TCAV scores below 0.5 for all CAVs.

B. Grad-CAM

We used a pre-made PyTorch Grad-CAM library[6] for the *Grad-CAM* implementation. The method was used on the first 10 correctly classified images of every label in the test set. We focused only on correctly classified test images to explore the model's reasoning when it is successful in classifying images. By limiting the analysis to correct predictions, we can better see if the explanations match the important parts of the image and can potentially detect dataset limitations.

For the method, we chose the convolutional layers for the explanation to be layers 22 and 29. The average heatmap of these layers is the final output. The average of these layers were chosen because both showed reasonable heatmaps in some test images but not for others, and vice versa. So an average of these seemed reasonable for the explanations.

As shown in Figure 1, with some labels the heatmaps focus more on the background information while others focus on

[6]Repository of the PyTorch Grad-CAM library: https://github.com/jacobgil/pytorch-grad-cam

the actual object in the image. For example, with the *airplane* label a majority of the heatmaps are either in the background sky, the ground, or just the tip of the airplane wing. Other labels, like the *automobile* have more coherent heatmaps on the actual label object in areas like the windshield, the tires or the bumper.

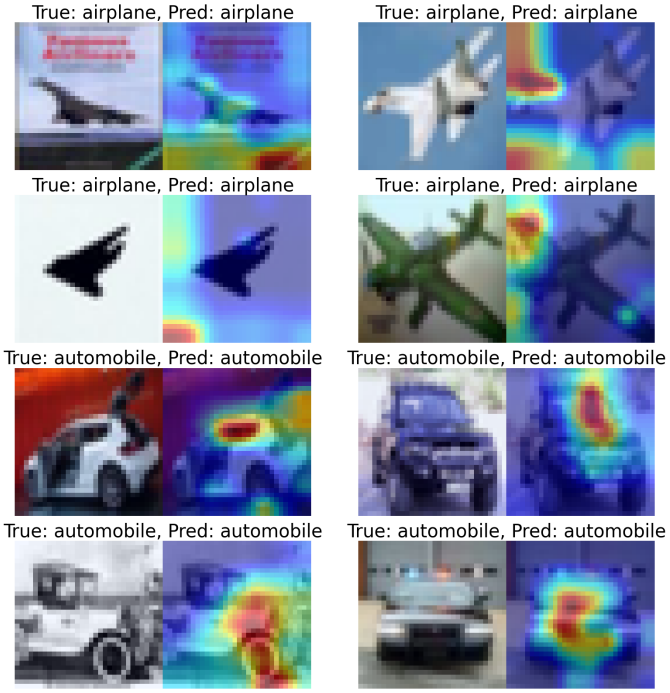


Fig. 1: Grad-CAM visualizations for the *airplane* and *automobile* labels.

Some labels have mixed results, and the heatmap's focus differs with the input image. For example in Figure 2, the *bird* label test images have the heatmap focused either on the bird's head, the beak or only on specific points in the background.

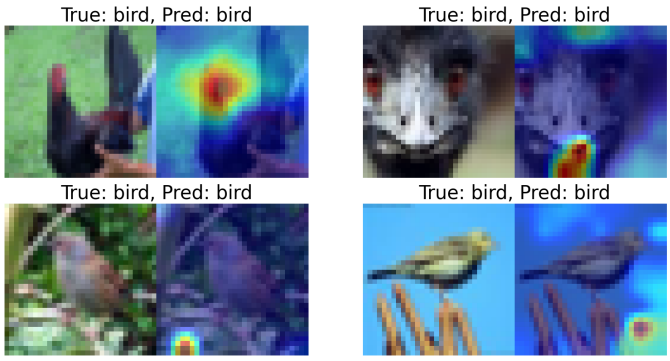


Fig. 2: Grad-CAM visualizations for the *bird* label across selected test images.

Similar mixed results can be seen with the *dog* label as shown in Figure 3. In some test images, the model focuses on the dog's color patterns or color differences between foreground and background. In other images, with different types of dogs from a different perspective, the heatmaps focus around the facial area or the whole body.

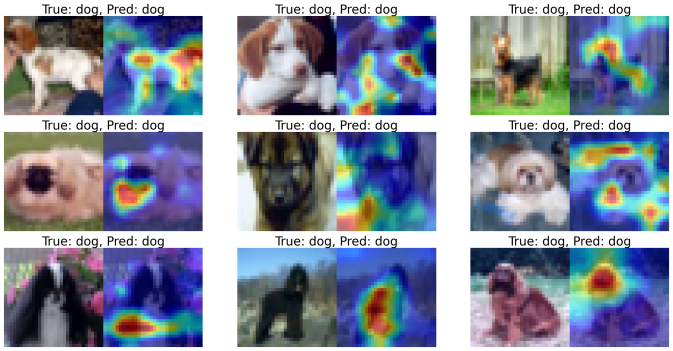


Fig. 3: Grad-CAM visualizations for the *dog* label across seleted test images.

More test images with heatmaps are shown in Figure 10.

C. XRAI

We used the XRAI method on the first 10 correctly predicted images of every class in the test set with the same reasoning as for *Grad-CAM*. We segmenting each image with Felzenswalb graph-based algorithm and more precisely the function `felzenszwalb` from the skimage python library [26]. We set the three inputs $scale$, σ and min_size to 30, 0.3 and 20 respectively after iteratively testing different combinations to find a good relation between number of segments and their size.
We calculated the attribution map of the input image using integrated gradients from the Captum library [27]. These were then averaged over each individual segment in the gain function. The segment with the highest gain was stored as a saliency mask that was used as input for the next iteration. For every iteration, a segment is added to the saliency mask until the whole image is covered. Every iteration was stored in the trajectory array T with the first element including only the highest scoring segment and the last element including the union of all segments. The unions were then passed through the model which predicted the class for each and it stopped when the classification was correct. This meant that the most recent image observed by the model, was the perturbed image with the least amount of segments that still got the correct classification.

For the labels *airplane* and *automobile* in Figure 4 and *bird* in Figure 5, the explanation focuses on the object or parts of the object in the image without necessary segments of the background.

Fig. 4: XRAI visualizations for the *airplane* and *automobile* labels across selected test images.

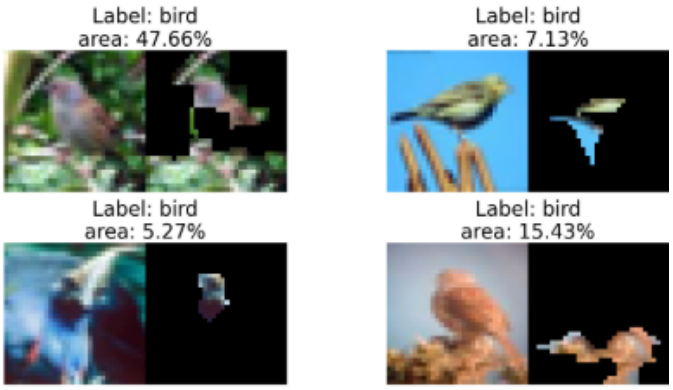


Fig. 5: XRAI visualizations for the *bird* label across selected test images.

The other labels have varying results. For example, the label *dog* in Figure 6 includes sometimes the face or the whole dog in the explanation. But for other images, a lot of the background is needed to predict it correctly. For the label *cat* in Figure 7, the model needs often only 1 or 2 segments. In the class *ship* in Figure 8, the explanation mostly includes the sea in order to make a correct prediction. All explanations can be shown in Figure 11.

Fig. 6: XRAI visualizations for the *dog* label across selected test images.

Fig. 7: XRAI visualizations for the *cat* label across selected test images.

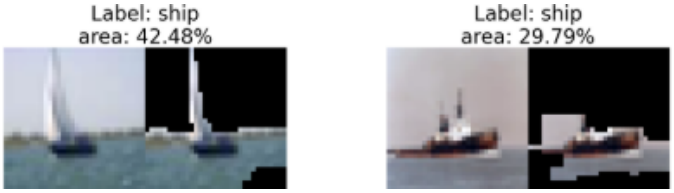


Fig. 8: XRAI visualizations for the *ship* label across selected test images.

V. Discussion

We now combine the individual results presented in the previous section to answer the research questions. First, we focus on the similarities of the findings using all three methods and discuss then in a second step individual differences between the methods.

A. What Image Parts Are Important for the Model?

With all our methods, we identified objects and image backgrounds to be important. These specific parts were dependent on the classes and sometimes we detected even differences between samples.

With *ICE*, we saw this with concepts that were present in

specific subparts of objects (e.g. truck front and trailer connection or the frog's tympanum) or the combination of object and background for Concept 4 stretching over the horse's belly and the green grass. Through *ICE* we saw that *airplane* and *ship* classes in which Concepts 5 and 9 were strongly present, the middle part of the image was always highlighted which could be an indication of a location-specific concept importance. However, to confirm this, further analysis is required.
With *Grad-CAM* we saw that for some labels like the *airplane*, the background seemed to be more important than the object. With other labels like the *automobile* or the *frog* the important areas focus almost solely on the object itself, even on the area where the concept was highlighted.
With *XRAI*, for labels like *airplane*, *automobile* and *bird* the object was important. For class *ship*, the background, including the sea, was often important.

B. What Patterns and Structures Has the Model Learned?

Using the different XAI techniques, we identified two patterns and structures that seem relevant for the model: colors and their transitions and class-specific structures.
For *ICE* this is shown in concepts looking at two different color patterns, see Figure 9a Concept 4 where the dark horse belly in combination with green grass was highlighted and Concept 5 showing the darker hull and lighter upper part of a ship. Regarding the class-specific structures, we identified the model's knowledge about windows of a car (Concept 2), the frog's tympanum (Concept 7) or the the deer's face with the space between its antler (Concept 8).

With *Grad-CAM* being a method for local explanations, broader patterns or concepts can't be determined from the heatmaps. Class-specific structures and patterns can be seen from the heatmaps. The same class-specific structures that were found with *ICE* can be seen with the *Grad-CAM* heatmaps. For example, the heatmaps focus around the frog's tympanum, car's windshield and between the deer's antlers in many test images. Some structures learned by the model can only be seen by *Grad-CAM*, such as the focus on the ship's chimney and helm on the *ship* label. With the *dog* label, we can see that the heatmaps focus often on the color transitions either patterns in the dog's fur, difference in color between the dog and a collar or an accessory or the difference between the foreground (dog) and the background.

The *XRAI* method shows that for the label *automobile*, the tires are important to predict the correct class. For the label *airplane*, the explanation shows the tip of the wings. The difference in color between the wing and the sky is, according to the explanation, an area the model looks at. Additionally, it seems like the model has learned the connection between the ocean and the ship for the *ship* label since the ocean is often present in the explanation.

C. What Limitations Does the Model Have?

The *dog* class stood out in our analysis. Generally, feature-based methods showed inconsistent heatmaps and segments indicating that the model looks at different regions for different samples. In addition, concepts could not represent the dog class well which is an indication that the model has not learned strong enough concepts compared to the others to represent this specific class. These struggles are also visible in the model accuracy for this specific class as it is the second lowest one. For other classes, the three applied methods returned mixed results.
ICE identified broader concepts for *cat* and *bird* classes which might be related to their lower test accuracies of (87.2% and 91.3%, respectively). Here when only looking at these concepts, one could suspect that the model has learned not a unifying feature that represents all samples of this class well. For higher accuracy classes, we detected these more well-defined concepts. Besides, when analyzing the concepts, we can see potential difficulties that could lead to misclassifications in the future: Concept 4 looking at the horses belly and the green grass might be too specific when the input only shows a specific part of the horse such as its face or the background changes. In addition, relying on the truck front and trailer connection in Concept 6 could be too specific for inputs that do not show a trailer. However, we must state that these are only reflections and future analysis needs to be done to verify these claims.
For some labels *Grad-CAM* provided mixed results with heatmap focus-points being either in the background or the object depending on the test image. These can be seen for *cat* and *bird* classes, which as stated before, have lower accuracies compared to other classes. It seems that the model has learned about the objects and the background on these classes, indicating that the model has not learned a single unifying feature.
When evaluating the class for cats, the results from the *XRAI* method were different from the other classes. The model often only needed one or two segments to correctly classify the images which meant that it was hard to interpret what features of the cat the model actually observes. This was probably an effect of the models structure and parameters. When a complete black image was fed through the model, the prediction was a cat. This could mean that when only one or two segments of a cat were shown, the model focuses on the black pixels around the segments. If this is the case, a higher threshold of pixels shown in the explanation could be a way of finding important attributes in images containing cats. There are also labels that has differences in explanation between images. For example, for the images with the *dog* label in Figure 6, the model was not able to correctly classify some images without the whole image showing.

D. In What Ways Can the XAI Techniques contribute to the Model's Improvement?

Finally, using all the information we have gathered in the discussion of the previous research questions, we are able to

provide an answer for the last one. We identified potential dataset limitations as the model struggles with the *dog* class. With this in mind, we would continue model training with a focus on this particular class and enhance the dataset with other dog images to ensure that the model is able to properly learn concepts for dogs and rely on one specific area when processing an input. For the other classes, the three methods did not show uniform results.

VI. Limitations

In this part, we critically reflect on our findings by addressing general limitations of the project and overlapping XAI method limitations. In addition, we dedicate subsections to reflect on the individual method's limitations. With this, we enable a proper calibration of the significance of our results.

A. General Limitations

In this project, we only used three different XAI methods which is not enough to provide a fully holistic approach into the explainability of this model. In addition, we only analyzed the results ourselves which led to subjective interpretations as humans have a confirmation bias where we pay attention to evidence confirming beliefs and tend to ignore information that challenges beliefs [28]. Moreover, our results might not provide a complete picture of the model as we did not analyse the 3 methods according to the *Co-12 Properties* [24] and cannot make any judgments regarding correctness or completeness of the explanations. Finally, we had to select specific model layers for our techniques which restricts the information we get from the model and we might have missed relevant explanations obtained through other layers.

B. Method-specific Limitations

1) Invertible Concept-based Explanations: While creating concepts using *ICE*, we noticed difficulties that limit the meaningfulness of the concepts. In our project, the requirement of the same dataset distribution for model training and creating the explanations [5] was fulfilled as we had access to the training dataset to extract the concepts. However, this is not a given in real-life applications. Besides, we needed to fix in addition to the layer hyperparameter a second one: the number of concepts as the correct number is unknown [14]. In this work, we focused on sparse concepts for human-understandability [24] and used the cosine similarity to achieve this goal. However, this limiting number of concepts might be too restricting to provide a complete understanding of the model. As we have seen in the TCAV scores, concepts are very general and prototypical images might not be enough to understand their full scope. Also, we need to address that concept calculation is sensitive to the data batch used and we find variation in the concepts due to the used batch. We tried to mitigate this by using a larger batch size (512) making concept extraction more stable. Unsupervised concepts introduce the problem of lacking any labeling and have therefore no semantic meaning [9]. This makes analysis difficult: We have seen differing results for prototypical concept images and TCAV scores for the corresponding concepts.

2) Grad-CAM: Due to the model structure and the small resolution of test images (32x32), the last layers proved to be unusable. The input is downsampled to 1x1 resolution by the end of the layers. This meant we had to choose layers more in the middle of the model (layers 22 and 29). Still, the resolution on these layers is 8x8 and 4x4 which do not convey much information about the image. Since the resolutions of the heatmaps and the test images are so small, it might be difficult for the heatmaps to focus on relevant areas. With larger resolutions, the problem could be mitigated. Also, the selection of the explanation layers is done almost completely by trial-and-error, trying to see which layers provided the most useful heatmaps. This is a common problem with *Grad-CAM* as there is no unique solution to find the optimal layer [7].

As stated before, the interpretation of the methods is subjective. With *Grad-CAM* being applied to only 10 images per label it does not give a comprehensive set of test images to base our conclusions on. More objective methods and metrics have been introduced by Chattopadhay et al. [29]. Where they calculate the confidence delta % when either relevant or irrelevant areas of the original image are perturbed. This way we could also analyze the faithfulness of the explanation method. If perturbing the relevant areas of test images causes the model's confidence to drop, we could determine that the heatmaps focus mainly on actually important areas. On the other hand, if we see a similar confidence drop when irrelevant areas are perturbed, we could determine that the heatmaps don't portray what the model has learned very well.

3) XRAI: The Cifar10 dataset consists of 32x32 images which are a lot smaller and has lower resolution than datasets that *XRAI* is often used on [8]. This can be a factor when segmenting the images because the segments become small and the contrast between them can be very hard to account for. This might lead to the model not focusing on particular segments when predicting an image and therefore it might need all segments present to classify the image correctly.

One big limitation of the *XRAI* method is also the time-consuming aspect of the algorithm. The method iterates through the segments one by one and adds one segment every loop. We use relatively small images in this project but the run time would drastically increase for larger images.

VII. Conclusion

This project aimed at providing a holistic approach to model explainability by applying three XAI methods (*Invertible Concept-based Explanations, Grad-CAM*, and *XRAI*) on a VGG16 model, pre-trained on the Cifar10 dataset.
We found that some parts of input features were more important for the model than others which varied for different classes. In addition, we were able to identify learned patters and structures relating to colors or object-specific features such as a frog's tympanum. We discovered the model's difficulties with the *dog* class and are able to improve the model by continuing training with a focus on this particular class to ensure a proper learning of important features and concepts.
Only through the combination of different methods, we were

able to find unifying and complementary explanations which gave insights into the model from three different perspectives, thus providing a more in-depth analysis.

REFERENCES

[1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, Aug. 2018. [Online]. Available: https://doi.org/10.1145/3236009

[2] A. Ghorbani, J. Wexler, J. Zou, and B. Kim, *Towards automatic concept-based explanations*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[3] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)," 2018.

[4] C. Yeh, B. Kim, S. Ö. Arik, C. Li, T. Pfister, and P. Ravikumar, "On Completeness-aware Concept-Based Explanations in Deep Neural Networks," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/ecb287ff763c169694f682af52c1f309-Abstract.html

[5] V. V. Ramaswamy, S. S. Y. Kim, R. Fong, and O. Russakovsky, "Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 10 932–10 941. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.01052

[6] R. Zhang, P. Madumal, T. Miller, K. A. Ehinger, and B. I. P. Rubinstein, "Invertible Concept-based Explanations for CNN Models with Non-negative Concept Activation Vectors," 2021.

[7] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016. [Online]. Available: http://arxiv.org/abs/1610.02391

[8] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry, "Xrai: Better attributions through regions," 2019, accessed: 2025-04-09. [Online]. Available: https://arxiv.org/abs/1906.02825

[9] F. Foscarin, K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Concept-Based Techniques for "Musicologist-Friendly" Explanations in Deep Music Classifiers," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022, pp. 876–883.

[10] K. Duvvuri, S. Chethana, S. S. Charan, V. Srihitha, T. K. Ramesh, and S. K S, "Grad-cam for visualizing diabetic retinopathy," in *2022 3rd International Conference for Emerging Technology (INCET)*, 2022, pp. 1–4.

[11] C.-J. Chen, L.-W. Chen, C.-H. Yang, Y.-Y. Huang, and Y.-M. Huang, "Improving cnn-based pest recognition with a post-hoc explanation of xai," 08 2021.

[12] R. Y. G. Paccotacya-Yanque, A. Bissoto, and S. Avila, "Are explanations helpful? a comparative analysis of explainability methods in skin lesion classifiers," in *2024 20th International Symposium on Medical Information Processing and Analysis (SIPAIM)*, 2024, pp. 1–5.

[13] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, p. 788–791, 1999.

[14] J. Maisog, A. DeMarco, K. Devarajan, S. Young, P. Fogel, and G. Luta, "Assessing methods for evaluating the number of components in non-negative matrix factorization," *Mathematics*, vol. 9, p. 2840, 11 2021. [Online]. Available: https://doi.org/10.3390/math9222840

[15] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 8730–8738. [Online]. Available: http://arxiv.org/abs/1801.03454

[16] D. Afchar, R. Hennequin, and V. Guigue, "Learning unsupervised hierarchies of audio concepts," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, P. Rao, H. A. Murthy, A. Srinivasamurthy, R. M. Bittner, R. C. Repetto, M. Goto, X. Serra, and M. Miron, Eds., 2022, pp. 427–436. [Online]. Available: https://archives.ismir.net/ismir2022/paper/000051.pdf

[17] G. Mikriukov, G. Schwalbe, F. Motzkus, and K. Bade, "Unveiling the anatomy of adversarial attacks: Concept-based xai dissection of cnns," in *Explainable Artificial Intelligence*, L. Longo, S. Lapuschkin, and C. Seifert, Eds. Cham: Springer Nature Switzerland, 2024, pp. 92–116.

[18] A. Nicolson, L. Schut, J. A. Noble, and Y. Gal, "Explaining explainability: Understanding concept activation vectors," *CoRR*, vol. abs/2404.03713, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2404.03713

[19] M. Sundararajan, S. Xu, A. Taly, R. Sayres, and A. Najmi, "Exploring principled visualizations for deep network attributions," in *Proceedings of an Unspecified Conference*, Los Angeles, 2019, p. 11, please update with exact conference or venue if known.

[20] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[21] T. Fel, A. M. Picard, L. Béthune, T. Boissin, D. Vigouroux, J. Colin, R. Cadène, and T. Serre, "CRAFT: concept recursive activation factorization for explainability," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 2711–2721.

[22] S. Sattarzadeh, M. Sudhakar, A. Lem, S. Mehryar, K. N. Plataniotis, J. Jang, H. Kim, Y. Jeong, S. Lee, and K. Bae, "Explaining convolutional neural networks through attribution-based input sampling and block-wise feature aggregation," *CoRR*, vol. abs/2010.00672, 2020. [Online]. Available: https://arxiv.org/abs/2010.00672

[23] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017, https://distill.pub/2017/feature-visualization.

[24] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert, "From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI," *CoRR*, vol. abs/2201.08164, 2022.

[25] J. R. BOLT and R. E. LOMBARD, "Evolution of the amphibian tympanic ear and the origin of frogs," *Biological Journal of the Linnean Society*, vol. 24, no. 1, pp. 83–99, 01 2008.

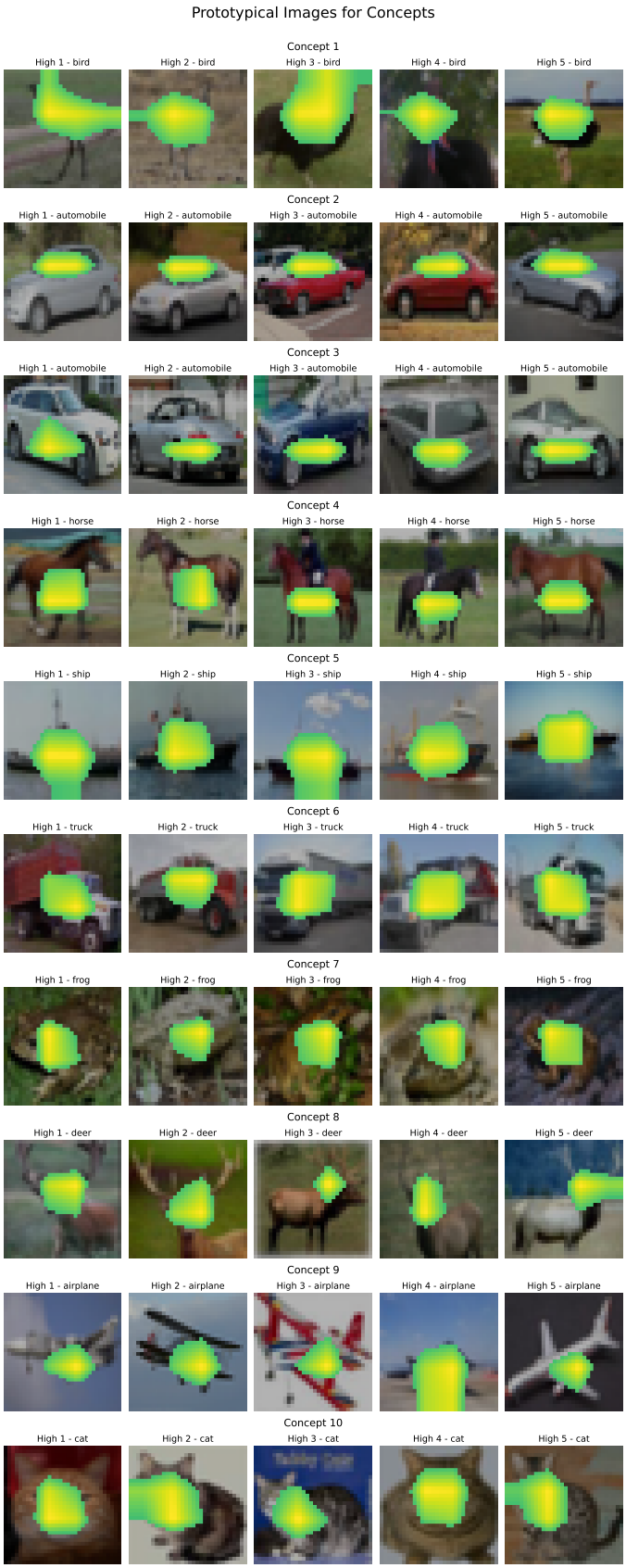
[26] scikit-image contributors, "skimage.segmentation — skimage 0.25.2 documentation," https://scikit-image.org/docs/stable/api/skimage.segmentation.html#skimage.segmentation.felzenszwalb, 2016.

[27] Captum contributors, "Integrated gradients · captum," https://captum.ai/docs/extension/integrated_gradients, 2025, accessed: 2025-04-09.

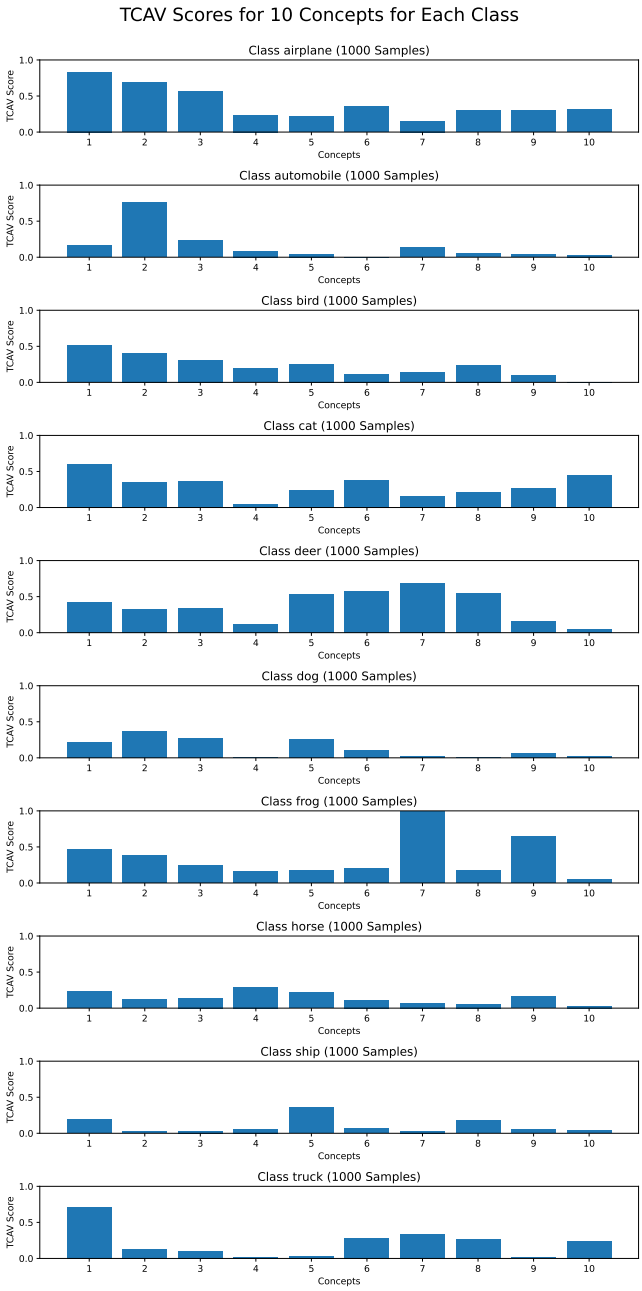
[28] R. S. Nickerson, "Confirmation bias: A ubiquitous phenomenon in many guises," *Review of General Psychology*, vol. 2, no. 2, pp. 175–220, 1998.

[29] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2018. [Online]. Available: http://dx.doi.org/10.1109/WACV.2018.00097

APPENDIX



(a) Visualization of the top 5 images with the highest average concept presence for all 10 concepts.



(b) TCAV scores for each concept applied to all test set samples of all classes.

Fig. 9: *ICE* concept visualizations and TCAV scores for 10 concepts extracted from the 29th model layer.

Fig. 10: Visualization of the first 10 correctly labeled images from the test set of each label with *Grad-CAM* heatmap applied.

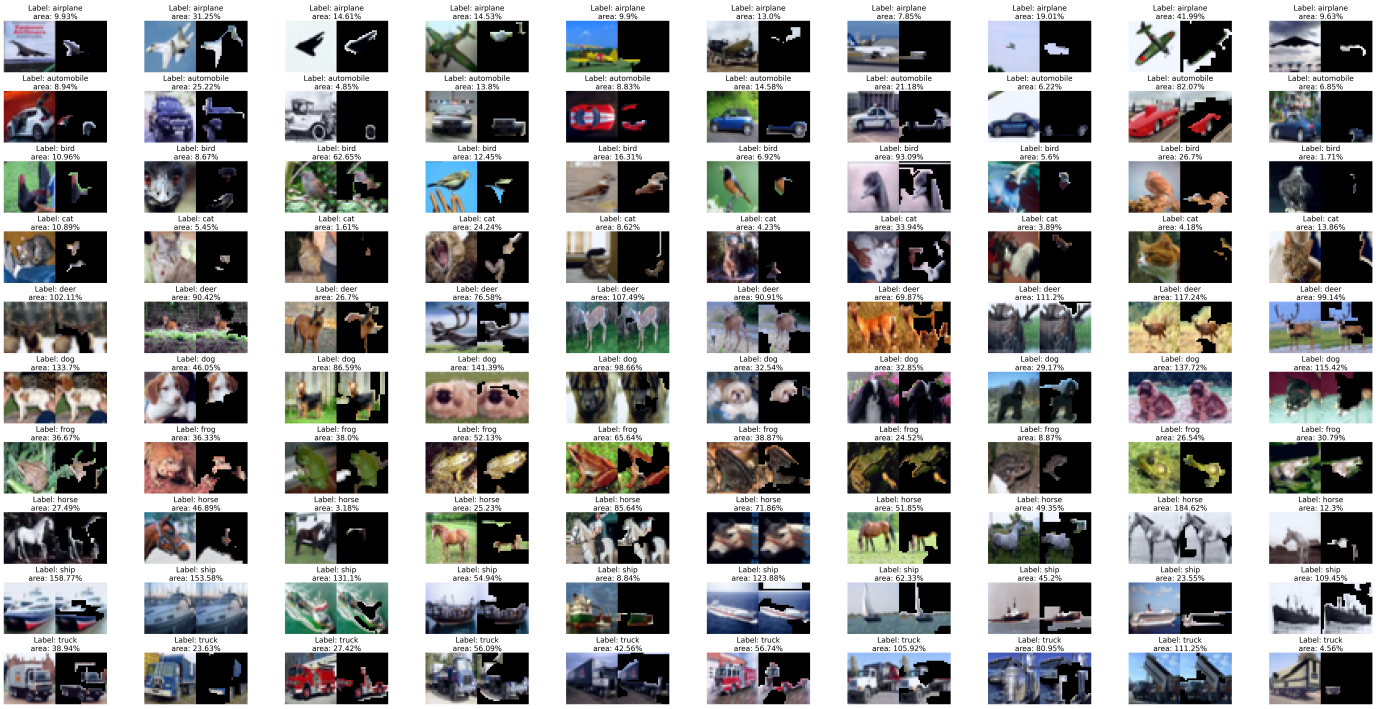


Fig. 11: Visualization of the first 10 correctly labeled images from the test set of each label with *XRAI* applied.