This Looks Like That, Because ... Explaining Prototypes for Interpretable Image Recognition

Meike Nauta
m.nauta@utwente.nl
University of Twente
Enschede, The Netherlands

Annemarie Jutte*
Jesper Provoost*
a.m.p.jutte@student.utwente.nl
j.c.provoost@student.utwente.nl
University of Twente
Enschede, The Netherlands

Christin Seifert
c.seifert@utwente.nl
University of Twente
Enschede, The Netherlands

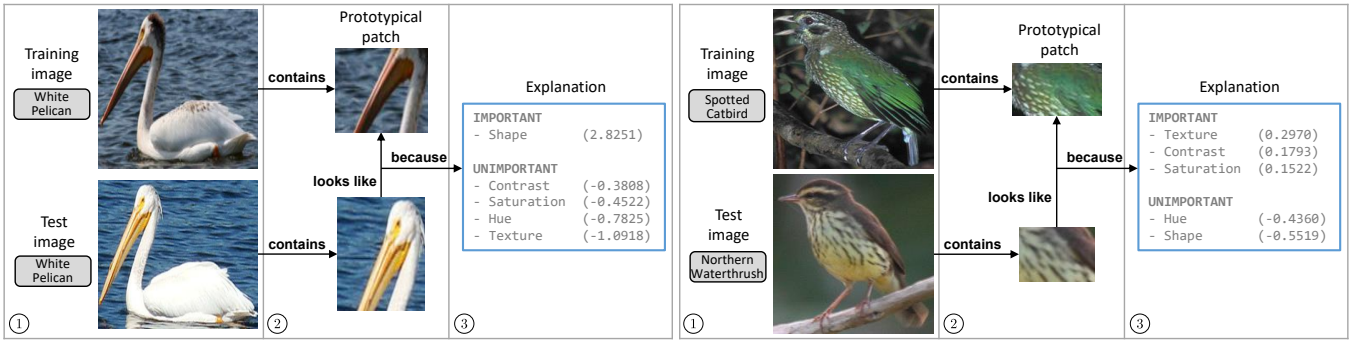


Figure 1: Overview of explaining prototypical learning for image recognition. (1) Data set with bird species for image recognition. (2) Image classification based on similarity with prototypical patches obtained by ProtoPNet [11]. (3) Our contribution: Explaining why the classification model considered the test image and the prototype to be similar. Local importance scores quantify the importance of visual characteristics.

ABSTRACT

Image recognition with prototypes is considered an interpretable alternative for black box deep learning models. Classification depends on the extent to which a test image "looks like" a prototype. However, perceptual similarity for humans can be different from the similarity learnt by the model. A user is unaware of the underlying classification strategy and does not know which image characteristics (e.g., color or shape) is the dominant characteristic for the decision. We address this ambiguity and argue that prototypes should be explained. Only visualizing prototypes can be insufficient for understanding what a prototype exactly represents, and why a prototype and an image are considered similar. We improve interpretability by automatically enhancing prototypes with extra information about visual characteristics considered important by the model. Specifically, our method quantifies the influence of color hue, shape, texture, contrast and saturation in a prototype. We apply our method to the existing Prototypical Part Network (ProtoPNet) and show that our explanations clarify the meaning of a prototype which might have been interpreted incorrectly otherwise. We also reveal that visually similar prototypes can have the same explanations, indicating redundancy. Because of the generality of our approach, it can improve the interpretability of any similarity-based method for prototypical image recognition.

1 INTRODUCTION

Convolutional Neural Networks (CNNs) [26] are the de-facto standard for object detection because of their impressive performance in numerous automated image classification tasks [19, 24, 41]. However, the black box nature of neural networks prevents a human to assess the model's decision making process, which is especially problematic in domains with high stakes decisions [37]. Following this demand on understanding automated decision making, explainable Artificial Intelligence (XAI) has been actively researched [1, 3, 18]. *Post-hoc* explanation methods learn a second, transparent model to approximate the first black box model [18], but these reverse-engineering approaches are not guaranteed to be faithful to the original model and might not show the *actual* reasoning of the black box model [37]. *Intrinsically* interpretable models on the other hand, are faithful by design and allow simulatability: a user should be able to reproduce the model's decision making process based on the input data together with the explanations of the interpretable model and come to the same prediction [9, 28]. One type of such models is prototypical learning [5], which has a transparent, built-in case-based decision making process. We focus on the problem of supervised image recognition where a machine learning model should assign a discrete label to an image. Prototypes in this context are usually 'nearest neighbours', i.e., images from the training set that look similar to the image being classified [2, 6, 27, 32]. The similarity between a prototype and an image is often measured in latent space, learned by the neural network, where images from

*Both authors contributed equally to this work.

the same class are close and dissimilar images are far apart with respect to a certain distance or similarity metric. Recently, the Prototypical Part Network (ProtoPNet) [11] was introduced which uses prototypical *parts* and identifies similar patches in an image. The classification depends on the extent to which *this* part of the image "looks like" *that* prototypical part. An example of this reasoning is shown in Fig. 1 (see Sect. 2 for a more detailed discussion on ProtoPNet).

Prototype Ambiguity In this paper, we address the ambiguity that prototypes can have and present a method to *explain prototypes*. Consider the left part in Figure 1, showing a prototypical patch ('prototype') of a white pelican. The similarity between this prototype and the patch in the test image is obvious. But what does this prototype exactly represent? Is the prototype looking for a white neck, an orange-colored beak or a specific shaped beak? Explanations however are especially needed when similarity is not so obvious. When seeing the two patches in the right part of Figure 1, a human might argue that these patches are dissimilar because of the colour differences. The classification model however considers these patches similar, even though the test image is from a different class than the prototype. Only visualizing prototypes can therefore be insufficient for understanding what a prototype exactly represents, and why a prototype and an image are considered similar.

It has been shown that CNNs trained on ImageNet are strongly biased towards recognizing texture [16], although other work shows that CNNs can be biased towards shape [33] or colour [20]. The classification strategy of a neural network therefore determines for what reason it considers a prototype and an image to be similar. Perceptual similarity for humans however is biased towards shape [25, 31], but also based on e.g. colour, size, semantic similarity, culture and complexity [22, 36, 39]. Rosenfeld et al. actually found that neural networks fall short on predicting human similarity perception [35]. Moreover, where humans are tolerant of moderate changes in object position, size, pose, contrast, illumination and clutter [14], deep image classification models might not be robust for such transformations [40, 44]. Since a user is not aware of the underlying classification strategy of the trained CNN and its potential biases, correct simulatability of the interpretable model is not guaranteed. A human and the CNN might have different reasoning processes, despite using the same prototypes. This issue may also arise with other explainability methods that show or highlight part of an image, such as attention mechanisms [10, 15, 47], components [38] and other part-based models e.g. [46, 48, 49].

Contribution We improve the interpretability of a predictive model (e.g. a CNN) by automatically enhancing prototypes with extra information about visual characteristics used by the model. Specifically, we present a methodology to quantify the influence of color hue, saturation, shape, texture, and contrast in a prototype. This helps a user to understand what the model pays attention to and why a model considers two images to be similar. While our method can extend any prototype-based model for image recognition, we show its applicability for the prototypical parts of ProtoPNet [11]. For example, again considering the left part of Figure 1, our explanation shows that color is not important, and that ProtoPNet considers these two patches to be similar because of the similar shape of the beak in the test image. Our method is especially useful when similarity is not so obvious. It can explain potentially misleading prototypes such as the right prototype in Fig. 1. Whereas a human might look for something green, our explanation reveals that ProtoPNet considers these two patches similar because of texture, contrast and saturation. The similarity is thus because of the dotted pattern and color hue was not important.

Our method automatically modifies images to change their hue, shape, texture, contrast or saturation, after which the similarity between the prototype and the original image is compared with the similarity of the prototype and a modified image. The intuition is that a visual characteristic is considered *unimportant* by the model when these two similarity scores are similar, and is deemed *important* when the similarity scores differ sufficiently. For example, a blue bird is changed to a yellow bird by changing the hue of the image. If hue would have been important for the specific prototype, it would be expected that the similarity between the prototype and the yellow bird will be low, whereas the similarity with the blue bird was high. The prototypes can subsequently be explained by annotating which visual characteristics were important and which not. As shown in Figure 1, the importance of each visual characteristic can be quantified and included in the explanation.

Our code is available via https://github.com/M-Nauta/Explaining_Prototypes.

The following section summarizes the ProtoPNet model [11]. Section 3 presents our methodology for quantitatively explaining prototypes w.r.t. visual image characteristics. The experimental setup is described in Section 4, after which its results are discussed in Section 5.

2 PROTOTYPICAL PART NETWORK

The methodology presented in this paper is applied to ProtoPNet, the Prototypical Part Network from Chen et al. [11] that follows the "*this* looks like *that*" reasoning. Prototypical parts learned by ProtoPNet are subsequently explained by our method. Key for presenting our explanation methodology is having a global understanding of the workings of ProtoPNet. We refer the reader to the original work by Chen et al. [11] for more specific details on ProtoPNet's training and visualization process.

The ProtoPNet architecture consists of a regular CNN, followed by a prototype layer and a fully-connected layer. The prototype layer consists of a pre-determined number of class-specific prototypes. The implementation of Chen et al. [11] learns 10 prototypes per class. The fully-connected layer learns a weight for each prototype. During training, prototypes are vectors in latent space that should learn discriminative, prototypical parts of a class. An input image is forwarded through the CNN, after which the prototype layer compares the resulting latent embedding with the prototype. A kernel slides over the latent image and at each location, the squared Euclidean distance between the latent prototype vector and a patch in the latent image is calculated. This creates an activation map, containing the distance to the prototype at each location in the latent image. To ensure that the prototype can be visualized, the training procedure of ProtoPNet requires that each prototype is *identical* to some latent training patch. The model loops through all training images and selects the image with the smallest distance to the latent prototype. The corresponding activation map can be

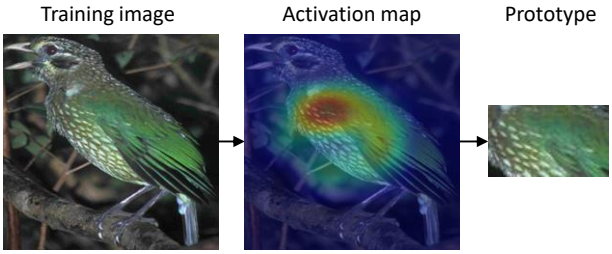


Figure 2: A prototype learned by ProtoPNet is the nearest patch of a training image.

upsampled to the size of the original image and visualized as a heatmap (see Figure 2). The prototype can now be visualized as the most similar patch of the training image in input space.

After training, ProtoPNet calculates the similarity between a prototype and a test image k. The distance $d_{j,k}$ between the nearest patch in latent image k to the j-th prototype is converted to a similarity score:

$$g_{j,k} = \log\left(\frac{d_{j,k}+1}{d_{j,k}+\epsilon}\right), \tag{1}$$

where ϵ is an arbitrarily small positive quantity to prevent zero division. To classify this image, the weighted similarity scores of the image and each prototype are summed per class, resulting in a final score for an image belonging to each class. Applying a softmax yields the predicted probability that a given image belongs to a certain class. The left part of Figure 4 shows an illustration of this reasoning process.

3 METHODOLOGY

In order to obtain importance scores for visual characteristics of prototypes, images are automatically modified. The characteristics in focus are contrast, color hue and saturation, shape, and texture (cf. Section 3.1). Our approach for automatically modifying these characteristics is described in Section 3.2. Section 3.3 presents a methodology to explain prototypes by quantifying the importance of a visual characteristic.

3.1 Important Visual Characteristics

The perceptual and cognitive processing in the human visual system is influenced by various features. To determine which image modifications we need to effectively explain prototypes, we discuss important visual characteristics for the human perceptual system from literature.

The data visualization domain has a ranking of channels to control the appearance of so-called marks [30]. A 'mark' is a basic graphical element in an image, such as a line, triangle or cube. Important visual channels for marks are position, size, angle, spatial region, color hue, color luminance, color saturation, curvature, motion and shape [30]. For static 2-dimensional natural images, motion is not applicable and we consider curvature related to shape. Furthermore, it is not necessary to modify the size, position, angle or spatial region of objects in images, since CNNs with pooling, possibly combined with suitable data augmentation, are invariant to these characteristics [17, 42].

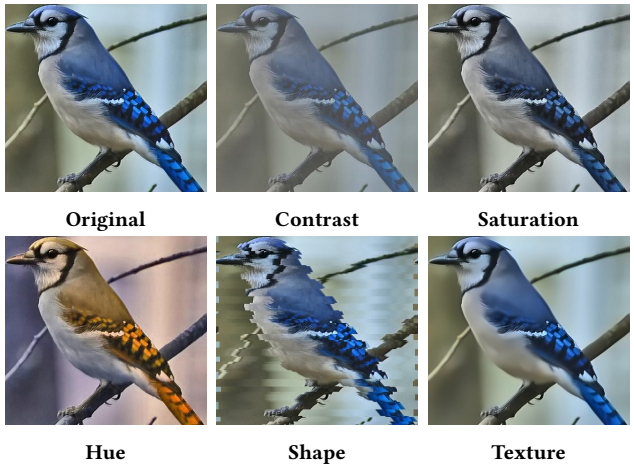


Figure 3: Image modifications for corresponding visual characteristics applied to an exemplar image.

Moreover, research in neuroscience shows that the human eye can recognize objects independent of ambient light level during the day [43], whereas contrast (spatial variation in luminance) is needed for edge detection and delineation of objects [30]. The human visual system is thus more sensitive to contrast than absolute luminance [43]. We therefore will not modify the absolute luminance, but the contrast in an image. Thus, the visual characteristics from the data visualization domain that we deem important for explaining a prototype are **hue**, **contrast**, **saturation** and **shape**.

Channels for marks in the visualization domain [30] are however too simplistic, because they do not include texture or material of an object. Research in neuroscience therefore also emphasizes the importance of texture for classifying objects in the natural world [8, 23]. Related to this, Bau et al. [4] disentangled visual representations by layers in a CNN and found that self-supervised models, especially in the earlier layers of the network, learn many texture detectors. We therefore also include **texture** as an important visual characteristic.

3.2 Image Modifications

For each of the visual characteristics, an image set is created. Each of these sets contains modified images, which are designed to be harder to classify based on the respective characteristic. For example, we generate a set of low-contrast images, such that contrast information can not (or hardly) be used by the model. Using these modified images, the importance of a characteristic for a specific prototype can be determined by comparing the differences between the prototype-image similarity of the original and modified image. The importance scores are described in Section 3.3.

To create the modified images, we apply image transformations to reduce the intensity of each characteristic, i.e., we create images with reduced contrast, saturation, hue, shape and texture. Figure 3 shows an example image and its modified versions. We opt for automated image modifications instead of manual modifications

used for experiments in psychology research (e.g. [34]) to be able to create a large number of modified images efficiently.

To create low **contrast** images, the original image is blended with the mean of its grayscale version. More concretely, we first create a grayscale version of the image and calculate its mean value. We then generate the modified image by pixel-wise averaging each channel (RGB) of the original image with the mean grayscale value.

Similarly, the low-**saturation** image is created by averaging the original image with its grayscale counterpart.

To generate an image with opposite **color hues**, the RGB image is converted to the HSV color space. We modify the H-dimension for each pixel, by adding 180 to the original image, which corresponds to the maximum shift in the H-dimension of the HSV color space.

In order to modify local **shapes** in an image, we distort the image as follows. We split the image in n_{bar} horizontal stripes of equal height. Each stripe is again split horizontally in two parts, the height of the split is chosen randomly. The upper part of the bar is shifted to either left or right (alternating between subsequent bars), the lower part is not modified. The upper parts of the bars are shifted with a random amount of 5 to 10 pixels. The part of the bar that is shifted out of the image is pasted on the other side of the image. We experimented with the shape distortion, and found that the above solution does distort local shapes, while preserving the global shape appearance. For our image size, we set $n_{\text{bar}} = 25$ in our experiments.

To modify **texture**, we apply a non-local means denoising technique [7]. This method removes small quantities of noise, and can therefore be used to blur the sophisticated texture of a bird while preserving its overall shape.

3.3 Importance Scores for Image Characteristics

We evaluate the importance of visual characteristics by calculating a local and a global importance score. The **local** score measures the importance of a visual characteristics for a single image, and is therefore applied on previously unseen images, i.e., the test data set S_{test}. The **global** score measures the importance for one prototype of the classification model and is independent of a specific input image. The global score is obtained from the training data set S_{train}.

Let $i \in \{\text{contrast}, \text{saturation}, \text{hue}, \text{shape}, \text{texture}\}$ denote the type of modification, $j \in \{1, 2, ..., n\}$ the prototype index and k the image. Furthermore, as introduced in Section 2, let the similarity of the original image and the prototype be denoted as g, and the similarity of the modified image and the prototype be denoted as $\hat{g}$. Then the local importance score $\phi^{i,j,k}_{\text{local}}$ of characteristic i for test image $k \in S_{\text{test}}$ on the j-th prototype is given by:

$$\phi^{i,j,k}_{\text{local}} = g_{j,k} - \hat{g}_{i,j,k}. \tag{2}$$

The global importance score of characteristic i on the jth prototype is the average of the local importance scores for all training images in S_{train}:

$$\phi^{i,j}_{\text{global}} = \frac{1}{|S_{\text{train}}|} \sum_{k=1}^{|S_{\text{train}}|} \phi^{i,j,k}_{\text{local}}. \tag{3}$$

Visual characteristics are classified as *important* if $\phi > 0$ and *unimportant* if $\phi < 0$.

These importance scores can be used to create *global* explanations that explain a prototype, and *local* explanations that explain the similarity score between a given image and a prototype. The global explanation for the j-th prototype lists for each visual characteristic i whether it is considered important for the prototype. The explanation can be quantified by including the importance scores $\phi^{i,j}_{\text{global}}$ for each visual characteristic i. This explanation is thus input independent and can be created before applying the prototype model to unseen images.

The local explanation is of use during testing and explains a single prediction by showing which visual characteristics were important for the similarity score between the j-th prototype and a patch in the k-th image.

4 EXPERIMENTAL SETUP

To evaluate our method for explaining prototypes, we first train a ProtoPNet [11] that results in an interpretable predictive model with prototypical parts for fine-grained image recognition. We apply our method to the resulting prototypes for generating global and local explanations. Section 4.1 discusses the data set and corresponding data augmentation, after which Section 4.2 presents the details for training ProtoPNet and the hyperparameters for our image modifications. We'll present the design of our experiments to evaluate our explanations in Section 4.3.

4.1 Data Set

For our experiments, we use the Caltech-UCSD Birds dataset [45], a data set for bird species identification which was also used by Chen et al. [11] for training their ProtoPNet. It contains 200 different classes with approximately 60 images per class. The data set provides a train-test split, leading to S_{train} with 5994 images and S_{test} with 5794 images.

To train a ProtoPNet, we apply the same data processing techniques as described in the original work [11]. We cropped the images according to the bounding boxes provided with the data set and apply data augmentation on S_{train} as described in the ProtoPNet paper [12] (including rotation, distortion, shearing and horizontal flipping). All images are resized to 224×224.

4.2 Model Parameters

To train ProtoPNet, we use the code provided by the authors[1]. We opted for DenseNet-121 [21] as pre-trained network for the initial layers of ProtoPNet, as this was reported to be the best-performing network on the Caltech-UCSD data set [11]. The DenseNet-121 network has been pre-trained on ImageNet [13].[2]

When forwarding the resized images through DenseNet, the input image dimensions, $H_{\text{in}} = W_{\text{in}} = 224$ and $D_{\text{in}} = 3$, are transformed to the output dimensions $H = 7$, $W = 7$ and $D = 128$. Depth D is a hyperparameter in ProtoPNet determining the number of channels for the network output and the prototypes, and is set to 128 as in ProtoPNet [11]. As in the paper by Chen et al. [11] we use 10 prototypes per class, leading to 2,000 prototypes in total. All other

[1] https://github.com/cfchen-duke/ProtoPNet, accessed May, 2020

[2] We use the same methodology as ProtoPNet [11] in order to reproduce their results, although it is known that there is some overlap between Caltech-USCD and ImageNet.

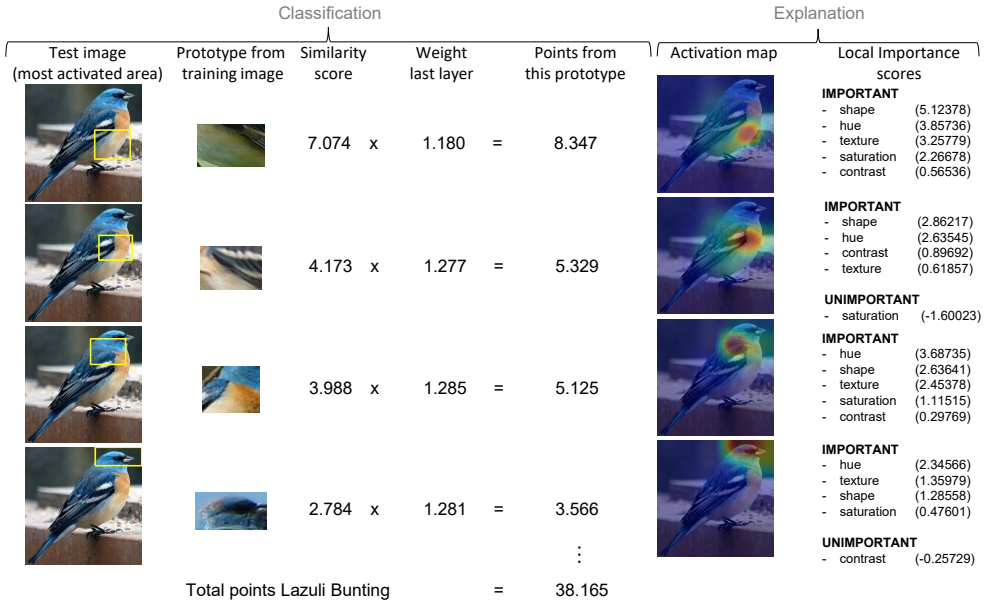


Figure 4: Left: ProtoPNet's reasoning with a subset of all prototypes of the Lazuli Bunting class. To classify a test image, ProtoPNet compares the class-specific prototypes of each class with the test image to calculate the total number of points for this class. An image is classified as the class with the most points. Right: The activation maps produced by ProtoPNet and our corresponding local explanations that explain which characteristics were important for a similarity score.

training parameters are also replicated from the implementation by Chen et al. [11].

For the color modifications (contrast, saturation and hue), we use PyTorch's image transformations[3]. More specifically, we use the `ColorJitter` function where we set the contrast, saturation or hue value to 0.5. The shape modification is manually implemented in Python. The texture modification is implemented with the Non-local Means Denoising algorithm [7] for colored images in OpenCV[4]. The filter strength of the denoising algorithm is set to 10 in order to maintain a balance between preserving shape and removing texture. The distribution of importance scores, and therefore the suitability of these hyperparameters, is evaluated in Section 5.3.

4.3 Experimental Design

Ideally, our explanations are validated by comparing it to some ground-truth. However, since we are opening up a "black box", this ground-truth is not available. We therefore qualitatively analyse a selection of local explanations (Section 5.1) and global explanations (Section 5.2). We especially analyse the effectiveness of our explanations for 'misleading' prototypes, where our explanations could be different from what a user might expect. Besides analysing examples, we also evaluate our importance scores across the complete training set. Section 5.3, analyses the distribution of global importance scores. These insights can also be used to validate the suitability of our image modifications. Lastly, we analyse the redundancy of prototypes in Section 5.4. We found that within a class, often multiple prototypes visualise the same area of the object. This raises the question whether these prototypes also focus on the same visual characteristics (i.e. redundancy) or that they complement each other.

[3]https://pytorch.org/docs/stable/torchvision/, accessed June 2020
[4]https://docs.opencv.org/3.4/d1/d79/group__photo__denoise.html#ga03aa4189fc3e31dafd638d90de335617, accessed June 2020

5 RESULTS AND DISCUSSION

The ProtoPNet is trained for 30 epochs, reaching a test accuracy of 77.42%. Because of the same data and training process as the original work [11], we do not know why our accuracy is lower than the accuracy reported in the original work (80.2%). We did notice that the network was overfitting on the training set which might explain the lower accuracy. However, the aim of this paper is not to train the best ProtoPNet, but to find a reasonable well-performing model such that we can explain its prototypes.

Figure 4 (left) shows a selection of prototypical patches ('prototypes') learned by ProtoPNet. ProtoPNet measures the similarity between a prototype and patches in a given test image. The resulting similarity scores are multiplied with learned weights to calculate a number of points per class. During training, a softmax is applied over the points per class to get a soft prediction. For testing, an image is classified as the class with the most points.

5.1 Analysing our Local Explanations

Figure 4 (right) shows how our local explanations complements the prototypical reasoning by explaining which visual characteristics were important for ProtoPNet's similarity score between a prototype and a specific image. Our local explanation shows the activation map as implemented by Chen et al. [11] and lists the

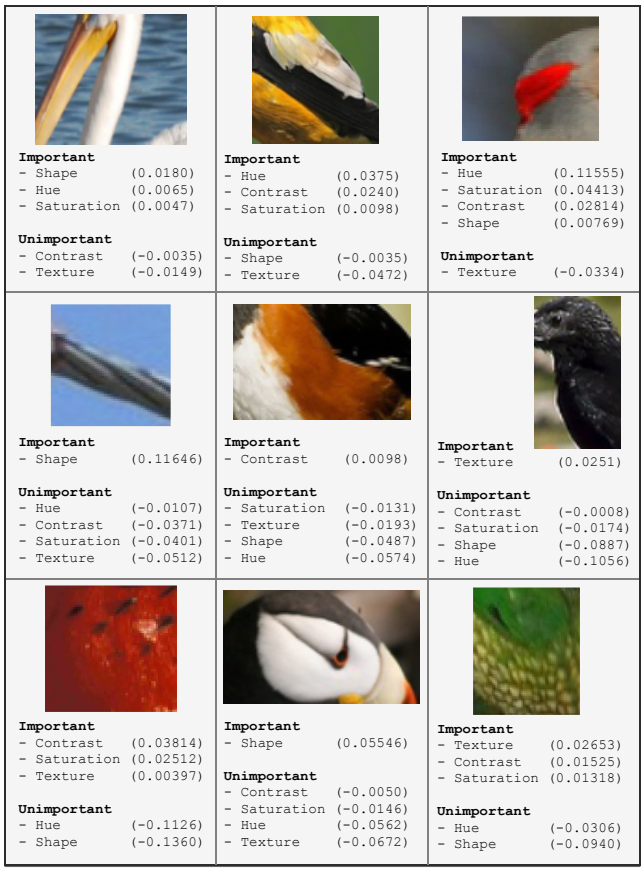


Figure 5: Selection of prototypes explained with their global importance scores. Top row: Prototypes with intuitive explanations. Center row: Prototypes for which only a single characteristic is important. Bottom row: potentially misleading prototypes (color hue might be expected to be, but is not important).

local importance score for each visual characteristic, divided between *important* and *unimportant*. With these extra insights, a user can understand why the model considers a patch in a test image and a prototype similar. This is especially useful when the given similarity score is in contrast with human perceptual similarity and an explanation is needed. It therefore serves as an extension to a prototypical model.

The importance of visual characteristics identified by our local explanations for the test image shown in Figure 4 seems reasonable given the typical blue color of the bird (hue), the contrast between the white belly and black feathers, and the shape of the white stripes on its feathers.

5.2 Analysing our Global Explanations

Since ProtoPNet has 2000 prototypes, every test image will have 2000 local explanations for each of the five characteristics. Local explanations can therefore be useful to explain unexpected results, but do not give a coherent, overall explanation of the model. Our

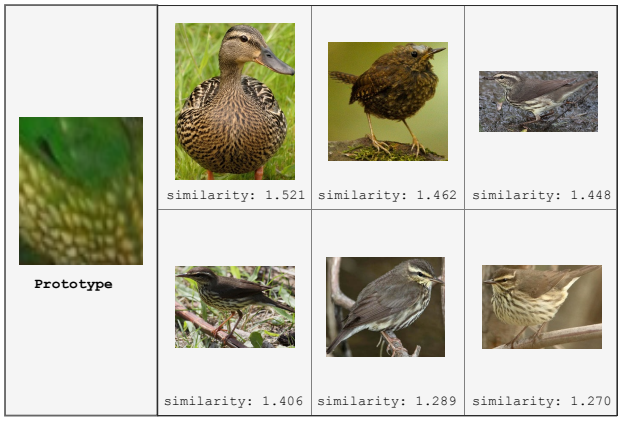


Figure 6: Images from the test set from a different class than the prototype-class which have the highest similarity scores with the prototype on the left. This example validates that texture is the most important characteristics (cf. Figure 5, bottom right).

methodology therefore also produces *global* explanations that give an average view regarding the importance scores for each prototype, as introduced in Section 3.3. These explanations can be generated *before* applying the model to a test image, since global explanations are independent of test input.

Figure 5 shows a selection of prototypes with their global explanation (additionally, Appendix A shows more examples). Our global explanation of the top middle image, shows that the color characteristics (hue, contrast and saturation) are important for this prototype, whereas shape and texture are not. In many cases, the importance of characteristics corresponds to the visually identifiable properties of the prototypes. However, while the explanations from the top row of Fig. 5 might seem intuitive, other explanations might come as a surprise. For example, our explanation shows that the middle left prototype is actually only activated by a specific shape, which would not be clear from the visualized prototype itself. In contrast, for the prototype in the center row only texture is found important. The bottom row of Fig. 5 shows prototypes that could be 'misleading'. Where a human might think that hue is important and look for something red (bottom left) or green (bottom right), our global explanation explains that color hue is actually not important. The right bottom prototype for example resembles something with dots, explained by the importance scores for texture, contrast and saturation. This explanations seems reasonable given that this prototype is a patch from the class "Spotted Catbird".

To validate our global explanations of these 'misleading' prototypes, we analyse test images that had a high similarity with the bottom right prototype from Fig 5. Although a prototype is trained to be class-specific, Figure 6 shows images from the test set that are from a different class but still have a high similarity score. From these images, it becomes clear that the bottom right prototype from Fig 5 is indeed not looking for something green, but instead is activated by a dotted pattern with a high contrast between the dots and their background. Our global explanation is thus in line with these results.

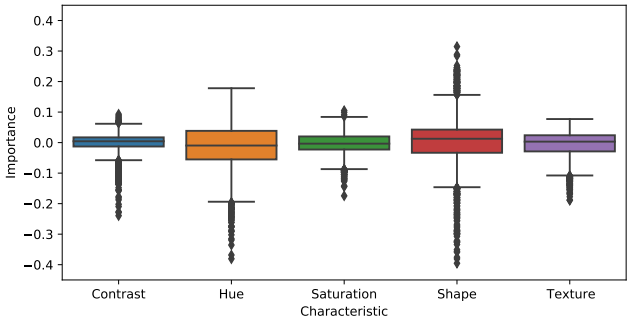


Figure 7: Box plot of global importance scores for each visual characteristic across the training set.

For a fully interpretable model, a human should be able to take the input data together with the explanations of the model to produce the same prediction as the model [28]. These examples show that without our explanations, a user would not be aware of the meaning of a given prototype and correct *simulatability* would not be guaranteed. Our global explanations can therefore clarify visual prototypes and improve the simulatability of a prototype-based model.

5.3 Distribution of Importance Scores

To get more insight as to how characteristics are used, we analyse how the global importance scores are distributed per characteristic. Fig. 7 shows a box plot for each visual characteristic describing the distribution of global importance scores of all prototypes across the complete training set. It can be seen that the median importance score is approximately equal to zero for all characteristics, showing that ProtoPNet uses all characteristics similarly. The fact that the importance scores of all characteristics are centered around zero also shows that our image modifications are of similar strength. For example, in case the contrast modification would change each image to a completely black image, contrast would always be considered important. A median importance score of zero therefore confirms a suitable hyperparameter choice for the image modifications.

The interquartile range in the boxplot indicates the variability of importance scores. Figure 7 shows that the variability of importance scores is small for contrast, saturation and texture, meaning that these characteristics usually only have a moderate influence on prototype similarity. In contrast, the high variability of importance scores for hue and shape means that hue and shape can be substantially (un)important for prototype similarity. This corresponds with the fact that hue and shape are considered more important and effective for humans in the data visualization domain than saturation or contrast [30].

5.4 Redundant Prototypes

A trained ProtoPNet has multiple prototypes per class, and we found that prototypes of the same class can visualize the same area of the bird, as shown in Fig. 8. Some visualized patches were even exactly identical. Such prototype redundancy was also discussed by [29], and was found to increase with the number of prototypes per class. While [29] presents as rule of thumb that the number of prototypes should be 2-3 times the number of classes, ProtoPNet [11] uses 10 prototypes per class. And although we analyse results from our trained instance of ProtoPNet, examples from the original authors [12] also include redundant prototypes.

Prototypes that are identical in latent space, will by definition lead to the same visualization and the same importance scores. From all prototypes from our trained ProtoPNet, 82 prototypes were identical to another prototype and are therefore redundant.

An interesting question, however, is whether prototypes that are slightly different in latent space (and therefore in their visualization), deem the same visual characteristics important. If their global importance scores are different, then these prototypes complement each other since they consider different characteristics important, even though they are perceptually similar.

Figure 8 shows prototypes from the same class focusing on a similar part of the bird. These examples indicate that prototypes that are visually similar, also have similar importance scores. Since global importance scores are often very similar, this indicates that ProtoPNet has more redundant prototypes, additional to the duplicates. However, the bottom row of Figure 8 shows that this is not always the case. Here, the global explanation shows that the left prototype focuses on different aspects than the middle and right prototype. Moreover, the explanations in Figure 8 again show that prototypes can be misleading. Whereas a human would look for something blue (3rd row) or blue-green (bottom row), our explanations show that hue is actually unimportant for both cases.

Besides showing examples, we evaluate the similarity of global importance scores for all prototypes. To prevent a subjective, manual assessment of whether two prototypes are showing a similar part of the bird, we consider prototypes to be visually similar when they are close (but not identical) to each other in the latent space learned by ProtoPNet. Calculating a correlation coefficient between latent prototypes and global importance scores would not be sufficient, since different prototypes can have similar explanations. For example, shape can be important for both the beak and the tail. To get an indication of redundancy, we therefore analyse whether visually similar prototypes have similar global importance scores.

We measure the Euclidean distance between the latent representation of two prototypes of the same class (a 'pair'). This gives $\binom{10}{2} = 45$ unique pairs per class, and $45 \cdot 200 = 9000$ pairs in total. We define P to be the set of unique pairs of two prototypes from the same class, such that $|P| = 9000$. Secondly, we can define $V \subset P$ as the set of pairs with two visually similar prototypes. We consider a pair of two prototypes i and j to be visually similar when the Euclidean distance in latent space $d_{i,j} < \tau$, where τ is a manual defined threshold. We found that $\tau = 0.15$ is a suitable threshold for perceptual similarity. This gives $|V| = 194$ unique pairs of 322 visually similar prototypes. To evaluate whether these visually similar prototypes also have similar global explanations, we measure the distance between explanations. We consider the global importance scores of a prototype as a vector of length 5 to calculate the Euclidean distance between the global explanations of two prototypes.

Table 1 shows the mean distance for all pairs in P, and the mean distance for only the visually similar pairs, V. It can be seen that

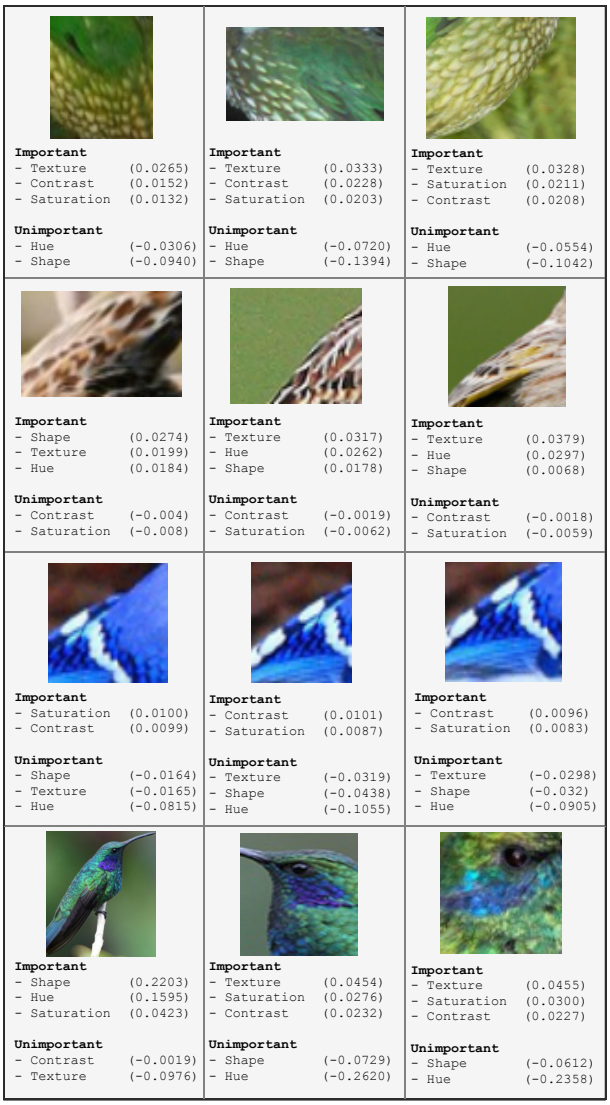


Figure 8: Prototypes annotated with their global importance scores. Prototypes that visualize a similar part of the bird often have similar importance scores.

two prototypes that are visually similar and therefore have a low distance in latent space, also have similar global importance scores. In addition, Figure 9 shows how the distance between the global explanations of two prototypes in a pair are distributed. The right plot shows that almost all pairs with visually similar prototypes have a small distance between their global importance scores, which is not the case in general (left plot). This confirms our findings from Figure 8 that visually similar prototypes often have similar explanations and thus focus on the same visual characteristics. With our methodology we therefore show the existence of redundant prototypes in our trained ProtoPNet, additional to the duplicates. From these results we can however also conclude that our method yields

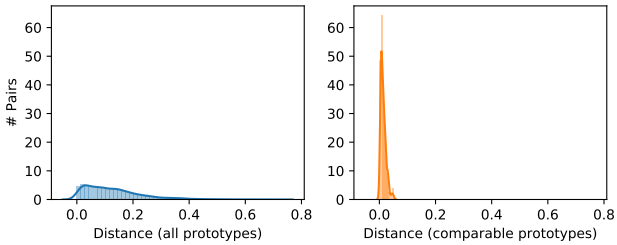


Figure 9: Histogram showing the distribution of Euclidean distances between the global explanations of two prototype of the same class. Left: all pairs $\in P$. Right: all pairs $\in V$, i.e. only visually similar prototypes.

	Euclidean Distance	
	Latent representation	Global importance scores
All prototypes (P)	1.442 ± 0.857	0.119 ± 0.026
Visually similar prototypes (V)	0.129 ± 0.103	0.013 ± 0.009

Table 1: The mean and standard deviation of the Euclidean distance between pairs of prototypes of the same class. The middle column shows the distance between two prototypes in latent space, whereas the right column shows the distance between the global importance scores of two prototypes.

consistent and robust global explanations between perceptually similar prototypes.

6 CONCLUSION AND FUTURE WORK

We addressed the ambiguity of prototype and argued why visual prototypes for image recognition should be explained. We presented an automated approach to explain visual prototypes learned by any prototypical image recognition model. Our method automatically modifies the hue, texture, shape, contrast or saturation of an image, and evaluates its similarity with a prototype. In this way, the important visual characteristics of a prototype can be identified. We applied our method to the prototypes learned by the Prototypical Part Network (ProtoPNet) [11]. The importance of visual characteristics identified by our explanations often corresponded to the visually perceptible properties of the prototypes, showing that our explanations are reasonable.

We also showed that perceptual similarity for humans can be different from the similarity learned by the model. For example, hue was not important for a prototype showing a patch of a green bird, although a human would probably put high emphasis on the green color when assessing similarity. Such 'misleading' prototypes will hinder correct simulatability and only visualizing prototypes can be insufficient for understanding why the model considered a prototype and an image highly similar. To the best of our knowledge, we are the first that address such ambiguity of prototypes. Lastly,

we found that perceptually similar prototypes from our trained ProtoPNet often focus on the same visual characteristics and therefore seem redundant. An interesting future experiment is to analyse how ProtoPNet performs when those redundant prototypes are removed.

A limitation of our method is the extra computation it requires: each image needs to be modified for each of the five visual characteristics, and this modified image should be forwarded through the prototypical model to compare similarity scores. However, we think the extra computational complexity is justifiable given the extra insights our method provides. Furthermore, because of the stand-alone nature of our method, it can be applied to any prototypical image recognition method. Our approach can also easily be extended with more visual characteristics or other image modifications a user is interested in.

Future work concerns the potential interactions between characteristics. Our importance scores currently assume that characteristics from image modifications are mutually exclusive. However, denoising the image to lower its texture could also slightly influence shape. We implemented the image modifications in such a way to limit interactions between characteristics as much as possible, but future analysis could determine to what extent visual characteristics are correlated.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[2] Sercan Ömer Arik and Tomas Pfister. 2019. Attention-Based Prototypical Learning Towards Interpretable, Confident and Robust Deep Neural Networks. *CoRR* abs/1902.06292 (2019). arXiv:1902.06292 http://arxiv.org/abs/1902.06292

[3] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Michael Biehl, Barbara Hammer, and Thomas Villmann. 2016. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science* 7, 2 (2016), 92–111.

[6] Jacob Bien and Robert Tibshirani. 2011. PROTOTYPE SELECTION FOR INTERPRETABLE CLASSIFICATION. *The Annals of Applied Statistics* 5, 4 (2011), 2403–2424. http://www.jstor.org/stable/23069335

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. 2011. Non-Local Means Denoising. *Image Processing On Line* 1 (2011), 208–212. https://doi.org/10.5201/ipol.2011.bcm_nlm

[8] C. Cavina-Pratesi, R.W. Kentridge, C.A. Heywood, and A.D. Milner. 2010. Separate Channels for Processing Form, Texture, and Color: Evidence from fMRI Adaptation and Visual Object Agnosia. *Cerebral Cortex* 20, 10 (01 2010), 2319–2332. https://doi.org/10.1093/cercor/bhp298 arXiv:https://academic.oup.com/cercor/article-pdf/20/10/2319/17302013/bhp298.pdf

[9] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. 2017. Interpretability of deep learning models: a survey of results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 1–6.

[10] Sneha Chaudhari, Gungor Polatkan, Rohan Ramanath, and Varun Mithal. 2019. An Attentive Survey of Attention Models. *CoRR* abs/1904.02874 (2019). arXiv:1904.02874 http://arxiv.org/abs/1904.02874

[11] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*. 8928–8939.

[12] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, Vol. Supplement S3,S9. 8928–8939.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 248–255.

[14] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. 2012. How does the brain solve visual object recognition? *Neuron* 73, 3 (2012), 415–434.

[15] Jianlong Fu, Heliang Zheng, and Tao Mei. 2017. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bygh9j09KX

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51, 5 (2018), 1–42.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[20] Hossein Hosseini, Baicen Xiao, Mayoore Jaiswal, and Radha Poovendran. 2018. Assessing Shape Bias Property of Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[22] Marcie L King, Iris IA Groen, Adam Steel, Dwight J Kravitz, and Chris I Baker. 2019. Similarity judgments and cortical visual responses reflect different properties of object and scene categories in naturalistic images. *NeuroImage* 197 (2019), 368–382.

[23] Zoe Kourtzi and Nancy Kanwisher. 2000. Cortical regions involved in perceiving object shape. *Journal of Neuroscience* 20, 9 (2000), 3310–3318.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[25] Barbara Landau, Linda B Smith, and Susan S Jones. 1988. The importance of shape in early lexical learning. *Cognitive development* 3, 3 (1988), 299–321.

[26] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation* 1, 4 (1989), 541–551.

[27] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep Learning for Case-Based Reasoning Through Prototypes: A Neural Network That Explains Its Predictions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 3530–3537. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17082

[28] Zachary C. Lipton. 2018. The Mythos of Model Interpretability. *Queue* 16, 3, Article 30 (June 2018), 27 pages. https://doi.org/10.1145/3236386.3241340

[29] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. 2019. Interpretable and Steerable Sequence Learning via Prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. Association for Computing Machinery, New York, NY, USA, 903–913. https://doi.org/10.1145/3292500.3330908

[30] Tamara Munzner and Eamonn Maguire. 2015. *Visualization analysis & design*. http://www.crcnetbase.com/isbn/9781466508934 OCLC: 897069361.

[31] Hans P. Op de Beeck, Katrien Torfs, and Johan Wagemans. 2008. Perceived Shape Similarity among Unfamiliar Objects and the Organization of the Human Object Vision Pathway. *Journal of Neuroscience* 28, 40 (2008), 10111–10123. https://doi.org/10.1523/JNEUROSCI.2511-08.2008 arXiv:https://www.jneurosci.org/content/28/40/10111.full.pdf

[32] F. Plesse, A. Ginsca, B. Delezoide, and F. Prêteux. 2018. Learning Prototypes for Visual Relationship Detection. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*. 1–6.

[33] Samuel Ritter, David G. T. Barrett, Adam Santoro, and Matt M. Botvinick. 2017. Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 2940–2949.

http://proceedings.mlr.press/v70/ritter17a.html
[34] Samuel Ritter, David G. T. Barrett, Adam Santoro, and Matt M. Botvinick. 2017. Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study *(Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, International Convention Centre, Sydney, Australia, 2940–2949. http://proceedings.mlr.press/v70/ritter17a.html
[35] Amir Rosenfeld, Markus D. Solbach, and John K. Tsotsos. 2018. Totally Looks Like - How Humans Compare, Compared to Machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
[36] Bruno Rossion and Gilles Pourtois. 2004. Revisiting Snodgrass and Vanderwart's Object Pictorial Set: The Role of Surface Detail in Basic-Level Object Recognition. *Perception* 33, 2 (2004), 217–236. https://doi.org/10.1068/p5117 arXiv:https://doi.org/10.1068/p5117 PMID: 15109163.
[37] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.
[38] Sascha Saralajew, Lars Holdijk, Maike Rees, Ebubekir Asan, and Thomas Villmann. 2019. Classification-by-Components: Probabilistic Modeling of Reasoning over a Set of Components. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 2792–2803. http://papers.nips.cc/paper/8546-classification-by-components-probabilistic-modeling-of-reasoning-over-a-set-of-components.pdf
[39] Marshall H Segall, Donald Thomas Campbell, and Melville Jean Herskovits. 1966. *The influence of culture on visual perception*. Bobbs-Merrill Indianapolis.
[40] Alex Serban, Erik Poll, and Joost Visser. 2020. Adversarial Examples on Object Recognition: A Comprehensive Survey. *ACM Comput. Surv.* 53, 3, Article 66 (June 2020), 38 pages. https://doi.org/10.1145/3398394
[41] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[42] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.1556
[43] Georg F Striedter. 2016. *Neurobiology: a functional approach*. Oxford University Press.
[44] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
[45] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. 2010. *Caltech-UCSD Birds 200*. Technical Report CNS-TR-2010-001. California Institute of Technology.
[46] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. 2014. Part-Based R-CNNs for Fine-Grained Category Detection. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 834–849.
[47] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. 2017. Learning Multi-Attention Convolutional Neural Network for Fine-Grained Image Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
[48] H. Zheng, J. Fu, Z. Zha, J. Luo, and T. Mei. 2020. Learning Rich Part Hierarchies With Progressive Attention Networks for Fine-Grained Image Recognition. *IEEE Transactions on Image Processing* 29 (2020), 476–488.
[49] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. 2019. Looking for the Devil in the Details: Learning Trilinear Attention Sampling Network for Fine-Grained Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

A EXEMPLAR GLOBAL IMPORTANCE SCORES

Figure 10 shows more prototypes explained with their global importance scores.

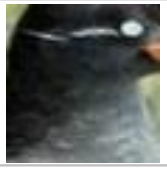
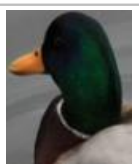
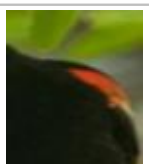
				
IMPORTANT - Shape (0.12612) - Hue (0.02251) UNIMPORTANT - Saturation (-0.00891) - Contrast (-0.00950) - Texture (-0.02859)	IMPORTANT - Shape (0.00535) - Contrast (0.00389) - Texture (0.00118) UNIMPORTANT - Saturation (-0.00164) - Hue (-0.03723)	IMPORTANT - Texture (0.00553) UNIMPORTANT - Shape (-0.02994) - Saturation (-0.06208) - Contrast (-0.07505) - Hue (-0.08469)	IMPORTANT - Shape (0.01462) UNIMPORTANT - Saturation (-0.01854) - Hue (-0.01919) - Contrast (-0.02048) - Texture (-0.09256)	IMPORTANT - Shape (0.25244) - Hue (0.11239) - Contrast (0.02250) UNIMPORTANT - Texture (-0.01268) - Saturation (-0.02115)
				
IMPORTANT - Texture (0.01643) UNIMPORTANT - Contrast (-0.00067) - Saturation (-0.01196) - Hue (-0.06697) - Shape (-0.08267)	IMPORTANT - Texture (0.01859) - Contrast (0.00055) UNIMPORTANT - Saturation (-0.01012) - Shape (-0.05742) - Hue (-0.06186)	IMPORTANT - Hue (0.09567) - Saturation (0.05292) - Contrast (0.02407) UNIMPORTANT - Shape (-0.00541) - Texture (-0.01439)	IMPORTANT - Shape (0.04607) - Contrast (0.01180) UNIMPORTANT - Saturation (-0.00041) - Texture (-0.01691) - Hue (-0.03502)	IMPORTANT - Saturation (0.05259) - Shape (0.04910) - Hue (0.03651) - Contrast (0.03263) UNIMPORTANT - Texture (-0.01779)
				
IMPORTANT - Shape (0.02832) - Texture (0.01200) UNIMPORTANT - Contrast (-0.00188) - Saturation (-0.00975) - Hue (-0.03142)	IMPORTANT - Texture (0.00481) - Contrast (0.00212) UNIMPORTANT - Hue (-0.00682) - Saturation (-0.00902) - Shape (-0.07529)	IMPORTANT - Hue (0.00666) UNIMPORTANT - Saturation (-0.01258) - Contrast (-0.04033) - Texture (-0.04318) - Shape (-0.07520)	IMPORTANT - Shape (0.03098) - Texture (0.01562) - Contrast (0.00163) UNIMPORTANT - Saturation (-0.00142) - Hue (-0.00314)	IMPORTANT - Saturation (0.05579) - Hue (0.05159) - Contrast (0.03007) - Shape (0.02568) UNIMPORTANT - Texture (-0.01181)
				
IMPORTANT - Hue (0.12581) - Saturation (0.05760) - Contrast (0.01156) - Shape (0.01004) - Texture (0.00903) UNIMPORTANT	IMPORTANT - Shape (0.06858) - Hue (0.02135) UNIMPORTANT - Saturation (-0.00302) - Texture (-0.00304) - Contrast (-0.01040)	IMPORTANT - Contrast (0.01790) - Texture (0.00795) - Saturation (0.00370) UNIMPORTANT - Shape (-0.08718) - Hue (-0.11308)	IMPORTANT - Hue (0.09339) - Saturation (0.04942) - Contrast (0.02737) - Shape (0.00208) UNIMPORTANT - Texture (-0.01586)	IMPORTANT - Saturation (0.02133) - Contrast (0.01990) - Texture (0.00985) UNIMPORTANT - Shape (-0.00116) - Hue (-0.03097)

Figure 10: Prototypes explained with their global importance scores.