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∗  
a.m.p.jutte@student.utwente.nl  
University of Twente  
Enschede, The Netherlands

Christin Seifert  
christin.seifert@uni-due.de  
University of Duisburg-Essen  
Essen, Germany

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 learned by the classification model. Hence, only visualising prototypes can be insufficient for a user to understand what a prototype exactly represents, and why the model considers a prototype and an image to be similar. We address this ambiguity and argue that prototypes should be explained. We improve interpretability by automatically enhancing visual prototypes with textual quantitative information about visual characteristics deemed important by the classification model. Specifically, our method clarifies the meaning of a prototype by quantifying the influence of colour hue, shape, texture, contrast and saturation and can generate both global and local explanations. Because of the generality of our approach, it can improve the interpretability of any similarity-based method for prototypical image recognition. In our experiments, we apply our method to the existing Prototypical Part Network (ProtoPNet). Our analysis confirms that the global explanations are generalisable, and often correspond to the visually perceptible properties of a prototype. Our explanations are especially relevant for prototypes which might have been interpreted incorrectly otherwise. By explaining such 'misleading' prototypes, we improve the interpretability and simulatability of a prototype-based classification model. We also use our method to check whether visually similar prototypes have similar explanations, and are able to discover redundancy. Code is available at https://github.com/M-Nauta/Explaining_Prototypes.

CCS CONCEPTS

• Computing methodologies → Computer vision; Machine learning; Image manipulation.

KEYWORDS

Prototypes, interpretability, explainable artificial intelligence

1 INTRODUCTION

Convolutional Neural Networks (CNNs) [24] are the de-facto standard for object detection due to their impressive performance in numerous automated image classification tasks [17, 22, 39]. 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 [36]. Following this demand on understanding automated decision making, explainable Artificial Intelligence (XAI) has been actively researched [1, 3, 16]. Post-hoc explanation methods learn a second, transparent model to approximate the first black box model [16], but these reverse-engineering approaches are not guaranteed to show the actual reasoning of the black box model [36]. 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, 26].
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 label 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, 25, 31]. The similarity between a prototype and an image is often measured in latent space, learned by the neural network, where images from 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] and ProtoTree [29] were introduced which use prototypical parts and identify similar patches in an image. The classification depends on the extent to which this part of the image “looks like” that prototypical part, measured by a similarity score. An example of this reasoning is shown in Fig. 1 (Sect. 2 discusses ProtoPNet in more detail).

**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. Although the similarity between this prototype and the patch in the test image is not surprising, it is unclear what this prototype exactly represents. Is the prototype looking for a white neck, an orange-coloured beak or is the shape of the beak specifically important? The similarity score between the two patches assigned by the model depends on its classification strategy, and hence its learned latent space. Explanations 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 assigns these patches a high similarity score, and thus considers them alike, even though the test image is from a different class than the prototype. This shows that a human and the CNN might have different reasoning processes, despite using the same prototypes. The classification strategy of a neural network, dependent on the learned latent space, determines the reason for considering two patches as being similar or different. It has been shown that CNNs trained on ImageNet are strongly biased towards recognizing texture [14], although other work shows that CNNs can be biased towards shape [32] or colour [18]. Perceptual similarity for humans however is biased towards shape [23, 30], but also based on e.g. colour, size, semantic similarity, culture and complexity [20, 35, 38]. It is also questionable whether humans and CNNs will ever follow the exact same similarity reasoning, since Rosenfeld et al. found that neural networks fall short on predicting human similarity perception [34]. Since a user is not aware of the underlying classification strategy of the trained CNN and might also be unaware of personal biases, only visualising prototypes is insufficient for understanding what a prototype exactly represents, and why a prototype and an image are considered similar. This issue may also arise with other explainability methods that show or highlight image parts, such as attention mechanisms [10, 13, 44], components [37] and other part-based models e.g. [43, 45, 46]. Including our explanations can help users to increase the simulatability [26] and general understanding of the model.

**Contribution** We improve the interpretability of a prototype-based CNN by automatically enhancing prototypes with extra textual quantitative information about visual characteristics used by the model. Specifically, we present a methodology to quantify the influence of colour hue, saturation, shape, texture, and contrast in a prototype. This clarifies what the model pays attention to and why a model considers two images to be similar. Hence, we disentangle localisation and explanation. Our method can extend any prototype-based model for image recognition, such as ProtoPNet [11] and ProtoTree [29]. In this paper, we show its applicability for the prototypical parts of ProtoPNet. For example, again considering the left part of Figure 1, our explanation shows that ProtoPNet considers the prototype and patch from the test image to be similar because of the similar colour hue and 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 shape. The similarity is thus because of the dotted pattern and colour hue was not important. This explanation seems reasonable given that the prototype is a patch from the class “Spotted Catbird”.

Our method automatically modifies images to change their hue, shape, texture, contrast or saturation. We forward both the original image and the modified image through ProtoPNet and analyse ProtoPNet’s similarity scores. Specifically, the similarity score of ProtoPNet between the prototype and the original image is compared with the similarity score of the prototype and a modified image. The intuition is that a visual characteristic is considered unimportant by the classification model when the difference between these two similarity scores is small (and will therefore get a low importance score), and is deemed important when the similarity scores differ sufficiently. For example, a blue bird is changed to a purple bird by changing the hue of the image. If hue would have been important for the specific prototype, it would be expected that ProtoPNet assigns a low similarity between the prototype and the purple bird, whereas the similarity with the blue bird was high. As shown in Figure 1, the prototypes can subsequently be explained by quantifying the importance of visual characteristics.

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 results are presented and discussed in Section 5.

## 2 PROTOTYPICAL PART NETWORK

We apply the methodology presented in this paper 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 to the original work by Chen et al. [11] for more specific details on ProtoPNet’s training and visualisation process.

The ProtoPNet architecture consists of a standard CNN (e.g. ResNet), 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 visualised, the training procedure of ProtoPNet requires that each prototype is identical to some latent training patch. ProtoPNet iterates through all training images and selects the image with the smallest distance to the latent prototype. The corresponding activation map can be upsampled to the size of the original image and visualised as a heatmap (see Figure 2). The prototype can now be visualised as the most similar patch in training images in input space.

After training, ProtoPNet classifies a test image \( k \) by calculating the similarity between a prototype and 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:

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

where \( \epsilon \) is an arbitrarily small positive quantity to prevent zero division. To classify this image, the similarity scores of the image and each prototype are weighted by the fully-connected layer and 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, colour 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.

The data visualisation domain has a ranking of channels to control the appearance of so-called marks [28]. A ‘mark’ is a basic graphical element in an image, such as a black triangle or moving red dot. Important visual channels for marks are position, size, angle, spatial region, colour hue, colour luminance, colour saturation, curvature, motion and shape [28]. 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 [15, 40]. Moreover, research in neuroscience shows that the human eye can recognise objects independent of ambient light level during the day [41], whereas contrast (spatial variation in luminance) is needed for edge detection and delineation of objects [28]. The human visual system is thus more sensitive to contrast than absolute luminance [41]. We therefore will not modify the absolute luminance, but the contrast in an image. Thus, the visual characteristics from the data visualisation domain that we deem important for explaining a prototype are hue, contrast, saturation and shape.

The channels for marks mentioned in the previous paragraph are however too simplistic, because they do not include the texture or material of an object. Research in neuroscience also emphasises the importance of texture for classifying objects in the natural world [8, 21]. 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
We evaluate the importance of visual characteristics by calculating 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.

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., [33]), 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 create an image with different colour hues, the RGB image is converted to the HSV colour space after which the H-dimension is modified for each pixel. In order to modify shapes in an image, we add a linear displacement by warping the image. Specifically, we shift pixels according to a sine wave in both the horizontal and vertical direction. 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. Implementation details are presented in Section 4.2.

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 the visual characteristics for a single image, and is therefore applied on previously unseen images, i.e., any image in the test dataset $S_{\text{test}}$. The global score measures the importance of visual characteristics for one prototype in general, and is independent of a specific input image. The global score is obtained from the training dataset $S_{\text{train}}$.

Let $i \in \{\text{contrast, saturation, hue, shape, texture}\}$ 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 $\tilde{g}$. Then the local importance score $\phi_{\text{local}}^{i,j,k}$ of characteristic $i$ for test image $k \in S_{\text{test}}$ on the $j$-th prototype is the difference in similarity scores:

$$\phi_{\text{local}}^{i,j,k} = g_{j,k} - \tilde{g}_{i,j,k}.$$  

For the calculation, we fix the patch location such that the part of image $k$ compared with prototype $j$ is the same for both the original and modified image.

Whereas the local importance score indicates to what extent a visual characteristic influences the similarity score given by the prototype-based model for a single image, the global importance score gives a general impression of the importance of visual characteristics in a prototype. The global score of characteristic $i$ on the $j$-th prototype can be calculated by taking all training images into account. A naive approach would be to average over the local scores of all training images. However, prototype $j$ might not be present in all images and modifying those images will therefore not (or barely) influence the resulting similarity score. For example, if prototype $j$ represents a specific beak which is not present in original image $k$, ProtoPNet will give a low similarity score $g_{j,k}$. Since this prototype will also be absent in the modified image, the difference between the similarity scores, Eq. 2, is near zero. This result could indicate that a certain characteristic is not important, although the result is actually indicating that the prototype was simply not present. Therefore, we create a more informative global importance score by calculating a weighted arithmetic mean by weighing the local scores of all images in $S_{\text{train}}$ by their similarity score with the prototype:

$$\phi_{\text{global}}^{i,j} = \frac{\sum_{k=1}^{|S_{\text{train}}|} \phi_{\text{local}}^{i,j,k} \cdot g_{j,k}}{\sum_{k=1}^{|S_{\text{train}}|} g_{j,k}}.$$  

Hence, if unmodified image $k$ gets a low similarity score with prototype $j$, it will get a low weight for the global importance calculation. In contrast, if prototype $j$ is clearly present in image $k$, ProtoPNet will assign a high similarity score and hence $k$ gets a high weight.

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 $i$ its importance by showing the importance scores $\phi_{\text{global}}^{i,j}$. 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.

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 dataset, corresponding data augmentation, and hyperparameters for training ProtoPNet. Section 4.2 presents the implementation details for our image modifications. The design of our experiments to evaluate our explanations is presented in Section 4.3.

4.1 Dataset and Parameters

For our experiments, we use the Caltech-UCSD Birds dataset [42], a dataset 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 dataset provides a train-test split, leading to $S_{\text{train}}$ with 5994 images and $S_{\text{test}}$ with 5794 images. For training ProtoPNet, we apply the same data processing techniques as the original work [11]. We cropped the images according to the bounding boxes provided with the dataset and apply data augmentation on $S_{\text{train}}$ as described in ProtoPNet’s supplementary material [11]. All images are resized to $224 \times 224$. 

Nauta et al.
4.2 Modification Implementation

When implementing the image modifications as described in Section 3.2, we aim for a similar modification strength for all characteristics in order to compare importance scores. Furthermore, the image modifications should be modest, since too extreme modifications can lead to out-of-distribution images that result in erratic behaviour of the underlying neural network of ProtoPNet. A similar modification degree depends on how ProtoPNet perceives the images. Therefore, we find a suitable modification degree by forwarding both the unmodified image and the modified image through the underlying CNN and compare their L1-norm distance in latent space. We tune the modifications parameters such that the mean distance between the unmodified latent training images and the latent modified images is exactly 0.0002 for all characteristics. This value is experimentally chosen such that it results in modifications that are clearly distinguishable for the human eye, while still being perceived by ProtoPNet as being close to the original image. For the colour modifications (contrast, saturation and hue), we use PyTorch’s image transformations. More specifically, we use the ColorJitter function where we set the contrast value to 0.45, saturation to 0.7 and hue to 0.1 for the respective modifications. The shape modification is manually implemented in Python. The texture modification is implemented with the Non-local Means Denoising algorithm [7] for coloured images in OpenCV. The filter strength of the denoising algorithm is set to 4 in order to get the correct mean latent distance.

4.3 Considerations for Evaluation

We would like to emphasise that we do not want to explain human perception, but the perception of the prototype-based model. Hence, we cannot ask users what they deem important, since we aim to explain the model’s reasoning. Also, we cannot construct a ground-truth since we are opening up a “black-box” for which no ground-truth is available. However, we can still do a quantitative analysis to evaluate the generalizability and robustness of the explanations. If generalised well, one would expect that global importance scores are similar when computed for different image sets. Therefore, we compare the global scores for the training set with

---

1 https://github.com/cchen-duke/ProtoPNet, accessed November, 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.
3 https://pytorch.org/docs/stable/torchvision/, accessed June 2020
4 https://docs.opencv.org/3.4/d1/d79/group__photo__denoise.html#ga03aa4189f6c31dadf638d90de335617, accessed June 2020
global explanations computed from the test set. We also evaluate
the distribution of global importance scores to get more insight
in the general model reasoning. Additionally, we qualitatively analy-
se a varied selection of local explanations (Sect. 5.1) and global
explanations (Sect. 5.2). We especially analyse the effectiveness
of our explanations for ’misleading’ prototypes, where our explana-
tions could be different from what a user might expect. Section 5.3
analyses the redundancy of prototypes to answer the question: do
visually similar prototypes focus on similar characteristics or do
they complement each other?

5 RESULTS AND DISCUSSION
ProtoPNet is trained for 30 epochs, reaching a test accuracy of 78.3%.
Having applied 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%). However, the aim of
this paper is not to train the best ProtoPNet, but to find a reasonable
well-performing model in order to explain its prototypes.

Figure 4 (left) shows a selection of prototypical patches (‘proto-
types’) learned by ProtoPNet. ProtoPNet measures the similarity
between a prototype and patches in a given test image. The result-
ning similarity scores are multiplied with learned weights resulting
in a final score per class. During training, softmax is applied to get
a soft prediction. For testing, an image is classified as the class with
the highest score (i.e. most points).

5.1 Analysing our Local Explanations
Figure 4 (right) shows how our local explanations complements
the prototypical reasoning by explaining which visual character-
istics were important for ProtoPNet’s similarity score between a
prototype and a specific image. We show the activation map as
implemented by Chen et al. [11] and list the local importances for
each visual characteristic. The importances identified by our local
explanations for the test image shown in Figure 4 seems reasonable
given the typical white colour of the pelican and its long neck.
Furthermore, our explanations enable a user to understand why
ProtoPNet gave a high similarity score to a prototype and a patch
in the test image. Whereas the prototypes of the white pelican all
look very similar to a human, our local importance scores explain
the differences between the prototypes. The topmost prototype in
Fig. 4 mostly focuses on shape and contrast, the second and fourth
prototype deem hue important and the third prototype focuses on
texture. Although prototypes might look similar to the human eye,
our explanations can differentiate between prototypes and estimate
the prototype’s purpose.

Our local explanations can also be useful to confirm the user’s
expectations, such as the importance of the yellow colour for the
prototype in Figure 5a. The explanations are however especially
insightful when the given similarity score is in contrast with human
perceptual similarity and an explanation is needed. Figure 5b expla-

(a) As expected, the yellow hue is the dominant characteristic, and
the local explanations correspond with the global importance.

(b) Test images can get a high similarity score with the prototype for
different reasons.

Figure 5: Similarity between a class-specific prototype and
test images from different classes explained by our local im-
portance scores.

therefore serves as an extension to a prototypical model, where the
local importance scores explain a single prediction.

5.2 Analysing our Global Explanations
Since our trained 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 an unexpected
result, but do not give a coherent, overall explanation of the model.
Our methodology therefore also produces global explanations that
give an average view regarding the importance scores for each
prototype. Specifically, the global importance scores are computed
for each prototype by taking the weighted mean of all training
images, as introduced in Section 3.3. Hence, these explanations can
be generated before applying the model to a test image, since global
explanations are independent of test input.

5.2.1 Quantitative Evaluation. To quantitatively evaluate our global
explanations, we compute global importance scores twice: in addi-
tion to calculating the scores for $S_{\text{train}}$, the scores are for evaluation
purposes also computed for $S_{\text{test}}$. We confirmed with a Shapiro-
Wilk test that the global importance scores for each characteristic
are normally distributed, such that we could apply the Welch t-test
to confirm that there is no significant difference between the global
importance scores of all prototypes calculated from $S_{\text{train}}$ and from
$S_{\text{test}}$ for each characteristic (p-values < 3.5^{-11}). This verifies that

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|l|}
\hline
Prototype & Test Image & Test Image & Test Image \\
\hline
Global Importance: & Local Importance: & Local Importance: & Local Importance: \\
\hline
Hue 0.1638 & Hue 2.1849 & Hue 2.0172 & Hue 1.9387 \\
Saturation 0.0397 & Saturation 0.8953 & Saturation 0.6743 & Saturation 0.6470 \\
Shape 0.0089 & Shape 0.7201 & Shape 0.4116 & Shape 0.1638 \\
Contrast -0.0025 & Contrast -0.1759 & Contrast -0.2301 & Contrast 0.0956 \\
Texture & & & Texture -0.4691 \\
\hline
\end{tabular}
\end{table}
the global importance scores are generalisable and robust, since the global explanation for a prototype does not significantly change when computed from a different image set.

To get more insight as to how characteristics are used, we also analyse how the global importance scores are distributed per characteristic. Figure 6 plots the distributions of global importance scores of all prototypes across the training set. It shows that the global scores are predominantly positive, which confirms our intuition that decreasing any of the visual characteristics usually leads to a lower similarity score. It also shows that the mean and variability of importance scores is small for saturation and contrast, meaning that those characteristics only have a moderate influence on prototype similarity. In contrast, the high variability for shape, texture and especially hue means that these characteristics can be substantially important for some prototypes. This corresponds with the fact that hue and shape are considered more important and effective for humans in the data visualisation domain than saturation or contrast [28].

5.2.2 Qualitative Evaluation. Figure 7 shows a varied selection of prototypes with their global explanation. For the upper two rows, the importance of characteristics corresponds to the visually identifiable properties of the prototypes and hence, the explanations seem reasonable. However, the explanations in the bottom row might come as a surprise. A human might think that shape is important for the bottom left prototype and that the prototype resembles fin-footed birds. Our global explanation indicates that shape is of little importance and that colour hue is the dominant characteristic. Since a ground-truth is not available, we verify the correctness of the global explanation by analysing test images that had a high similarity with the prototype. Although a prototype is trained to be class-specific, Figure 8a shows images from the test set that are from a different class but still get assigned a high similarity score. These images confirm that the prototype deems hue important and therefore resembles red feet, instead of webbed feet. The reverse is true for the bottom right prototype of Figure 7. Humans could think that the prototype resembles a red eye, and would be surprised by a high similarity score with a black-eyed bird, and hence might lower their trust in the model. Our global importance scores indicate that the importance for hue is rather low, and hue and shape are of similar importance. When analysing birds that get assigned a high similarity with this prototype as shown in Figure 8b, it is easily verified that the prototype does indeed not represent a red eye, but dark and light stripes around an eye.

These examples show that without our explanations, a user would not be aware of the meaning of a given prototype and correct simulatability [26] would not be guaranteed. Our global explanations can therefore clarify visual prototypes and improve the simulatability of a prototype-based model.

5.3 Redundant Prototypes
An interesting question is whether prototypes that are slightly different, deem the same visual characteristics important. If their global importance scores are different, then these prototypes complement each other, whereas similar explanations indicate prototype redundancy. Such prototype redundancy was also discussed by [27], and was found to increase with the number of prototypes per class. We consider prototypes to be visually similar when they are close to each other in the latent space learned by ProtoPNet. 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. Let \( P \) be the set of unique pairs of two prototypes from the same class, such that \( |P| = 9000 \), and \( V \subset P \) the set of pairs with two visually similar prototypes. We consider a pair of two prototypes \( i \) and \( j \)...
Figure 9: Histogram with the distribution of Euclidean distances between the global explanations of two prototypes of the same class (a ‘pair’).

A prototype-based image recognition model learns visual prototypes and localises a patch in a test image that looks alike a prototype to assign it a similarity score. We argue that these prototypes should be explained with respect to the model’s reasoning and extend localisation with explanation. 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 the model’s similarity score with a prototype. In this way, we identify which visual characteristics of a prototype the model deems important. 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, indicating the need for explaining the model’s reasoning. Such ‘misleading’ prototypes will hinder correct simulatability and only visualising 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 to address such ambiguity of visual prototypes and the elegant simplicity of our approach makes it a suitable stand-alone solution.

Future work concerns the potential interactions between characteristics. Our importance scores currently assume that characteristics from image modifications are mutually exclusive. 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, including ProtoPNet [11] and ProtoTree [29]. 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, 5238–5336.

[2] Sercan Omer Arik and Tomas Pfister. 2019. Attention-Based Prototype 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 Barreda-Tritsch, Atsila Diaz-Rodrigues, Javier Del Ser, Adrien Bernet, Sahim Tahik, Alberto Barbado, Salvador Garcia, Sergio Gil López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Conceptions, taxonomies, opportunities and challenges toward responsible AI. Conception 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] Antóni Brudé, 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.bcnl

[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 BMI Adaptation and Visual Object Agnosia. Cerebral Cortex 20, 10 (1 October 2010), 2319–2332. https://doi.org/10.1093/cercor/bhp298 arXiv:https://arxiv.org/abs/cercor/article/pdf/10.20/2319/17302013/bhp298.pdf

[9] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mana Seravastava, 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 Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/BDCom/JOPSICS). 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, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep Learning for Interpretable Fine-grained Image Recognition. arXiv preprint arXiv:1812.02406 (2020).

[12] 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

[13] F. Plesse, A. Ginsca, B. Delezoide, and F. Prêteux. 2018. Learning Prototypes for 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/10.1145/3292500.3339908

[14] Tamara Munøn and Eamonn Maguire. 2015. Visualization analysis & design. http://www.cernebase.com/?s=9781465059894. OCLC: 897096916.

[15] Meike Nauta, Ron van Bree, and Christin Seifert. 2020. Neural Prototype Trees for Interpretable Fine-grained Image Recognition. arXiv preprint arXiv:2012.08460 (2020).

[16] 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

[17] Amir Rosenfeld, Markus D. Sobchak, and John K. Tistros. 2018. Totally Looks Like - How Humans Compare, Compared to Machines. 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/AAAIAA18/paper/view/17082

[18] Zachary C. Lipton. 2018. The Myths of Model Interpretability. Queue 16, 3, Article 30 (June 2018), 27 pages. https://doi.org/10.1145/3263836.3241340

[19] Yao Ming, Panpan Xu, Haomin 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/10.1145/3292500.3339908

[20] F. Plesse, A. Ginsca, B. Delezoide, and F. Prêteux. 2018. Learning Prototypes for Interpretable Fine-grained Image Recognition. arXiv preprint arXiv:1812.02406 (2020).

[21] Marshall H Segall, Donald Thomas Campbell, and Melville Jean Herskovits. 1966. The influence of culture on visual perception. Bobbs-Merrill Indianapolis.

[22] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[43] 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.

[44] 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)*.

[45] 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.

[46] 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)*.