Are Learned Concepts Understandable?

Matteo Destro (221222)
Advanced Topics in Machine Learning and Optimization
M.Sc. in Artificial Intelligence Systems, University of Trento
matteo.destro@studenti.unitn.it

Abstract

A neural network explainability is an important feature for ensuring that it can be trusted to make the correct decisions. In this work, we analyze the prototypes learned by a Prototypical Part Network (ProtoPNet) for two different domains and provide an objective way to evaluate the degree of interpretability of the learned concepts. Possible improvements for increasing the model's explainability are then discussed, either by introducing additional regularization or by exploiting auxiliary data. The source code and the datasets are made available at https://github.com/materight/understandable-ProtoPNet.

1 Introduction

Convolutional Neural Networks have become the standard for image classification due to their performance on a large number of domains. However, these models operates as black boxes, making it difficult for a human to understand why a network produced a certain prediction. This can be particularly problematic in domains where the decision making process needs to be transparent.

For this reason the field of explainable AI is actively researched and multiple solutions have been proposed in the last years for increasing the understandability of these models. For instance, *Prototype Classification Networks* (Li et al., 2017) are a family of models designed to increase the explainability of neural networks by learning prototype representations of the classes, which can then be used to understand the decision process of the model. *Prototypical Part Networks* (ProtoPNet) (Chen et al., 2018) are an evolution of PCNs: the learned prototypes are represented by small patches in the existing training samples, instead of the full images. With this approach the network is able to learn more specific concepts, i.e. a beak type or plumage texture, which in turn increases the explainability of the predictions.

Previous studies have been shown that these models are able to reach state-of-the-art performances on multiple domains (e.g. bird species, car models (Chen et al., 2018), Covid-19 detection (Singh and Yow, 2021) and road signs (Gautam et al., 2021)), with the advantage of providing a good level of interpretability, since each prediction can be explained by the most activated prototypes of the input image.

However, these methods do not enforce any constraint to ensure that the learned prototypes are actually understandable by a human observer. This can lead to learning seemingly meaningless concepts, which cannot be used to understand the decision making process. Previous studies tried to automatically infer explanations from the learned concepts based on color and texture patterns (Nauta et al., 2020), but they completely disregard the semantic content of the prototypes.

In this work an analysis of the learned prototypes for two different domains is given, with a focus on identifying those cases where the model fails in providing a good explanation on the classification prediction. Possible improvements to mitigate those cases are then discussed, either by introducing additional regularization or by exploiting auxiliary data.

2 Experimental setting

The first part of this project involved training a ProtoPNet model on two datasets to perform a first evaluation on the network performances. The following sections will explain how the datasets were prepared and how the models were trained for the classification tasks.

2.1 Datasets

Two datasets were used for training and evaluation: *CUB-200-2011* (Wah et al., 2011) and *CelebAMask-HQ* (Lee et al., 2020).

CUB-200-2011 is a dataset comprising of 11 788 samples from 200 classes of birds species. Each samples is also annotated with 15 part locations, like "beak", "crown" or "tail". The dataset was split into

5 994 samples for training and 5 794 for testing.

CelebAMask-HQ is a large-scale celebrity face image dataset comprised of 30 000 samples. For each sample, a segmentation mask of facial attributes is provided. Moreover, each image is annotated with 19 attributes of facial features, like "hair color", "gender" or "age". The dataset was split into 21 047 samples for training and 8 953 for testing. Since no classes are provided, multiple classification tasks were extracted from the attributes annotations, as shown in Table 1.

Attribute	Extracted classes
Hair color	Bald / Black / Blonde / Brown / Gray
Gender	Female / Male
Age	Old / Young
Makeup	Heavy makeup / No makeup

Table 1: Classification subtasks extracted from the attributes of the CelebAMask dataset.

2.2 Model training

Initially a different model was trained for the CUB-200 dataset and for each of the CelebAMask subtasks. The training was performed for 500 epochs with a batch size of 128, using a ResNet-34 model as backbone. A different number of prototypes were learned based on the type of classification task: in general 10 prototypes were used for each class, following the rule of thumb presented in the original paper. At each epoch the training set was augmented with random affine, perspective, and horizontal flip transformations.

2.3 Classification performances

An overview of the model performances on the classification tasks can be found in Table 2. It must be noted that for the CUB-200-2011 dataset the model obtained a best accuracy on the test split of 77.52, while in the original work by (Chen et al., 2018) the same backbone with the same parameters obtained 79.20. An example of the learned concepts for each dataset can be found in Figure 1.

Dataset	# Proto.	Acc.	$\overline{d}_{\mathbf{P}}$
CUB-200	2000	77.52	4.12
CelebA[Hair color]	50	94.42	3.26
CelebA[Gender]	20	98.85	4.86
CelebA[Age]	20	89.88	3.95
CelebA[Makeup]	20	91.80	4.25

Table 2: Performances obtained by the model on the different datasets and subtasks. $\overline{d}_{\mathbf{P}}$ represents the average Euclidean distance between the prototypes embeddings.

Figure 1: Examples of the learned prototypes with their corresponding activation map in the original training sample.

3 Prototypes evaluation

To better understand if the part prototypes are actually representative of a set of distinctive features that can be easily recognized by humans, a first manual evaluation was conducted by visualizing the learned concepts and their activations on different images. A more objective benchmark was then performed by computing an *alignment score*, exploiting the available part annotations in the datasets.

3.1 Odd behaviours of learned concepts

The original study (Chen et al., 2018) showed that the ProtoPNet approach is effective in learning meaningful concepts, allowing a human observer to understand the decision making process without affecting the classification performances. However, this is not always the case, and there are multiple instances where the learned concepts do not have any clear meaning or are taken from apparently pointless regions. These odd behaviors can be classified into different categories:

Random regions: in some cases the learned concepts are seemingly meaningless, or tend to activate on apparently random regions of the input image. An example is shown in Figure 2(c): while the prototype is meaningful since it represents an instance of black hair, the corresponding activations on the samples from the test split are not, except for the last case.

Wide-spread concepts: sometimes a learned prototype might span a large portion of the image, as it can be seen in Figure 2(e). While the prototypes are still relevant since they are more similar to the concepts

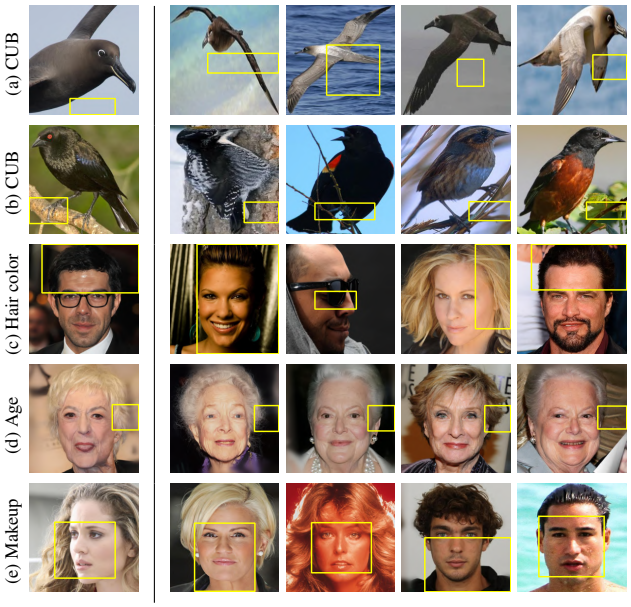


Figure 2: Examples of prototypes (on the left) and corresponding highly activated patches (on the right) on apparently meaningless or background regions.

learned by a Prototype Classification Network, they make it impossible to identify a single part or detail which was relevant in the decision process.

Background prototypes: in some other cases the network might learn concepts from background patches, as show in Figure 2(a) and Figure 2(b). This might happen because even if these prototypes do not provide a human observer with an useful explanation of the model decision, they still can be exploited effectively by the model to distinguish the different classes: for instance, detecting a patch of water might indicate a species of birds mostly found in water regions, or a tree branch may point to a species mostly found in forests.

Misclassified concepts: since the prototypes are taken from small image patches and therefore lack of global context, it might happen that a concept of a class strongly activates on a sample from another class which has a similar representation, as shown in Figure 3. This issue is particularly important since it affects the model classification performances, but is partially solved by the pruning step discussed in Section 3.2.

Dissimilar activations: another odd behavior appears when a prototypes activates concepts that apparently have nothing to do with the original part, even if the class of the input image and the prototype is the same. Some examples of these cases are presented in Figure 4. This is most likely cause by the type of prototypes extracted: a texture prototype might activate on different locations of an object that presents

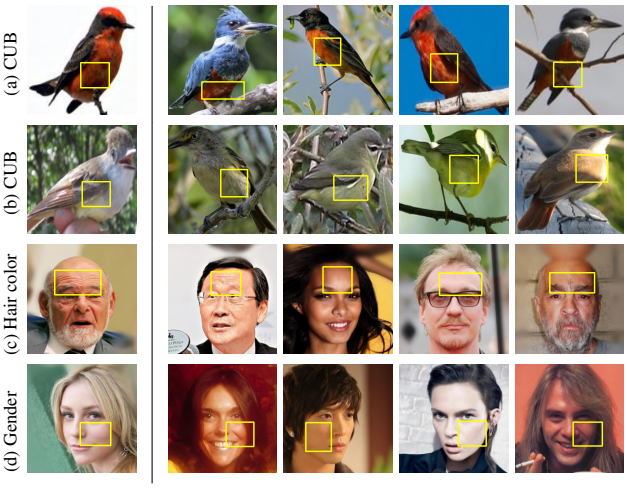


Figure 3: Examples of prototypes strongly activated on similar concepts present in different classes.

the same patterns. Indeed, the only instances of this type of behavior were found in the CUB and hair color datasets, in which texture prototypes are learned on the plumage and hair respectively. While this behaviour is less damaging than the previous ones, since the concept do not hinder the model performances and a certain level of explainability is retained, any attempt to link a prototype to a specific object part is ineffective on these cases. This is particularly relevant for the alignment study discussed in Section 5.

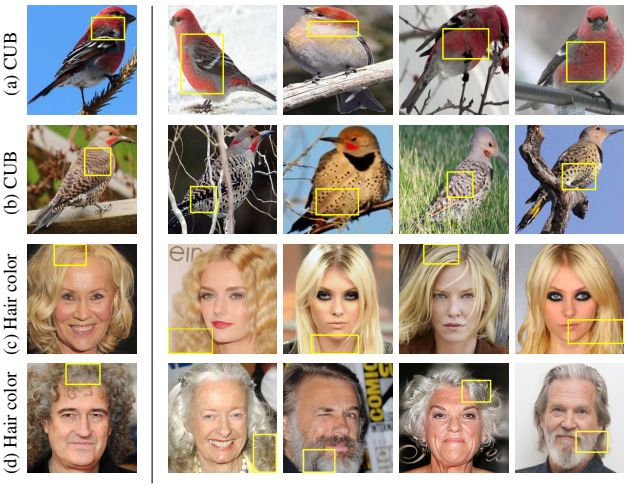


Figure 4: Examples of single prototypes activated on different parts on the same classes of images.

Duplicated concepts: sometimes duplicated prototypes are learned, as shown in Figure 5, since the projection step regularize the learned concepts to be closer to a patch in a training sample. This is generally not an issue if the classification problem is rather complex, since the network is still able to learn multiple concepts for each class. However, if the task is rather

simple to solve, for instance binary classification on a person gender, this might result in the network learning a single prototype which representation might be discriminative enough, as shown for the case in Figure 5(c).

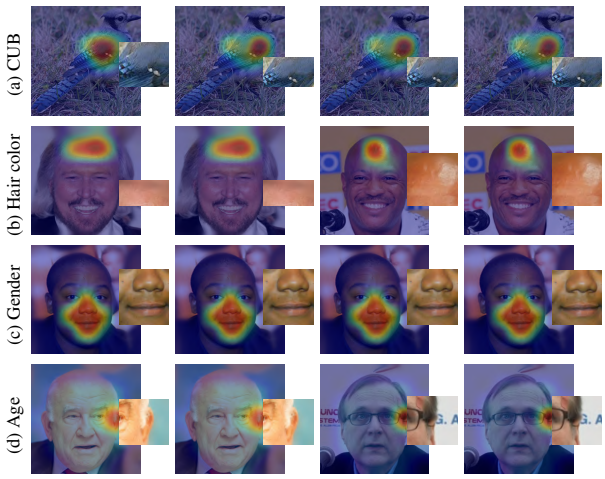


Figure 5: Examples of duplicated prototypes learned by the model.

3.2 Prototypes pruning

The original paper proposes a *pruning* step to mitigate some of these issues, in particular to remove concepts that strongly activates patches in the wrong classes. In this step, for each prototype, the k most activated training images are retrieved, and the prototype is discarded if out of those k, τ images are from a class different from the one of the learned concept. Since background concepts tend to activate on samples from multiple classes, this approach is also effective in removing them: for instance, in the example discussed previously there are multiple classes representing water birds, so prototypes representing water patches will be discarded.

4 Improving prototypes diversity

As discussed in Section 3.1, a common issue with the projection step of the training algorithm is that it tends to create duplicated prototypes, since the learned vector embeddings are forced to be similar to existing image patches in the training samples. This is generally fine if the duplicates are limited, but as already seen it might happen that a large number of class prototypes, if not all, are just copies of each other. Even if this may have no impact in the model performances, it generally hinders the explainability of the results, since few or a single prototype might be too general to be used to understand the model predictions.

To limit these issues, a *diversity* regularization term can be introduced to force the learned prototypes to be different w.r.t. each others up to a certain degree. This term is computed as the sum of Euclidean distances between all possible pairs of prototypes, which is then maximized during SGD for those pairs whose embeddings are too close. In more details, a *Div* term is introduced in the loss function, computed as shown in Eq. 1:

$$\text{Div} = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \max\left(0, d_{min} - \|\mathbf{p}_i - \mathbf{p}_j\|_2^2\right) \quad (1)$$

where d_{min} is an hyperparameter denoting the minimum acceptable distance between prototypes. With this implementation, a distance between $\mathbf{p}_i$ and $\mathbf{p}_j$ above d_{min} it is considered "acceptable". Otherwise the two prototypes embeddings are pushed apart to maximize diversity.

This regularization term is then added to the general loss function presented in the original work (Chen et al., 2018) as show in Eq. 2:

$$\min_{\mathbf{P},\omega_{\text{conv}}} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt} + \lambda_1 \text{Clst} + \lambda_2 \text{Sep} + \lambda_3 \text{Div} \quad (2)$$

An optimal value of 1.0 for d_{min} was determined by analysing the distances between the prototypes learned without the diversity regularization, while the value for λ_3 was set to 0.5 after a cross-validation experiment.

Table 3 shows that this approach is effective in producing more diverse prototypes, in some cases even improving the classification performances. The average prototypes distance $\bar{d}_{\mathbf{P}}$ is measured as the average Euclidean distance between each pair of prototypes, and an higher score indicate more diverse prototypes: compared to the values shown in Table 2, this regularization method results in a generally higher $\bar{d}_{\mathbf{P}}$.

Dataset	d_{min}	**Acc.**	$\bar{d}_{\mathbf{P}}$
CUB-200	0.1	76.72	6.23
CUB-200	1.0	80.97	9.07
CUB-200	5.0	78.67	9.12
CelebA[Hair]	1.0	93.75	5.37
CelebA[Gender]	1.0	98.94	7.92

Table 3: Performances obtained by the model trained with the diversity regularization of Eq. 1.

5 Part locations alignment

For a learned concept to be easily understandable by a human, it should represent a particular distinctive feature of the subject, otherwise it might be difficult for an

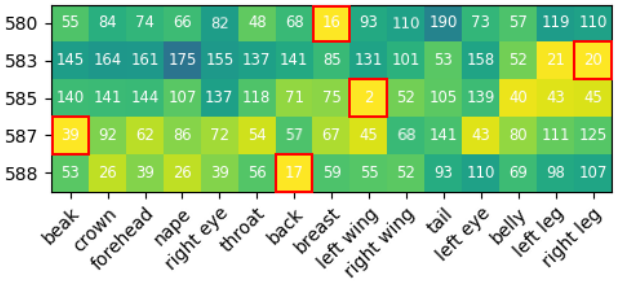
observer to understand what the discriminative feature is. For instance, a good prototype for classifying birds might be the color of the beak: if a prototype activates exclusively on the beak, it is easy for an observer to understand why the model produces that prediction. However, if a prototype activates on a large part of the body, or on an apparently meaningless spot, the explainability of the model is lost.

Dataset	Annotated parts
CUB-200	beak, crown, forehead, nape, right eye, left eye, throat, back, breast, left wing, right wing, tail, belly, left leg, right leg
CelebA	left brow, right brow, left eye, right eye, left ear, right ear, lower lip, upper lip, mouth, neck, nose, skin, hair, hat

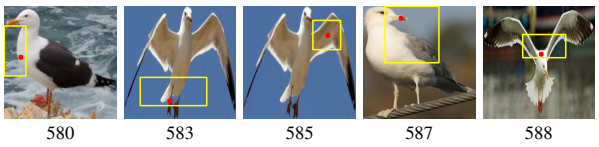
Table 4: Available part locations for the two datasets.

To investigate whether the learned prototypes activates on meaningful parts and to provide a more objective evaluation, an *alignment score* was implemented by exploiting the available part annotations of the datasets. Each part is characterized by a name and a (x, y) tuple denoting its position in the image. Table 4 shows the available parts for each datasets. To compute the alignment score, for each sample the patches that highly activate each prototype from the sample's class are computed. Then, the distances between their centroids and each annotated part is used to compute a matrix of the sample's alignment scores of size $|\mathrm{P}_j| \times |\mathrm{L}_i|$, where P_j is the set of prototypes for class j, and L_i is the set of part locations for sample i, with class $\mathrm{C}_i = j$. The parts for which no location annotation is given are ignored. All the alignment matrices of all the samples of a given class are then averaged to produce the final alignment matrix, as shown in Figure 6 and Figure 7. This matrix can be used to easily describe which prototypes are learned by the model to discriminate a certain class and how they relate to human-understandable concepts.

To then summarize these scores into a single metric, for each concept the part location with the smallest average distance (annotated in red in the previous figures) is taken as final score for that concept. This single value represents how much a prototype is representative of a certain part: a lower score indicates a good representative concept, while an higher value might indicate that the prototype activates on different parts on different samples, and therefore might be more difficult to understand for a human observer.



(a) Average distance between the prototypes activations on images of the same class w.r.t part location annotations. The parts with the highest alignment are annotated in red.



(b) Prototypes bounding boxes with nearest part location (red dot).

Figure 6: Alignment matrix and prototypes for the "California gull" class from the CUB-200 dataset.

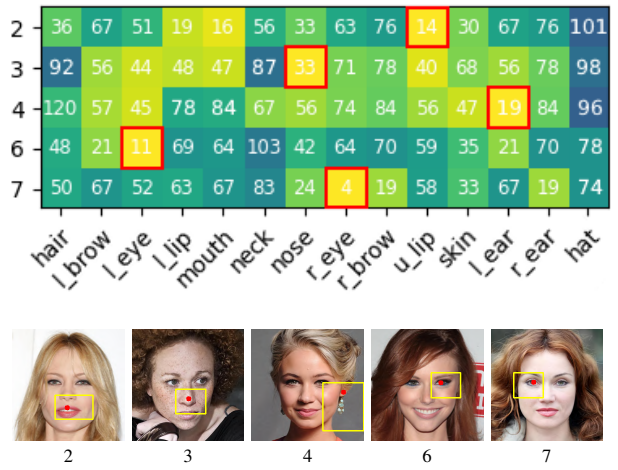


Figure 7: Alignment matrix and prototypes for the "Female" class from the CelebA[Gender] dataset.

The scores of each prototype can be then averaged to obtain a single global alignment score $\alpha_{\mathbf{P}}$ for the whole dataset. This metric can be an effective way for validating different techniques to improve the concepts understandability.

To further explore this idea, an additional study was conducted on the CUB-200 dataset: Table 5 presents how the global alignment score changes w.r.t. the number of epochs, the projection step, the prototypes pruning, and the diversity regularization discussed in Section 4. Looking at the results it is clear that there is an inverse correlation between the alignment score and the model accuracy: in general a lower (better) alignment corresponds to an higher accuracy. Moreover, it is clear that all the methods discussed previously

Method	Epochs	Acc.	$\alpha_{\mathbf{P}}$
None	100	71.68	45.09
	250	73.82	42.51
	500	76.96	36.46
Projection	500	77.52	31.72
Pruning	500	79.14	34.62
Diversity reg.	500	80.97	28.46
All	500	81.22	25.21

Table 5: Accuracy and alignment scores on the CUB-200 dataset w.r.t. different methods to improve the learned concepts. For the "None" method the model was trained and evaluated without any additional technique.

are effective in increasing the alignment score, and thus the understandability of the concepts, especially when combined. The diversity regularization seems to be the most effective, probably because it forces the model to learn new concepts and therefore increases the probability of learning prototypes with a better alignment.

6 Conclusions

In this work the interpretability of the learned concepts of a ProtoPNet model was explored. The results showed that a significant number of prototypes are not easily understandable by a human observer, which might be problematic when trying to use the learned concepts to explain a model predictions. An objective evaluation was then performed by computing an *alignment score* on the learned prototypes w.r.t. the available part annotations of each dataset, which can be used to validate different methods for increasing the concepts interpretability. Moreover, this study introduced a *diversity regularization*, which proved to be effective in producing more diverse prototypes that generally result in an higher alignment score. Further improvements could include using the available part locations as an additional regularization term, to force the model to learn well-defined concepts.

References

Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. 2018. This looks like that: Deep learning for interpretable image recognition.

Srishti Gautam, Marina M. C. Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. 2021. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2017. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions.

Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. 2020. This looks like that, because ... explaining prototypes for interpretable image recognition.

Gurmail Singh and Kin Choong Yow. 2021. An interpretable deep learning model for covid-19 detection with chest x-ray images. *IEEE Access*, 9:85198–85208.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.