Explainable Image Classification using Segment Clustering and Decision Trees

Le Thi Nhi Ha
lethinhiha@student.utwente.nl

Niclas van Eyk
j.vaneyk@student.utwente.nl

June 27, 2021

Abstract

This paper describes a method for explaining the results of an image classifier, similar to existing approaches like ProtoTree [1], but based on simpler classification algorithms. ProtoTree was first introduced in April 2020 by Nauta, Bree, and Seifert. The main idea is building a decision tree from a trainable node to recognize a fine-grained image. Although the approach is clever, the training process is rather complex and not all image patches can be easily interpreted by the user. These drawbacks motivated us to implement a simpler approach to construct the decision tree, based on the similarity between segments of the input images and higher-level concepts surfaced through clustering. The method achieves an accuracy of up to 72% and enables us to visualize the decision process using a graph instead of a tree.

1 Introduction

Machine Learning (ML) models are used around the world to aid humans in decision making processes or to obtain new insights from existing data. However it is often hard to interpret *why* a certain model came to its resulting decision, which is why the interpretability of such models is getting more important as their usage grows.

Projects like ProtoTree [1] show how such interpretability can be achieved. A decision tree is built, which contains an image patch at each node. This way the decision making process is transparent to the user, as the image patch of the node can be easily compared to the input image. However it has some drawbacks, namely that the image patches are not always easily interpretable and that the learning process is quite complicated.

This work is now concerned with the construction of a similar type of interpretable classifier using an alternative approach, which tries to avoid the drawbacks of the ProtoTree method. Images are split into smaller segments to isolate shared concepts between classes. After obtaining a higher-level interpretation of these segments by passing them through a *Convolutional Neural Network* (CNN), similar segments are grouped together. The similarities of an input image to these clusters then forms the basis for our classification and visualization method.

2 Related Works

As already stated, this work builds upon the ideas of ProtoTree [1], which combines decision trees, neural networks and an image patch at each node. ProtoTree is in turn based on the ideas of ProtoPNet [3], which is where the idea of using parts of an image to form a prototype for interpretability originates. This work uses the "bag-of-visual-words" shared by both papers, but using hierarchial clustering algorithms instead of neuronal networks to build a decision tree.

Another way to achieve global explainable AI in image recognition is Automatic Concept-based Explanation (ACE) [2]. The idea behind this model is extracting higher level concepts about textures, col-

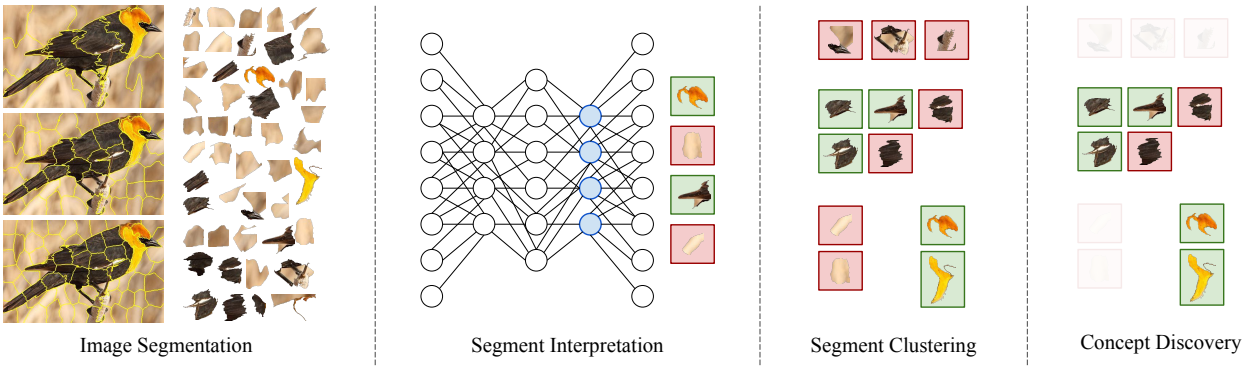

Figure 1: The steps described in subsection 3.1 to discover concepts in the source images. These are similar to [2], but use a different discovery mechanism. Correctly classified segments are highlighted in green, while incorrect ones have a red background.

ors and shapes from the images. For example the most distinct feature of a zebra is that it has stripes, a less unique feature is that it is has legs or that is has a certain size. To create these concepts for each class, images are segmented into patches at different sizes and clustered according to their similarity. After an additional outlier removal step the TCAV importance score [4] is calculated. Our methodology starts with similar steps to those of the ACE paper but goes further by using the clusters to build a decision tree similar to ProtoTree.

Specific excerpts of an input image were also used during the training process of CNNs like BagNet [5] or NtsNet [6]. Both of these networks are trained on "zoomed-in" regions of the images, which could be as small 33x33 pixels for BagNet, while NtsNet divides a picture into a fixed amount of regions per input image. This enables the networks to classify an image based on only segments. Using this technique, they achieve a classification accuracy for the input images of 87%. We want to leverage this ability, while using non-rectangular excerpts of the images that are filled with a placeholder color, similar to how it was done in the ACE paper.

3 Methodology

We use the *Caltech-UCSD Birds 200* (CUB-200) dataset [7] to classify the species of a bird shown in a given image. The goal is to be able to reason about the classification result by showing a visualization to the user.

Overall, to build the explainable classification decision tree, our approach contains two main steps. The first one is concerned with extracting *concepts* from all source images, for example displaying a black tail, red wings or a particularly shaped beak. The second step is building a decision tree based on the segments of each image and their similarity to the discovered concepts. We will elaborate further how we conduct on each stage.

3.1 Discovering Concepts

Our approach for this step is similar to the one in the ACE paper [2] and can be seen in Figure 1.

Image Segmentation The images are segmented according to the *Simple Linear Iterative Clustering* (SLIC) algorithm [8] implemented by the *scikit-learn* [9] library. It divides the image using K-Means clustering in the (x,y,z) color space. The segmentation is repeated at multiple resolutions (15, 50 and 80 segments per image, see Figure 1), to capture the

complete aspects of an image like patterns, colors, parts and objects [2]. Only the unique segments are kept, similar segments are discarded based on the jaccard similarity of their segment outlines. The remaining segments are finally resized to the shape required by the network used in the following steps (448x448).

Segment Interpretation The segments are passed through a pre-trained CNN. We use NtsNet [6], which was trained on the same CUB-200 dataset and achieves a classification accuracy of up to 87.5%. This specific network was trained on different "zoomed-in" regions for each input image, so a similar approach to our segmentation was followed. This increases the classification accuracy for our segments, which more closely resemble these "zoomed-in" regions, rather than the full input image. Each segment is passed through the network, while extracting the activation space from the last fully connected layer (indicated by the blue nodes in Figure 1), which has a shape of (2048, 0). This layer captures the networks higher-level interpretation of what the image contains, which will be used in the next step.

Segment Clustering & Concept Discovery Now we use K-Means clustering to group similar segments based on the euclidean distance between the activation spaces obtained from the prior step. By tracing back the segments in each cluster to their original bird species, we gain insights about what prototypical features (*concepts*) a certain species is made up of.

There is some cleaning necessary in this step to actually arrive at meaningful concepts that are related to the contained birds. In the dataset, the portion of the pictures that actually display parts of the bird are pretty small. Each image usually is segmentized in 15 segments, and only two or three of them contain parts of the bird, the remainder shows parts of the background. This leads to multiple clusters that contain segments of background information, which should not be learned by our model. To exclude these, we save whether or not the segment was classified correctly when it gets interpreted by the CNN in one of the previous steps (indicated by the red and green boxes in Figure 1). This enables us to compute an average classification accuracy for each cluster. The theory is that this accuracy will be higher for clusters containing parts of the birds, as the network was trained to classify them. A segment showing a bird will have a higher chance of being classified correctly, than one only showing a patch of grass.

After calculating these accuracy scores for each cluster, we extract those over a certain accuracy threshold. For our tests, using 50% yielded the best results. This results in a set of *important clusters* representing only the concepts regarding the birds. An excerpt of such a cluster is displayed in Figure 2, where the segments are exclusively displaying parts of the bird and no background greenery, even though they contain predominantly green segments.

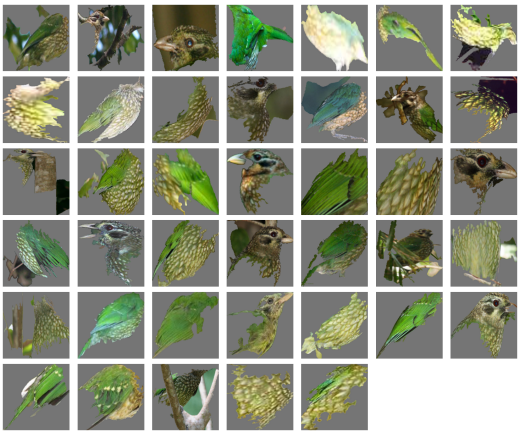

Figure 2: Important cluster containing green segments (20 classes and 40 clusters were used).

3.2 Decision Tree Construction and Classification

To be able to build the Decision Tree model, we compute a feature vector for each image. This feature vector contains the similarity of the image to each of the important clusters. For each image we iterate over the clusters, compute the cosine similarity between its center and all segments originating from

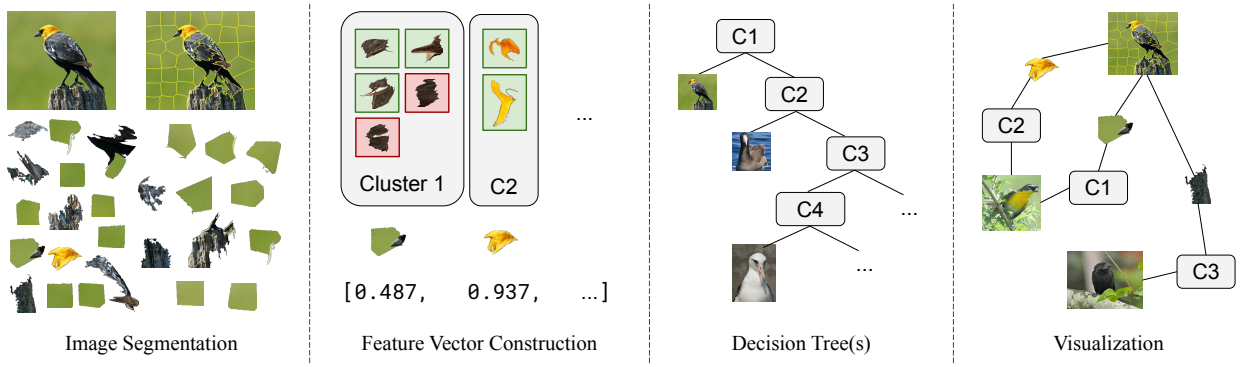

Figure 3: The steps described in subsection 3.2 to map each image to it's feature vector and construct a decision tree, which classification result is finally visualized using a graph.

the image, and keep the highest similarity value for each cluster. A visual representation of the computation is displayed in the left half of Figure 3. The result for a single image is a one-dimensional vector, where each index corresponds to an important cluster and the value at that index represents the highest similarity score between the cluster center and the segments of that image.

For the training process we use the `DecisionTreeClassifier` [10] provided by scikit-learn, which implements a modified version of the *Classification and Regression Trees* (CART) [11] algorithm and the Gini impurity measure. Each node in the resulting decision tree can be translated to the binary question of whether or not the concept represented by a cluster can also be found in the input picture.

Using this method, there is a huge distance between the training and test accuracy, which indicates overfitting. To avoid this, we utilize the Random Forest algorithm (100 estimators). When using this approach, we also slightly changed the concept discovery step. Instead of having a generic threshold for the accuracy of a cluster, we discard segments that were misclassified, which also improves the accuracy.

Visualizing the decision process is now as simple as attaching one or more representants of each cluster to its corresponding node in the decision tree, as can be seen in the right half of Figure 3. As the resulting trees are very unbalanced, we do not visualize them directly. If a certain threshold that was learned by the tree is exceeded at a node, this cluster/concept will have a high similarity with the input image. Then a graph is constructed, which connects the source image's segments and the similar concepts that were surfaced by the decision process of the tree. An example of such a graph is displayed in Figure 4. To not only show the user a local representation of the decision path, we decided to add images of classes which segments are also contained in the concepts used for the decision. This way our visualization is neither fully global or local, and the viewer can explore related classes, which improves the overall recall of our method. We do not present a single classification result, but also other options, which may contain the correct class in the case of a classification error.

4 Results

All in all this method[1] achieves a classification accuracy of up to around 72%. While it is substantially higher than guessing randomly for the 200 different bird classes contained in the dataset, it is also lower than the 87% that NtsNet and Prototree achieve.

The results of the classification vary depending on which parameters are used during the training and decision tree construction process. The most influential parameter is the amount of clusters k used when

[1] Code available at `https://github.com/stubbornfox/pcc`

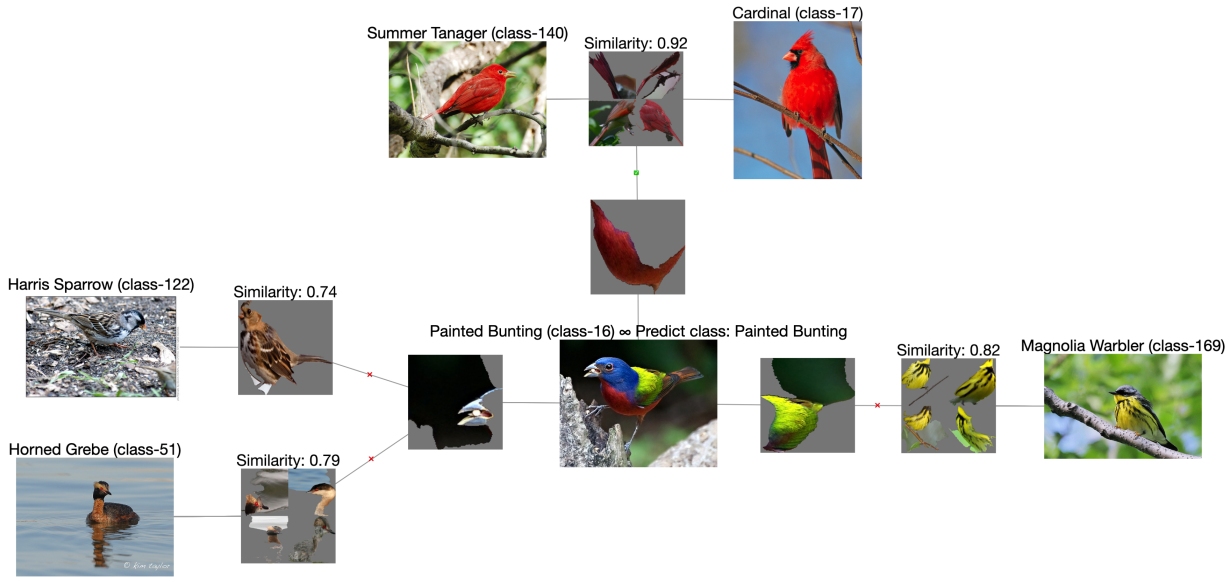

Figure 4: Graph explaining the classification of an image containing a Painted Bunting. As can be seen, the beak and other characteristic colors are matched with clusters containing similar images and birds similar to that cluster.

Table 1: Comparison between the accuracies achieved by different methods.

Method	clusters	accuracy	
		train	test
Ours (one tree)	1000	83%	58%
Ours (random forest)	1300	92%	72%
NtsNet			87%
ProtoTree			87%

grouping similar segments. If k is too high, one can observe a duplication of concepts. In this case, as e.g. multiple bird species in the dataset have a black chest or black wings, there are multiple clusters containing primarily black segments. If k is chosen too low, there is less duplication, but segments displaying slightly different concepts are sometimes grouped together, which is also not ideal and is likely decreasing the accuracy. For our results, we tested multiple values of k and chose the ones yielding the highest accuracy.

Another reason for the lower accuracy is that we do not take every aspect of the pictures into account, even though they might increase the accuracy. Pictures of certain bird species like seagulls are likely to have some part of the ocean or other parts of water contained in their images. The CNN could use this information to learn that seagulls are likely to appear near water and use this knowledge in the classification process. This might not be a problem when classifying birds, but could be an issue for other classification problems, e.g. identifying brain tumors on x-ray images. Furthermore our method of filtering out background segments is not perfect and can sometimes lead to clusters being excluded even though they contain mostly segments of birds, or include clusters that contain parts of the background.

The visualization is able to explain the classification result well. The only cases where it is not as

expressive, is when there are only a small amount of concepts to be found that are close enough to be considered by the decision tree. In this case the decision tree bases its decision on the similarity to possibly only one cluster. A strength of the visualization is that this is transparent to the user, who can take this into account when communicating the results to others.

5 Conclusion

In this paper we showed an approach for explaining the reasoning behind image classification results which combines conventional clustering methods with specialized CNNs. Concepts are extracted by grouping similar segments based on their higher level interpretation obtained from the activation layer of a neural network. Simple classification methods like decision trees and random forests are then responsible for the actual classification of the image based on the similarity in activation space between segments of the input image and the concepts derived from the training data. This final classification result is displayed on a graph connecting the image segments of the input image to the derived concepts. To provide more context and achieve better recall, images of similar classes are also added to the graph.

This method achieves a classification accuracy of up to 72%. This is roughly 15% points lower than the one of the CNN used to obtain the higher level interpretation of an image and the one of more complex explainable AI methods like ProtoTree. This cost can be attributed to the simplicity of the final classification method and the fact that some information like the background is discarded in order for the model to only focus on the actual classification subject. While this cost is significant, the method makes up for it by enabling a visual explanation for the classification result.

References

[1] Meike Nauta, Ron van Bree, and Christin Seifert. "Neural Prototype Trees for Interpretable Fine-grained Image Recognition". In: *CoRR* abs/2012.02046 (2020).

[2] Amirata Ghorbani et al. "Towards automatic concept-based explanations". In: *arXiv preprint arXiv:1902.03129* (2019).

[3] Chaofan Chen et al. "This looks like that: deep learning for interpretable image recognition". In: *arXiv preprint arXiv:1806.10574* (2018).

[4] Been Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)". In: *International conference on machine learning*. PMLR. 2018, pp. 2668–2677.

[5] Wieland Brendel and Matthias Bethge. "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet". In: *arXiv preprint arXiv:1904.00760* (2019).

[6] Shah Nawaz et al. "Are These Birds Similar: Learning Branched Networks for Fine-grained Representations". In: *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ 2019)*. Dec. 2019. URL: `https://github.com/nicolalandro/ntsnet-cub200`.

[7] P. Welinder et al. *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010.

[8] URL: `https://scikit-image.org/docs/dev/api/skimage.segmentation.html?highlight=slic#skimage.segmentation.slic`.

[9] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[10] URL: `https://scikit-learn.org/stable/modules/tree.html`.

[11] Leo Breiman et al. *Classification and regression trees*. CRC press, 1984.