Explaining Convolutional Neural Networks by Tagging Filters

Anna Nguyen[1], Daniel Hagenmayer[1], Tobias Weller[2] and Michael Färber[1]

[1]Karlsruhe Institute of Technology (KIT), Institute AIFB, Karlsruhe, Germany
[2]University of Mannheim, Mannheim, Germany

Abstract

Convolutional neural networks (CNNs) have achieved astonishing performance on various image classification tasks, but it is difficult for humans to understand how a classification comes about. Recent literature proposes methods to explain the classification process to humans. These focus mostly on visualizing feature maps and filter weights, which are not very intuitive for non-experts. In this paper, we propose FILTag, an approach to effectively explain CNNs even to non-experts. The idea is that if images of a class frequently activate a convolutional filter, that filter will be tagged with that class. Based on the tagging, individual image classifications can then be intuitively explained using the tags of the filters that the input image activates. Finally, we show that the tags are useful in analyzing classification errors caused by noisy input images and that the tags can be further processed by machines.

Keywords

CNN, images, explainable AI, semantic interpretability

1. Introduction

Deep convolutional neural networks (CNNs) are the state-of-the-art machine learning technique for image classification [1, 2]. In contrast to traditional feed-forward neural networks, CNNs have layers that perform a convolutional step (see Figure 2 for the relations in a convolution). Filters are used in a convolutional step which outputs a feature map in which activated neurons highlight certain patterns of the input image. Although CNNs achieve high accuracy on many classification tasks, these models do not provide an explanation (i.e., decisive information) of the classifications. Thus, researchers recently focused on methods to explain how CNNs classify images.

Related Work. Some of the earliest works on explaining CNNs focus on visualizing the activations of individual neurons [3, 4]. However, these methods cannot explain more complex relationships between multiple neurons, as no human-understandable explanation is used. Olah et al. [5] defined a semantic dictionary by pairing every neuron activation with its abstract visualization using a channel attribution, determining how much each channel contributes to the classification result. This may explain the role of a channel in the classification of an individual image, but it does not explain the role of that channel across all possible input images. Hohman et al. [6] try to overcome this problem by aggregating particularly important neurons and identifying relations between them. Other approaches focus on filters, the discerning feature of CNNs. For example, Zeiler and Fergus [7] visualize the filter weights to illustrate the patterns these filters detect. However, these visualizations are based on the inputs of the layers to which the respective filter belongs to. Thus, only the filter patterns of the first layer can be directly associated with patterns on the input image of the network. To overcome this, the method Net2Vec [8] quantifies how concepts are encoded by filters by examining

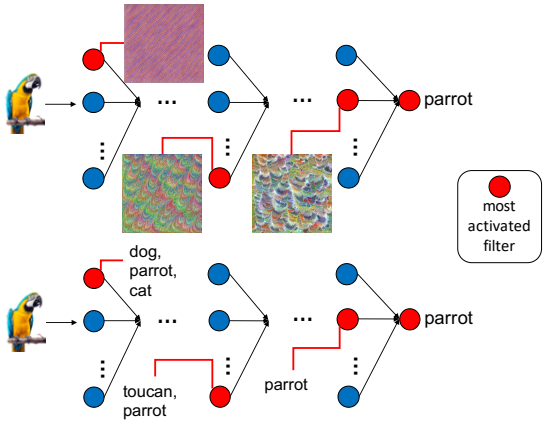

Figure 1: Explanations of Convolutional Filters. The upper part shows a visual explanation. The lower part contains an example of our tagging approach FILTag.

AIMLAI'22: Advances in Interpretable Machine Learning and Artificial Intelligence (AIMLAI@CIKM'22), October 21, 2022, Atlanta, Georgia, USA

anna.nguyen@kit.edu (A. Nguyen);
daniel.hagenmayer@student.kit.edu (D. Hagenmayer);
tobi@informatik.uni-mannheim.de (T. Weller);
michael.faerber@kit.edu (M. Färber)
0000-0001-9004-2092 (A. Nguyen); 0000-0001-5458-8645 (M. Färber)

filter embeddings. Alternatively, Network Dissection [9] uses human-labeled visual concepts to bring semantics to the convolutional layers. However, visualizations and embedding filters only explain the outcome of a model implicitly, whereas we assign explicit tags to filters which can be understood by non-experts. Most visualizations used for explaining CNNs are similar to the upper example in Figure 1, which visualizes the most activated convolutional filters. Clearly, such visualizations are difficult to understand on their own. Adding an explicit explanation such as a semantic tag (e.g. "dog," "parrot," "cat," or "toucan") as shown in the bottom example would dramatically improve the explanation, including for non-experts.

Contribution. Our contribution is threefold. First, we introduce FilTag, an automatic approach to explain the role of each convolutional filter of a CNN to non-expert humans. We use the fact that each filter is dedicated to a specific set of classes [7, 10, 11, 12]. Indeed, the idea of FilTag is to quantify how much a filter is dedicated to a class, and then tag each convolutional filter with a set of particularly important classes. The lower part of Figure 1 shows an example of what a CNN tagged in this way could look like. In that example, the rightmost filter highlighted in red plays a role in classifying parrots, whereas the filter in the middle only plays a role in classifying birds in general, as both, toucans and parrots are both birds. This filter extracts features that are specific to these classes (e.g. wings, feathers, etc.). Second, our approach can also be used to explain the classification of an individual image. In the example in Figure 1, the classification of the input image as a parrot would be explained by the union of the tags of the activated filters, which are all animals, particularly tagged with parrot. Third, FilTag is suitable to analyze classification errors. We analyze our approach with thorough experimentation using multiple CNNs, including VGG16, as well as ImageNet as a data set. All source code is available online.[1]

2. Approach

In Section 2.1, we propose a method to provide explanations based on the role of each filter in a CNN (independent from concrete input images) using our concept of filter tags. Then, in Section 2.2, we explain how a particular input image can be explained, namely in terms of the filters that it activates.

2.1. Explanations of Filters

Our explanation of filters works in two steps. In the first step, we quantify how much each filter is activated by images of each class. In the second step, we use this information to tag the filters.

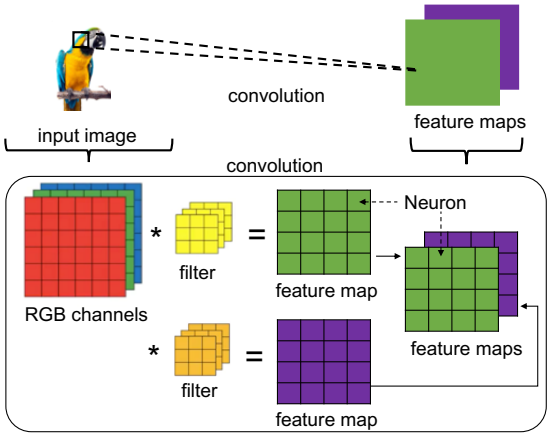

Figure 2: Terminology of a filter in a convolution.

Quantifying Filter Activations. Feature maps with high activations can be used as an indication of the importance of the preceding filter for the input image [6, 7]. Traditional explanation approaches focus on one image and therefore use the most activated feature map while our approach focuses on a set of images of the same class. Given a pre-trained CNN with a set of convolutional layers M with its respective set of filters $I_{(\cdot)}$ and a labeled data set D with labels $c \in C$ from a set of labels C, let $d \in D$ be an input image and $m \in M$ a convolutional layer. First, we collect the activations in the feature map to get the importance of the filters regarding an input image, i.e. the output in the feature map for a given filter (see terminology in Figure 2). Second, we scale these activations per layer between $[0, 1]$. In scaling the activations, we ensure that no image is overrepresented with overall high activation values. We scale the activations per layer because each layer has its specific pattern compositionality of filters. For example, the first convolutional layers detect simple patterns such as lines and edges whereas the layers, in the end, detect compositional structures which match better to human-understandable objects [7]. Let $a(m, i, d, j)$ be such a scaled activation in the jth element in the feature map calculated from image d and filter $i \in I_m$ in convolutional layer m. In order to get a total activation value per feature map, we define $\bar{a}(m, i, d) = \frac{1}{n} \sum_j^n a(m, i, d, j), 0 \leq \bar{a}(m, i, d) \leq 1$, as the arithmetic mean of the scaled activations in a feature map where n is the number of activations in the feature map. We do this for all filters $i \in I_m$ and repeat these steps for all layers $m \in M$.

Next, we use the labels as the desired explanation. Let d_c be an input image with label c. We define $z_c(m, i) = \frac{1}{|D_c|} \sum_{d_c}^{|D_c|} \bar{a}(m, i, d_c), 0 \leq z_c(m, i) \leq 1$ as arithmetic mean

[1] https://github.com/michaelfaerber/FilTag

of $\bar{a}(m, i, d_c)$ over one class c where $|D_c|$ is the number of images in class c. This way, $z_c(m, i)$ is the averaged value of all activations of the images in one class respective its filter i in layer m. Thus, we can rank the classes according to the highest averaged activation of the filter per layer which will be the decisive criterion for the labeling. We, therefore, compare the received values for each feature map. We repeat these steps for all images in D per label class.

Filter Tagging. We tag the filters according to their corresponding values received in $z_c(m, i)$ with the label of the input image class. We are interested in the feature maps with high activations of a certain class because they indicate important features associated with that class [6]. We define two methods to select those feature maps per class and per layer (because of the mentioned complexity in different layers): (i) k-best-method (choose the k feature maps with highest activation values) and (ii) q-quantile-method (choose the q-quantile of feature maps with highest activation values). These tags serve as an explanation of what the filter does. For example, in Figure 1, the leftmost activated filter has the three tags *dog*, *parrot* and *cat*, which suggests that this filter plays a role in recognizing animals.

2.2. Explanations of Individual Classifications

While previous visual methods for explaining filters are difficult for humans to understand, textual assignment can lead to unambiguous explanations (as later seen in our experiments in Figure 3). To get an explanation given an input, we assume that the tags have a better information value with the classification of the CNN if the tags match with the classification output. Therefore, we want to measure the hit of the prediction with the tags in the most activated filters. To do this, we determine the most frequently occurring labels for each image of a class according to the previous mentioned method using the metric Hits@n. Hits@n measures how many positive label tags are ranked in the top-n positions. For example, in Figure 1, the classification of the input image as a parrot is explained by its high activation of filters tagged with *parrot*.

2.3. Analysis of Classification Errors

FilTag can be used for error analysis using Hits@n. Taking misclassified input images, Hits@n indicates if the most relevant filters were activated. If Hits@n is high, we can assume that there are similar features of the misclassified class and original image. Analyzing the tags, we may find correlations in their semantics. Furthermore, linking the tags and filters to knowledge graphs such as ConceptNet [13] or FAIRnets [14] can bring more insights. ConceptNet is a semantic network with meanings of words and FAIRnets is a neural network graph with metadata about the architecture. For example, in Figure 1, if we input an image of a car but the most activated filters have tags of animals, we can conclude that the wrong filters were activated.

3. Experiment

3.1. Experimental Setup

Data Set. Following related work, we use ImageNet [16] from ILSVRC 2014 to conduct experiments on the introduced approach. This data set contains over one million images and $1,000$ possible class labels including animals, plants, and persons. Each class contains approximately $1,200$ images. We use a holdout split, using 80% of the images to tag the filters, while ensuring that there were at least 500 images from each class in the set, and the remaining 20% to test the explanations.

Baseline. We compare our approach with two state-of-the-art visualization methods in explaining neural networks. The selection of the methods was based on their focus on feature visualization. One of the methods used provided the fundamental basis of visualization of features and uses minimal regularization [15], the other method uses optimization objectives [4].

Implementation. We implemented our method in Python3 and used TensorFlow as deep learning library. The experiments were performed on a server with Intel(R) Xeon(R) Gold 6142 CPU@2.60 GHz, 16 physical cores, 188GB RAM and GeForce GTX 1080 Ti. We used pre-trained neural network models from Keras Applications. The filters of a VGG16 were explained in the experiments using the introduced method. VGG16 was used as CNN as it is frequently used in various computer vision applications. We also evaluated on VGG19 and InceptionNet but omit them due to page limitations.

3.2. Analysis of the Explanations

In this analysis, we want to study the explanations of the filters using k-best-method, with $k = 1$, in order to provide a better comparison with the state-of-the-art methods since they frequently visualize the most activated feature map. Figure 3 shows exemplary the visual explanations of the baseline methods, and the tags of our approach FilTag. As shown, the visual explanations of the baseline methods [15, 4] do not provide satisfactory comprehension. At first sight, there is not much to understand. Considering our tags, one can imagine what the visualizations display. We additionally include pictures corresponding to our tags, to show the information value compared to only visualizations of the filters. Filter 95

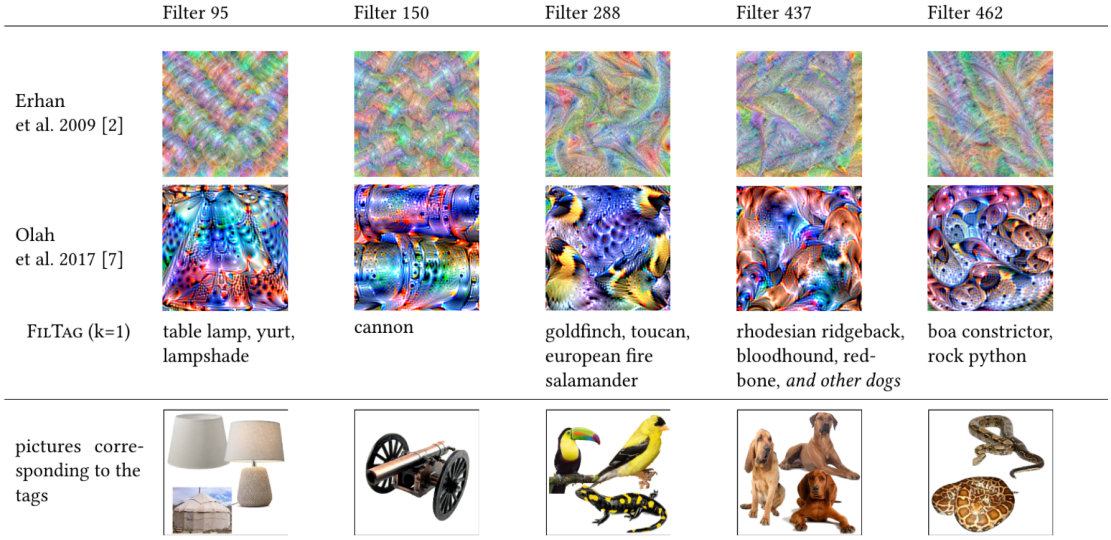

Figure 3: Comparison of filter explanations of the last conv. layer of VGG16 [1]. The visualizations of the baseline methods [15, 4] are ambiguous and difficult to interpret. Our approach FilTag allows a more precise understanding which features the filters detect. Pictures corresponding to our tags were added to show the information value.

seems to recognize a lampshade especially a trapezoidal shape. Filter 150 is only tagged with *cannon*, i.e. the filter is specific for this class. Filter 288 detects a head of a goldfinch especially with consideration of the yellow and black pattern. Filter 437 and Filter 462 recognize ears of brown dogs and the body of snakes, respectively. This information would be hard to retrieve without the tags. Even without considering the visualizations, one has a good impression of what a filter detects. For example, it is quite impressive that Filter 288 detects this black yellow pattern which we can follow from the tags *goldfinch, toucan*, and *european fire salamander*. As well, Filter 95 detects the trapezoid in *table lamp, yurt*, and *lampshade*.

In addition to comparing our method to the state-of-the-art methods in CNN explanations, we linked the tags to concepts from ConceptNet [13] to achieve a coarsening of common tags. ConceptNet is a semantic network with meanings of words. This comparison revealed that many tags have both visual and semantic commonalities (e.g., see Filter 437 in Figure 3, rhodesian ridgeback, bloodhound and redbone are all of type dog). Following this evaluation process, we manually reviewed 100 filters in the context of common visual and semantic commonalities. Here we found 88% conformance with common tags in the filters.

3.3. Impact of Hyperparameters

In the following we evaluate which impact the hyperparameters k and q have on the correlation of Hits@n and accuracy. If the labels, and thus Hits@n, do not correlate with the output of the neural network, and thus with the accuracy, then the filters have not been tagged sensibly with our approach to gain an accurate explanation. We will interpret Hits@n and accuracy with different hyperparameters k and q, respectively. In Figure 4, we compute Hits@n with the test set from ImageNet depending on k and q. We can see that Hits@n increases for increasing k, q and n. For $q = 25\%$ and $n = 50$, we even get a hit rate of 80% over all 1,000 object classes. This result shows that FilTag can be taken as a significant explanation for the classification. For example, we have observed that the class *shoji* gets the highest hit rate of 98.47% followed by the classes *slot, odometer* and *entertainment center* with

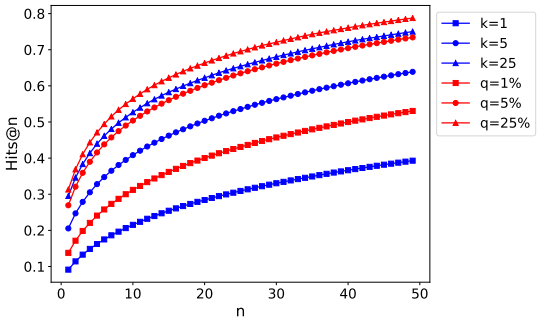

Figure 4: Hits@n with different k and q on ImageNet

also around 98%. This correlates with the likelihood of the best classes, which are exactly the same classes: *shoji* (81.22%), *slot* (92.30%), *odometer* (91.73%) and *entertainment center* (82.89%). Likewise, Hits@*n* also correlates with the accuracy of the worst classes, which are *spatula*, *schipperke*, *reel*, *bucket*, and *hatchet*. These results fit to the top-1 accuracy of VGG16 with 74, 4% for all classes. The high correlation with Hits@*n* and accuracy shows that the relevant features, labeled by our approach, are in fact detected from the images, which confirms the hypothesis that the tags are useful to generate explanations by means of our approach. However, for larger values of q we observed that the interpretability decreases because the number of tags increases for each filter. This makes it harder to find similarities between the classes. Thus, there is a trade-off between expressiveness for the classification and interpretability for the filters.

3.4. Using the Explanations

FɪlTag can be used for error analysis using Hits@*n*. Taking misclassified input images, Hits@*n* indicates if the most relevant filters were activated. If Hits@*n* is high, we can assume that there are similar features of the misclassified class and original image. Analyzing the tags, we may find correlations in their semantics.

Figure 5 (a) shows an image of the class *mortarboard* in ImageNet. Using VGG16, the class *academic gown* is predicted with a confidence of 83.8%, while the actual class *mortarboard* is predicted with a confidence of only 16.2%. Considering the image, we notice that both objects are part of this image, making this result reasonable. Reviewing the activated filters, we observe that filters tagged by FɪlTag with the tag *mortarboard*, as well as with the tag *academic gown*, are usually activated. As a result, we can verify that features are extracted from these two classes and used for prediction. This allows to give non-experts an understanding of the reason for the misclassification, as often features of the other class are extracted from this image. Likewise, we can use the information to increase the number of images in which the mortarboard is the actual class but not in the main focus of the image, in order to continue learning the network to make the predictions more accurate.

Figure 5 (b) shows an image from the class *computer*. This image is classified by VGG16 as *cash machine* with a probability of 99%. Looking at the tagged filters, filters of the tags *cash machine* are mostly activate, followed by *screen, CD player*, and *file*. Considering Figure 5 (b) and having knowledge about the other images of the class *computer* in ImageNet, the reason this image is not assigned to this class becomes clear. Generally, frontal images of a computer were used for the *computer* class for learning. However, this image does not correspond to the same distribution. Thus, it is difficult for the neural network to assign it correctly. Moreover, it is an old computer, whereas the other images in ImageNet generally represent rather modern computers. In order to classify this image correctly, further images showing old computers from the side have to be included to change the distribution and train the VGG16 to classify this image correctly.

(a) Mortarboard (b) Computer

Figure 5: Example images from ImageNet

4. Conclusion

We have introduced FɪlTag, an approach to provide human-understandable explanations of convolutional filters and individual image classifications. These tags can be used to query and identify specific filters that are relevant for feature detection. In contrast to state-of-the-art explanations, our approach allows for explicit, non-visual explanations which are more understandable for non-experts.

A limitation of our approach is the use of the class labels as tags to describe the filters. As a result, filters are not described in terms of specific objects such as ears, wings, or legs. We would like to address this limitation in the future by using ConceptNet and other knowledge bases to identify commonalities of the tags and thus add specific object descriptions to the filters.

References

[1] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015.

[2] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, 2016, pp. 2818–2826.

[3] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digital Signal Processing 73 (2018) 1–15.

[4] C. Olah, A. Mordvintsev, L. Schubert, Feature Visualization, Distill (2017).

[5] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, A. Mordvintsev, The Building Blocks of Interpretability, Distill (2018).
[6] F. Hohman, H. Park, C. Robinson, D. H. P. Chau, Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations, IEEE transactions on visualization and computer graphics 26 (2019) 1096–1106.
[7] M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, in: Computer Vision - ECCV 2014, volume 8689 of *Lecture Notes in Computer Science*, 2014, pp. 818–833.
[8] R. Fong, A. Vedaldi, Net2Vec: Quantifying and Explaining How Concepts Are Encoded by Filters in Deep Neural Networks, in: Conference on Computer Vision and Pattern Recognition, CVPR 2018, 2018, pp. 8730–8738.
[9] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network Dissection: Quantifying Interpretability of Deep Visual Representations, in: Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017, pp. 3319–3327.
[10] R. B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, in: Conference on Computer Vision and Pattern Recognition, CVPR 2014, 2014, pp. 580–587.
[11] K. Simonyan, A. Vedaldi, A. Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, in: 2nd International Conference on Learning Representations, ICLR 2014, 2014.
[12] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. A. Riedmiller, Striving for Simplicity: The All Convolutional Net, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015.
[13] R. Speer, J. Chin, C. Havasi, ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 4444–4451.
[14] A. Nguyen, T. Weller, M. Färber, Y. Sure-Vetter, Making Neural Networks FAIR, in: Knowledge Graphs and Semantic Web - Second Iberoamerican Conference and First Indo-American Conference, KGSWC 2020, volume 1232 of *Communications in Computer and Information Science*, 2020, pp. 29–44.
[15] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing Higher-Layer Features of a Deep Network, Technical Report, Univeristé de Montréal (2009).
[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252.