Principal Feature Visualisation in Convolutional Neural Networks [*]

Marianne Bakken[1,2][0000−0003−4958−8194], Johannes Kvam[1],
Alexey A. Stepanov[1], and Asbjørn Berge[1]

[1] SINTEF Digital, Forskningsveien 1, 0373 Oslo, Norway
marianne.bakken@sintef.no
https://www.sintef.no/en/
[2] Norwegian University of Life Sciences (NMBU), 1432 Ås, Norway

Abstract. We introduce a new visualisation technique for CNNs called Principal Feature Visualisation (PFV). It uses a single forward pass of the original network to map principal features from the final convolutional layer to the original image space as RGB channels. By working on a batch of images we can extract contrasting features, not just the most dominant ones with respect to the classification. This allows us to differentiate between several features in one image in an unsupervised manner. This enables us to assess the feasibility of transfer learning and to debug a pre-trained classifier by localising misleading or missing features.

Keywords: Visual explanations, deep neural networks, interpretability, principal component analysis, explainable AI

1 Introduction

Deep convolutional neural networks (CNNs) have had a significant impact on performance of computer vision systems. Initially they were used for image classification, but recently these methods have been used for pixel-level image segmentation as well. Segmentation methods are able to capture more information, but require significantly more expensive labelling of training data. Moreover, classification (bottleneck) networks are still used for many applications where the problem can't be formulated as a segmentation task or pixel-wise labelling is too expensive.

One of the main issues with bottleneck networks is that they provide no visual output, that is, it is not possible to know what part of the image contributed to the decision. As a consequence, there is a demand for methods that can help visualise or explain the decision-making process of such networks and make it understandable for humans.

A range of visualisation and explanation methods have been proposed. Class Activation Mapping, e.g. [10], is a computationally efficient way to show the support of a class in the input image, but the resulting heatmap is quite coarse.

[*] Funded by The Norwegian Research Council, grant no. 259869

Gradient-based methods like [3] give a more localised response, but require back-propagation through the whole network, and is very sensitive to edges and noise in the input image.

All these methods operate in a *supervised* manner on one category or feature at a time. In contrast, our method is *unsupervised* and visualise several categories or features in one pass. It can be applied directly to any bottleneck network without any additional instrumentation.

Our approach provides a visualisation that maps the principal contrasting features of a batch of images to the original image space in a single forward pass of the network. We target bottleneck networks, such as image classifiers, and use a singular value decomposition on the feature map of the layer we wish to visualise, e.g., the final convolutional layer, to extract the principal contrasting features for a batch of images. These features are then interpolated back to the original image space, and the activation maps of the earlier layers are used to weight the resulting feature visualisation. An overview of the method is shown in Fig. 1.

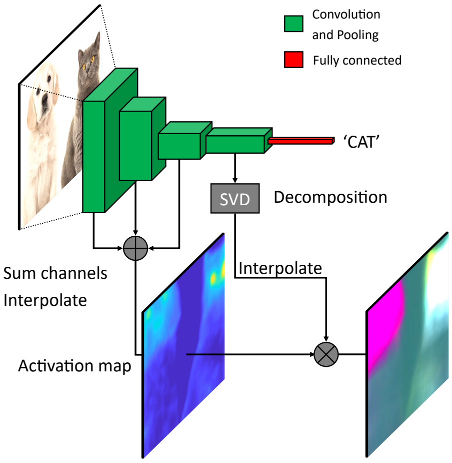


Fig. 1. Overview of our Principal Feature Visualisation (PFV)method.

The main advantages of our method are:

1. Contrast: Per-pixel visualisation of the principal contrasting features.
2. Lightweight: Requires a single forward pass of the original unmodified network, using only intermediate feature maps.
3. Easy to interpret: suppresses non-relevant features.
4. Unsupervised: No additional input or prior knowledge about image classes is required.

We show how the advantages of the method allow it to be used as a tool for debugging misclassification and assessing the feasibility of transfer learning in Section 5.

Our code is publicly available at `https://github.com/SINTEF/PFV`.

2 Related Work

Several categories of methods to interpret CNNs have been proposed. We focus on the methods that provide a visual human-understandable representation, in particular methods that relate the attention of the network back to the original image space in the form of masks or heatmaps.

One way of attributing classifier decision to location in the input is to perform simple perturbations (e.g. occlusion) to the input [16,13] and make a heatmap per class based on change in the output. Similarly, more advanced methods for perturbation of the input image has been proposed [9,2]. The drawback of these methods is that the number of required forward passes is proportional to the number of classes and resulting heatmap resolution.

Other methods focus on localisation of semantically meaningful concepts in the input. For instance by extracting and clustering superpixels, and then compute the saliency as a ranking [7] over these extracted "concepts" [6]. Network dissection is another direction [4], where the response in network hidden units (convolutional layers) are scored according to a predefined set of visual concepts.

Gradient-based visualisation is a group of methods that provide more localised responses and are widely cited in literature. The simplest form of this is to compute the partial derivatives of the output with respect to every input pixel [13]. Several additions to this principle, for instance DeepLIFT [12], Guided Backpropagation [15] and Layer-wise Relevance Propagation (LRP) [3], has improved the localisation and visual appeal. However, as showed through simple sanity checks in [1], many of these methods rely too much on information from the input image, and are actually insensitive to changes in the model. Additionally, they can require a lot of instrumentation, such as special types of layers and separate training of hyperparameters.

Class Activation Mapping provides a direct mapping from the class score to the activations from the forward pass of a CNN. The original work in [5] required a special network architecture, but Grad-CAM [10] provided a more general way to compute the mapping by backpropagation from the class score to the last convolutional layer (not all the way back to the inputs as pure gradient-based methods). Grad-CAM passes the sanity checks in [1], but gives a less localised response than gradient-based methods, and still requires backpropagation from each class to produce responses from multiple classes or objects. Our approach use the activations from the forward pass in a similar manner as Grad-CAM, but rather than computing a mapping through backpropagation, we do a simple unsupervised learning during the forward pass.

Some methods include counter-evidence to give a richer explanation. Grad-CAM and LRP for instance, suggest using negative gradients in addition to the positive ones to show evidence against a class. In [17], a top-down attention propagation strategy is proposed, that performs backpropagation of both positive and negative activations to create a contrasting visualisation. Our method provides an inherent contrast, and does not need to treat this specifically.

There are also several methods that apply clustering or spectral techniques for model explanation. One such method [8] applies spectral clustering on a set of relevance maps computed with LRP, and performs eigengap analysis and t-SNE visualisation to identify typical prediction strategies. This requires several steps of processing, and is applied on one class at a time. Another work [11] uses Eigenspectrum analysis of the feature maps in neural networks to optimise neural architectures and understand the dynamics of network training. Our approach uses spectral information in a similar manner to these approaches, but to our knowledge is the first one to project this type of information back to image space in one pass.

Compared to existing explanation methods, we aim for an approach that is simple to execute, that depends on activations from the network itself rather than edges in the input image, and can highlight the contrast between several features and classes in one pass.

3 Principal Feature Visualisation

3.1 Method description

Our goal is to obtain a low-dimensional representation of the feature space of feed-forward bottleneck networks which can be mapped to the original image space. Such a visualisation should be achieved in an efficient manner by using a single forward pass of the network, without any additional instrumentation.

Principal component analysis (PCA) projects a signal onto a set of linearly uncorrelated variables (principal components) ranked by the amount of variance explained in the original signal. Conveniently, the projection of features onto these components introduces an implicit measure of contrast, due to the orthogonality of the components.

In brief, our method decomposes a feature map into its principal contrasting features for a batch of images. This is accomplished by extracting principal components through singular value decomposition. The decomposed feature map is then interpolated back to the original image space, where we use the activation maps in the preceding layers as spatial weighting. An overview of the method is shown in Fig. 1, and we describe it in detail below.

Consider a CNN with N convolution and pooling layers. For each layer l a feature map F_l is an $n_B \times n_{c,l} \times n_{x,l} \times n_{y,l}$ matrix, where n_B is the number of images passed through the layer (batch size), $n_{c,l}$ number of channels and $n_{x,l}, n_{y,l}$ is the spatial size of that layer. We denote by $(n_{x,0}, n_{y,0})$ the size of original input images.

Suppose we want to visualise the last convolutional layer N. Our method proceeds as follows. First, for each intermediate F_l we calculate activation maps for each image in batch

$$A_l^b(i,j) = \sum_{c=1}^{n_{c,l}} F_l(b,c,i,j), \quad b \in \{1,\dots,n_B\} \tag{1}$$

We then compute the total activation map A^b for each batch image as a sum of upsampled activation maps for each layer. That is

$$A^b = \sum_{l=1}^{N-1} \mathrm{P}(A_l^b; n_{x,0}, n_{y,0}), \tag{2}$$

where $\mathrm{P}(A_l^b; n_{x,0}, n_{y,0})$ denotes upsampling of A_l^b back to original input image size.

Now consider the feature map F_N of the final layer. Our approach is to use PCA to decompose the features for visualisation. First, we reshape F_N to a $n_{c,N} \times (n_B \cdot n_{x,N} \cdot n_{y,N})$ matrix. In this way we treat each per-pixel channel response as a separate observation. We denote this reshaped matrix as F' and centre it by subtracting mean values:

$$F' = F' - \bar{F}' \tag{3}$$

Then we find the principal feature responses by decomposing F' using singular value decomposition as

$$F' = USV^T, \tag{4}$$

where S is a diagonal matrix containing the singular values and U is the decomposition of F' into the space described by the eigenvectors V.

The principal components are then the sorted columns of the following matrix

$$F_{\text{PCA}} = US = [\mathbf{d}_1 \quad \dots \quad \mathbf{d}_r] \tag{5}$$

For visualisation convenience, we choose a subset of F_{PCA} columns $\{\mathbf{d}_1, \dots, \mathbf{d}_{n_d}\}$. For the rest of the paper we assume $n_d = 3$, which allows us to visualise F_N by mapping $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ to red, green and blue channels. We denote by D_N a matrix consisting of these columns

$$D_N = [\mathbf{d}_1 \quad \mathbf{d}_2 \quad \mathbf{d}_3] \tag{6}$$

By reshaping D_N back to $n_B \times 3 \times n_{x,N} \times n_{y,N}$ size and treating each batch image as a separate D_N^b we can upsample D_N^b back to the original size $(n_{x,0}, n_{y,0})$. We use the activation map A^b to weight the upsampled D_N^b and normalise the result as follows

$$V^b = \text{normalise}\left(A^b \circ \mathrm{P}(D_N^b; n_{x,0}, n_{y,0})\right), \tag{7}$$

where $\circ$ is an element-wise product and P is upsampling operator. Note that the colours in the final images V^b are relative to the processed batch.

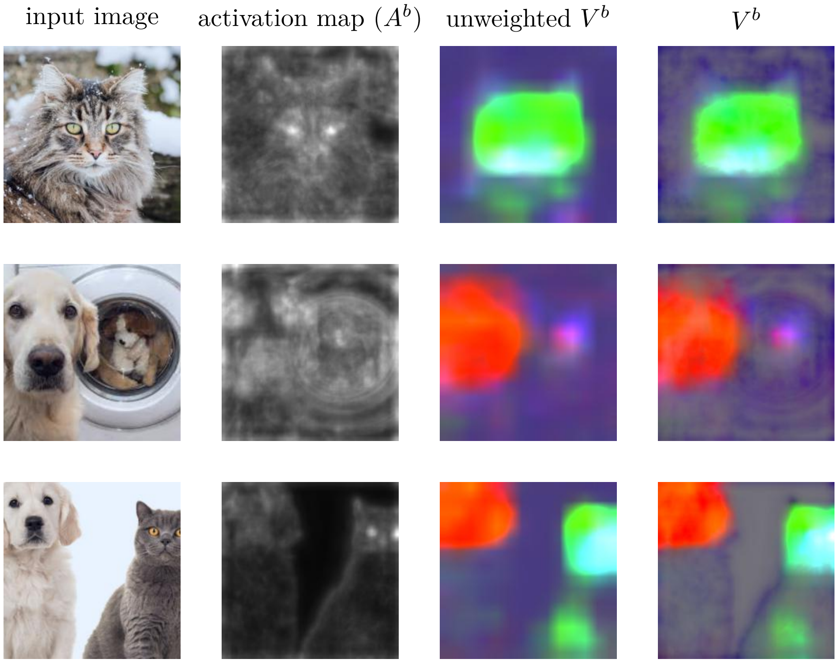


Fig. 2. Variations of our Principal Feature Visualisation method applied on a pre-trained bottleneck CNN (VGG16) and a batch of dog and cat images. The activation map A^b is used to weight the feature map. Colours represent the strongest principal features of the batch and their location in image space. Best viewed in colour.

3.2 VGG Example

We illustrate the properties of our method with a simple example of a few dog and cat images and a VGG16 network [14] pre-trained on ImageNet.

First, we show the final visualisation V^b together with two intermediate steps: the activation maps A^b, and *unweighted* V^b from upsampling directly without weighting. V^b was computed with a forward pass on a batch of six images of dogs and cats. The intermediate activation maps A^b_l were extracted before each max pool layer, and the feature map of the final layer, F_N, was extracted before the last max pool layer. We used bilinear interpolation for upsampling. The results are shown in Fig. 2. For this batch, the principal feature maps assign different colour channels to dogs, cats and background. Studying the intermediate steps, we see that the principal feature map without weighting shows more response from the channel in the background. The weighting with earlier activation maps thus enhances the *strongest* features, while the principal components provides *contrast* between different features.

Second, we illustrate how the visualisation depends on the composition of the input batch. Fig. 3 shows our method applied on different single-image input batches. The colours now represent different features within that image only. For

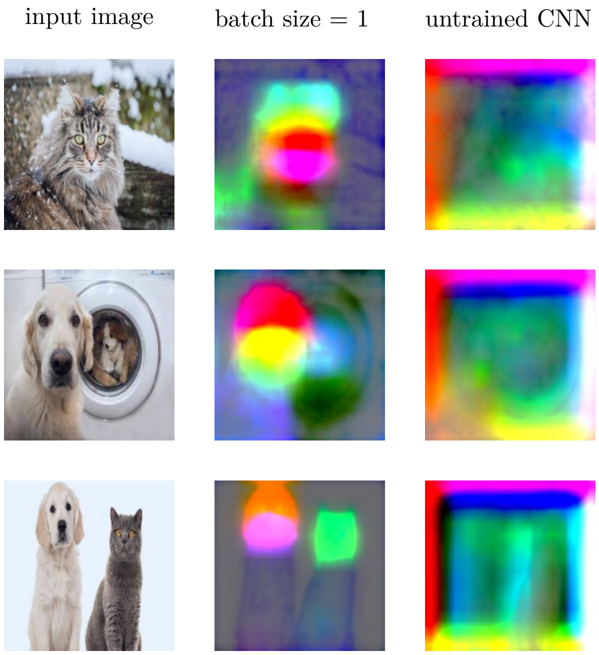


Fig. 3. Batch size illustration and sanity check with untrained network. Second column shows the visualisation of single-image input batches. The colours now represent different high-level features like ears and nose rather than the class-level features in Fig. 2. Third column shows a simple sanity check: the visualisation of an untrained network of randomly initialised weights. The result is completely different, as expected. Best viewed in colour.

the image with two objects, there is still some class-related contrast. This brief example indicates that batch composition can be used deliberately as a tool to control the contrast in the visualisation and tailor it to any application. More examples of this are shown in Section 5.2 and supplementary material.

In order to be useful for model debugging, a visualisation method should be sensitive to the model parameters. We perform a simple parameter randomisation test as suggested in [1], by running our method on a randomly initialised untrained version of the network. As seen in Fig. 3, the resulting visualisation of the random model is visually very different from the pre-trained one. This indicates model sensitivity in our visualisation, which can be used for debugging the training process.

4 Comparison with other methods

We compare our method (PFV) with Grad-CAM [10] and Contrastive Excitation Backprop (c-EBP) [17] on VGG16 pre-trained on ImageNet. We use a batch of images that is not included in ImageNet, but contains objects of ImageNet categories. A few examples are shown in Fig. 4, where we have used the top-3 predicted classes as targets for Grad-CAM and c-EBP.

Grad-CAM and c-EBP are supervised methods based on backpropagation, that generate a heatmap conditioned on the predicted class. Consequently, these methods highlight evidence for a particular class, and suppress sources that do not contribute to the decision. Contrastive EBP approximates the probability that a given image pixel contribute positively or negatively to the decision. When the target classes are unknown and we simply specify them as the top-k predictions, these methods require a potentially large number of backward passes to describe the feature diversity in the image.

In contrast, our PFV is an unsupervised method calculated based on a single forward pass, that highlights the principal contrasting features in a batch of images. As our method is based on principal components which form an orthogonal basis where one component cannot explain another, it focuses on feature variance instead of evidence for a decision. The colours of PFV represent different features, with no direct connection to the final classification. However, by performing PFV on a batch of images, e.g. the three images in Fig. 4, colours are consistent across the batch and show which objects that have similar features.

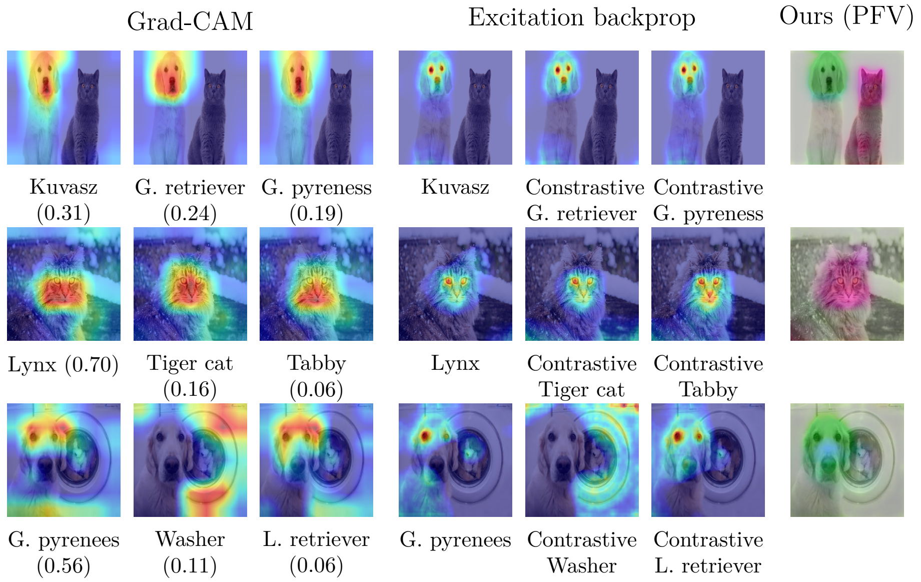


Fig. 4. Comparison of GradCAM, Constrastive Excitation Backprop (c-EBP) and PFV on VGG16 pre-trained on ImageNet. Grad-CAM and c-EBP results are shown for the top-3 predicted classes. PFV results is for a batch of the three images shown. Colours represent heatmaps for Grad-CAM and c-EBP, and principal features for PFV. Best viewed in colour.

5 Applications

In this section we apply our method to two use-cases: debugging misclassified examples by localising misleading and missing features in the input image; and ad hoc prediction of the success of transfer learning with a pre-trained network.

5.1 Debugging classification errors

When a network fails to classify an image correctly, it can be hard to know what part of the image is to blame. We show how our method can be used to identify misleading or missing features and their location in the image by comparing principal feature maps of incorrectly and correctly classified samples.

To do this, we apply PFV on an example task: dog breed classification. There are 120 dog breeds among the 1000 categories of the ImageNet dataset, and the features of the pre-trained VGG16 network should therefore be well suited for this task. We ran prediction on a handful of images of the class “English Springer Spaniel” not present in the original dataset, and identified the failed samples. It turns out that all the failed samples show dogs in water, and we want to examine why they fail. Is it because of the water, occlusion of body parts, or something else?

We applied the following procedure: For each misclassified sample, PFV was applied on a batch of six correctly classified samples; three of the true class and three of the mistaken class. To aid the comparison of the PFV images, we also plot the distribution of red, green and blue in the foreground of the PFV image, i.e., the three strongest principal components.

Figure 5 shows the result of running PFV on two batches of images containing two misclassified images: Batch A (“Springer spaniel” misclassified as “goose”) and Batch B (“Springer spaniel” misclassified as “Sussex Spaniel”). To identify missing or misleading features, we compare the PFV distributions of the other images in the batch with the failed sample, and look for the location of the colours with large deviation. In the left case (Batch A), the misclassifed sample has a red component on the head as in the true class “springer”, but is missing the red component on the rest of the body. It also has a strong green component on the body as in “goose”. In the right case (Batch B), the misclassified sample is missing the strong green component located on the white fur in front in the “springer” image, and the PFV distribution is more similar to that of “sussex spaniel“, which has no white fur. For both cases, the location of the missing features reveal that the failed classifications can most likely be blamed on body parts occluded by water.

This example shows that our method can be used to localise missing or misleading features, because it highlights the *contrasting* features within a batch, not just the most dominant features from the classification.

5.2 Transfer learning

Transfer learning is often applied when there is limited training data available to train a deep neural network from scratch. In this section we show that it is

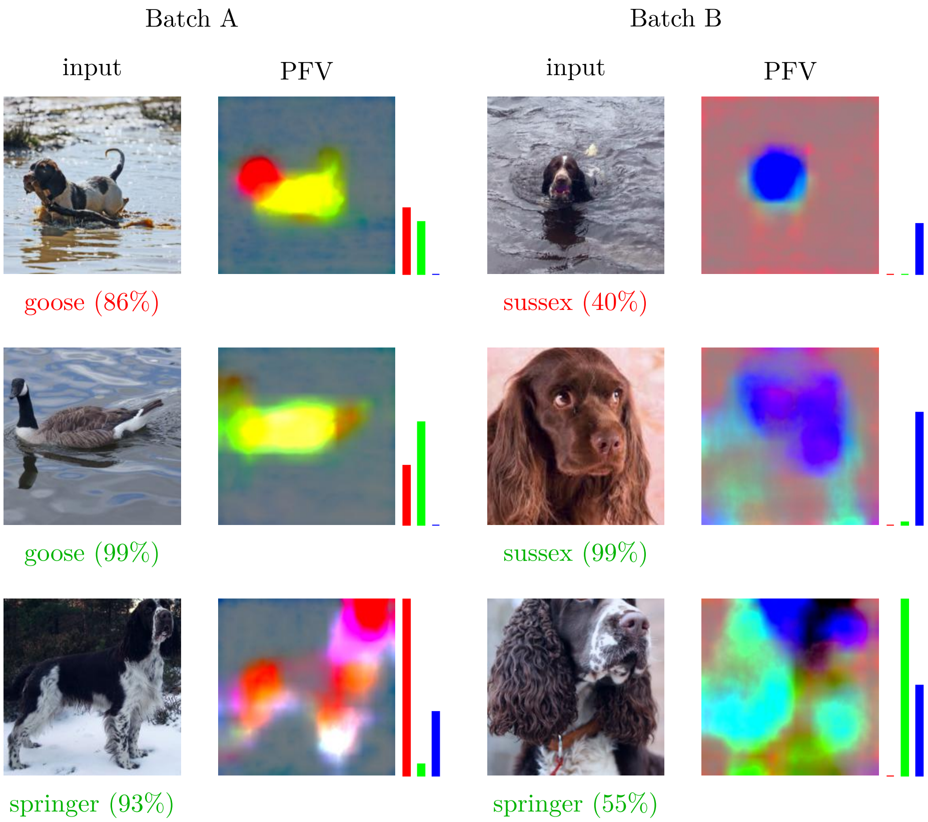


Fig. 5. Principal Feature Visualisation (PFV) on misclassified samples compared to correctly classified samples. In the first row, the two input images are of the category "English Springer Spaniel", but has been classified as "goose" and "Sussex Spaniel". In the second and third row, the input images are examples from the two different PFV batches. Bars show the distribution of red, green and blue foreground pixels of the PFV image. The colour encoding is not consistent because the method is applied on two different batches, and hence the principal vectors are different. Best viewed in colour.

possible to predict the success or failure of transfer learning on a new dataset by visualising the principal features of the pre-trained network on images from this dataset.

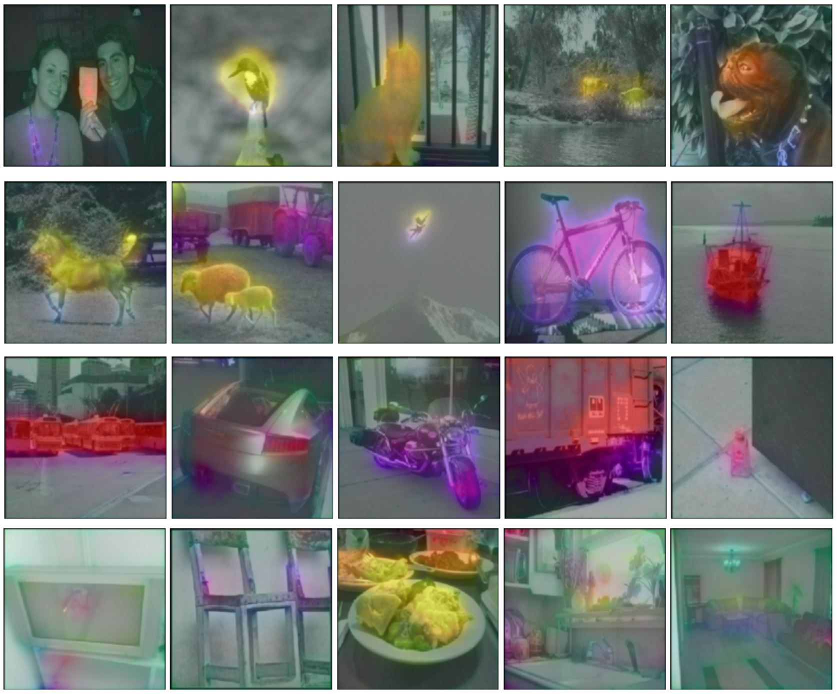


Fig. 6. Initial principal feature visualisation of VGG16 features on the Pascal VOC2012 dataset. The dataset contains 20 classes, features are visualised for a random example from each class. Similar colours indicate similar features. Best viewed in colour.

We analyse the features of VGG16, pre-trained on ImageNet, applied to the Pascal VOC2012 dataset.

Initially, we randomly sample one image from each of Pascal VOC2012's 20 classes and form a batch of these images. We then apply PFV and visualise the principal contrasting features of this batch, shown in Fig. 6. For simplicity, the feature visualisations are shown as an overlay to a grey scale version of the input image. As the images are quite dissimilar, decomposing the features of the images into three principal features, only gives us a coarse indication of which examples contain similar feature sets. Based on this visualisation we observe that the animal classes appear to have similar features, while vehicles and bicycles appear to have a different set of features. Interestingly, we also see observe that there are only weak feature responses for chair, sofa and potted-plant, while for the class dining-table, the main responses are from the objects on the actual table.

To further investigate the difference between the features in the animal categories, that have similar colours in Fig. 6, we randomly sample new batch of images from these categories. This time, we sample 4 random images from each of the categories: “dog”, “cow”, “cat”, “horse” and “sheep”. We then again apply PFV to find the principal contrasting features for this batch of images, shown in Fig. 7. Note again, that the colours in the images are relative to each batch. As the class variation in this batch of images is lower than in the initial experiment, we observe that we obtain a finer decomposition. Here we see that cats and dogs become more clearly separated from the other classes. The other three classes; cows, horses and sheep, does appear to contain similar features. In addition, one example from the “dog” class and one from the “cat” class appear as outliers, which might be due to the images being difficult examples or that ImageNet contains multiple cat and dog breeds.

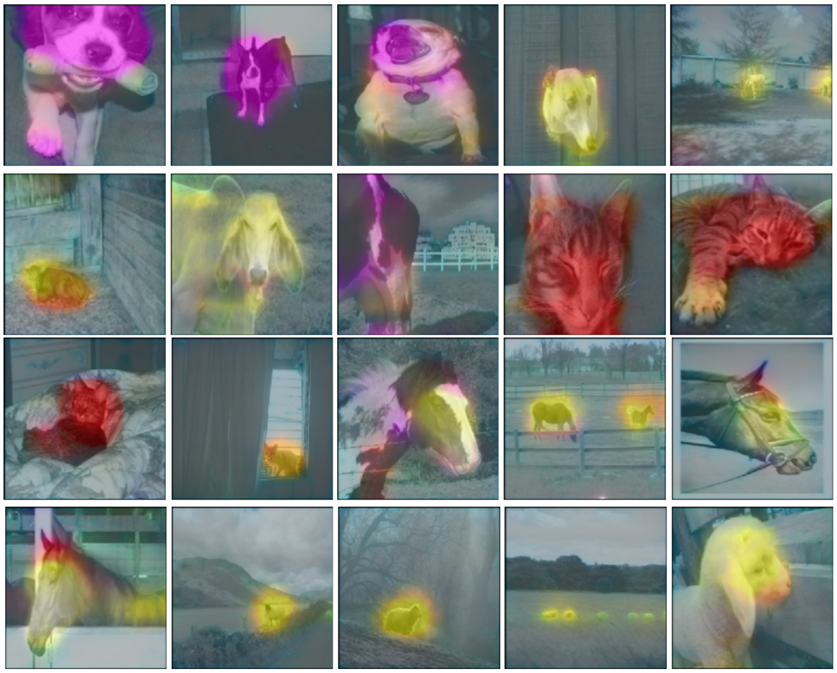
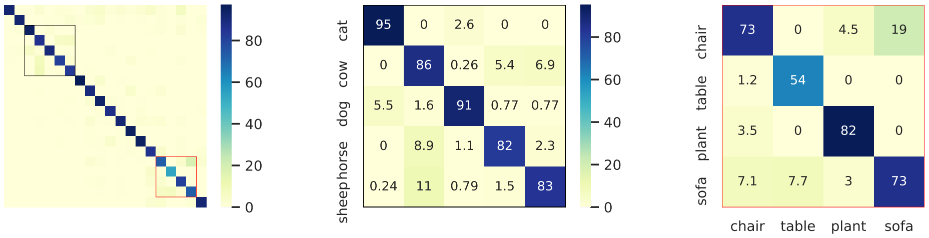


Fig. 7. Principal feature visualisation of VGG16 features on the Pascal VOC2012 dataset for the classes; “dog”, “cat”, “cow”, “horse” and “sheep”, with a batch of four random examples sampled from each class. Similar colours indicate similar features. Best viewed in colour.

Based on this analysis we hypothesise that in a fine-tuned model using VGG16 ImageNet features, we expect little confusion between the cat and dog class, a more pronounced confusion between the “horse”, “cow” and “sheep” classes. In addition, the weak feature responses for classes “chair”, “diningtable” and “sofa”, indicate an overall poorer performance in the detection of these classes.

To check this hypothesis we fine-tune VGG16 pre-trained on ImageNet on the Pascal VOC2012 dataset. We retrain only the final fully connected layer (the classifier), the rest of the network (i.e., all convolutional layers) is kept fixed during training. For simplicity we only select images containing one class per image, to be able to use a standard cross-entropy loss in the optimisation. We train until the validation loss stops decreasing and investigate the final performance in terms of a confusion matrix. The confusion matrix is shown in Fig. 8.



cat	95	0	2.6	0	0
cow	0	86	0.26	5.4	6.9
dog	5.5	1.6	91	0.77	0.77
horse	0	8.9	1.1	82	2.3
sheep	0.24	11	0.79	1.5	83

	chair	table	plant	sofa
chair	73	0	4.5	19
table	1.2	54	0	0
plant	3.5	0	82	0
sofa	7.1	7.7	3	73

Fig. 8. Confusion matrix for the validation set for a VGG16 network, after fine-tuning on Pascal VOC2012. Left, overall view of confusion matrix. Middle, confusion between classes “cat”, “cow”, “dog”, “horse” and “sheep. Right, confusion between classes “chair”, “table”, “plant” and “sofa”. Best viewed in colour.

The worst performing categories are of the classes “dining table”, “sofa”, and “chair”. We also observe that “cow” is significantly confused with classes “horse” and “sheep”. These observations suggest that such a feature visualisation strategy can give an intuition about when pre-training will be beneficial and when it might fail.

6 Conclusion

We have presented a method for visualising the principal contrasting features of batch of images during forward pass of a bottleneck CNN. Our approach has several advantages over related methods, namely that it combines low overhead with intuitive visualisation, and doesn’t require any user input or modification of the original CNN. We have shown how these advantages allow us to interpret the performance of CNNs in two common settings: debugging misclassification and predicting the applicability of transfer learning.

Our code is available at `https://github.com/SINTEF/PFV`.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems. vol. 2018-Decem, pp. 9505–9515. Neural information processing systems foundation (10 2018), http://arxiv.org/abs/1810.03292
2. Agarwal, C., Schonfeld, D., Nguyen, A.: Removing input features via a generative model to explain their attributions to an image classifier's decisions (2019), http://arxiv.org/abs/1910.04256
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE **10**(7), 1–46 (2015). https://doi.org/10.1371/journal.pone.0130140
4. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
5. Bolei Zhou and Aditya Khosla and Agata Lapedriza and Aude Oliva and Antonio Torralba: Learning Deep Features for Discriminative Localization. CVPR **2016**(1), M1–M6 (8 2016). https://doi.org/10.5465/ambpp.2004.13862426, http://journals.aom.org/doi/10.5465/ambpp.2004.13862426
6. Ghorbani, A., Wexler, J., Zou, J., Kim, B.: Towards Automatic Concept-based Explanations. NeurIPS (2 2019), https://github.com/amiratag/ACEhttp://arxiv.org/abs/1902.03129
7. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., Sayres, R.: Interpretability beyond feature attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In: 35th International Conference on Machine Learning, ICML 2018. vol. 6, pp. 4186–4195 (2018)
8. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.R.: Unmasking Clever Hans predictors and assessing what machines really learn. Nature Communications **10**(1), 1–8 (2019). https://doi.org/10.1038/s41467-019-08987-4
9. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should i trust you?" Explaining the predictions of any classifier. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. vol. 13-17-Augu, pp. 1135–1144 (2016). https://doi.org/10.1145/2939672.2939778
10. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. International Journal of Computer Vision **128**(2), 336–359 (2020). https://doi.org/10.1007/s11263-019-01228-7, http://gradcam.cloudcv.org
11. Shinya, Y., Simo-Serra, E., Suzuki, T.: Understanding the effects of pre-training for object detectors via eigenspectrum. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
12. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3145–3153. PMLR, International Convention Centre, Sydney, Australia (06–11 Aug 2017), http://proceedings.mlr.press/v70/shrikumar17a.html
13. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. 2nd International Con-

ference on Learning Representations, ICLR 2014 - Workshop Track Proceedings pp. 1–8 (2014)

14. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR **abs/1409.1** (2014). https://doi.org/10.1016/j.infsof.2008.09.005, http://arxiv.org/abs/1409.1556
15. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. In: 3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings (2015)
16. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **8689 LNCS**(PART 1), 818–833 (2014). https://doi.org/10.1007/978-3-319-10590-1_53
17. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-Down Neural Attention by Excitation Backprop. International Journal of Computer Vision **126**(10), 1084–1102 (10 2018). https://doi.org/10.1007/s11263-017-1059-x, http://arxiv.org/abs/1608.00507