Activation Maximization and Visualization of Features of Deep Convolutional Neural Networks

Anar Amirli
Universität des Saarlandes

April 12, 2021

Abstract

Convolutional Neural Networks (ConvNets) has achieved impressive results in a variety of image recognition tasks. Apart from the quantitative evaluations, their architecture has been treated as a black box until recently. In this term paper, we address one of the methods of visualization in order to have better reasoning about the capabilities of the models learnt by using deep ConvNets. The main goal of this term project is to have a better interpretation of the high levels feature representations attained by such models, as true posterior are intractable for high layers. To that extend, we adapted a gradient-based optimization technique called activation maximization to ConvNets for reconstructing the feature space characterized by the filters in high layers. Using this method we carried out several experiments to demonstrate how to make qualitative inferences about such models in terms of their capability to learn the feature space captured by training samples.

1 Introduction

Since they were first introduced by Lecun et al. (1998), deep ConvNets has demonstrated excellent performance at various image recognition tasks from classification to segmentation. As ConvNets quickly went to become the main choice of architecture for image recognition problems, it underwent various improvements and extensions ensuring to deliver outstanding performance on even more challenging and broader visual recognition problems (Hendricks et al., 2016; Gao et al., 2016). However, despite all these seminal breakthroughs, most of the success was reduced to trial-and-error in the early stages as these models were treated as a black box until recently. The problem of having very little insight into the internal computation and visual reasoning captured by deep ConvNets made it significantly challenging to develop better architectures. On the other hand, reasoning techniques such as visualization of the model at different layers, can assist to sidewalk these challenges granting more intuition about the model's operation which would eventually assist researchers to simply craft better architectures.

Although initially in practice there was still some intuition about ConvNets by visualizing the feature activation of a given layer. However, that explanation methods are only limited to initial layers and not desirable for the mid and high layers as the size of the feature map at high levels are small to project any meaningful pixel space. To that extend, researchers began to scrutinize the ways of learning middle and high-level feature representations to capture the information about image structure at a different level in order to have better qualitative reasoning about the capabilities of the developed models. One of the early examples of such attempts was Erhan et al. (2009), visualizing the high layer features by finding an image that maximizes the neuron activity of interest for Deep Belief Network (DBN) (Hinton et al., 2006). Later, the problem of ConvNets visualization was addressed by Zeiler et al. (2011) proposing a multilayer hierarchical architecture, Deconvolutional Networks (DeconvNets), which reconstruct the visualization of each layer from its output through series of convolutional sparse encoding and max pooling. Using DeconvNets Zeiler and Fergus (2014) introduced a feature generalization to find out which input stimuli reveal which individual feature maps at any layer. Simonyan et al. (2013) proposed two visualization techniques, based on the computing gradient of the class score with respect to the input image, to show what elements of the image account for which class of interests. The first approach is generating an image that maximizes the class score in classification layer of the pre-trained ConvNets, thus revealing general feature space for each class. The second, passing a single back-

propagation through the ConvNets computes an image-specific class saliency map to highlight the areas of the image discriminative to the given class.

In this term paper, we extend the original method, performing maximization of filter's activation, introduced by Erhan et al. (2009) and adapt it to ConvNets. This technique ensures to reconstruct pixel space that approximately generalizes the feature space representation of images that would receive the highest stimuli for a given filter. The main goal is to explore what feature space representation that the filters in any arbitrary layer of ConvNets account for. That method enables us to examine the proposed models in ways, such as qualitatively assessing the quality of the training process or simply the ability of the architecture itself, to have more insight into their performance. Such a technique is quite straightforward and efficient to apply. The main disadvantage is that finding the optimal stimulus for each filter by performing gradient descent requires the careful initialization of parameters. Furthermore, in addition to the visualization with activation maximization, we also perform simple sensitivity analysis of the occluded input images by assessing most activated filters to reveal which parts of the images are important for feature space.

2 Approach

2.1 Maximization of Activation

The main idea here is to calculate gradient updates for noise image maximizing the activation of a filter to push it to approximate the feature representation revealed by that filter. To put it simply, in other words, we search for an input pattern that maximizes the activation of a given filter. This approach allows us to find what the filters at any layer of the model is responsible for, thus, revealing how distinctive they actually are.

The suggested method relies on the intuition that the activation function of a filer is a function of the input space, thus input patterns must be somehow proportional to the filters. To that end, we assume that feature representation to which the filters are sensitive can be a good estimation of what actually filters are responsible for. A very simple way to do this is to choose a sample image either from train or test set and find which filters are activated the most. Later reconstruct the pixel space using gradient updates for noise image by maximizing those filters. However, there is still one main shortcoming that there is not an easy way to automatically choose the number of samples to be used for each filter. That is because there is no efficient way to 'combine' feature space of that samples to see which samples have in common. For this, we will maximize a filter activation in a layer using a single sample. Moreover, it is also not easy to determine by inspection what subset of image vector leads to the high activation. To address the latter, in this term paper we did a simple sensitivity analysis of the occluded images to reveal which parts of the sample vector is important for activation of filters.

Needless to say, instead of searching for generalized patterns from the training or test tests by inspections, with the suggested approach, the problem is reduced to a more generalized context as an optimization problem - maximization of the activation of a filer. We built this setting on convolutional blocks of pre-trained ConvNets assuming the parameters (weights and biases) are fixed. Let's denote the parameters with θ and input sample with x. Then define our activation function as a function of θ and x. For instance, let $h_{ij}(\theta, x)$ be the activation of a given filter j from a given layer i in the ConvNets, we implement the activation maximization for corresponding the filter as

$$x^* = \arg\max h_{ij}(\theta, x). \tag{1}$$

The equation above leaves us with a non-convex optimization problem, however, it is still possible to solve that to find a local minima. This can easily be performed by applying gradient ascent in the random noise input. We achieve this by setting our gain to $h_{ij}(\theta, x)$ and try maximize it by computing gradient as $\frac{\partial h_{ij}}{\partial x}$ and push the noise image in the opposite direction of this gradient in order to reconstruct the pixel space that the selected filter at the given layer is sensitive to. There are two possible outcomes to this solutions, either we will end up in the same qualitative minimum every time or several local minima will be found. In both cases, the filter can be characterized as the one which yields the maximum activation.

In essence, this method approximately looks for distribution of $\arg\max p(h_{ij} = 1|x)$ to generalize h_{ij}, where h_{ji} is 1 only for the selected layer and filter. The reason that approximates the optimization is because of the non-convexity of the problem as the true posteriors are intractable for higher layers and they are approximated by the output of the corresponding filter. So if can find

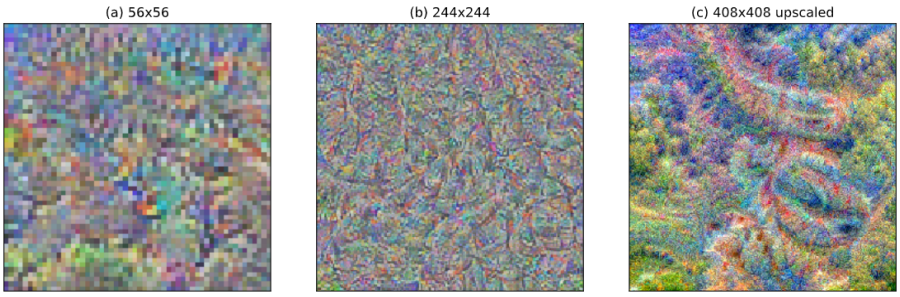

Figure 1: Activation of 164th filter on the relu function from last basic block of the 7th layer the model. (a) shows the reconstruction with image 56×56 dimension, it has low-frequency and low-resolution pattern. (b) shows the reconstruction with image 224×224 dimension, it has a high-frequency and high-resolution pattern. (c) shows reconstruction obtained with an image dimension of 408×408 using up-scaling technique starting at the dimension of 56×56, it has low-frequency and high-resolution pattern.

a way to produce a distribution of $p(x|h_{ji} = 1)$, then we can also characterize h_{ij} by using the distribution $p(x|h_{ji} = 1)$. In that setting, it would be possible to characterize the filter by many samples from this distribution by computing the expectation of $E[x|h_{ij} = 1]$. We know that by definition $E[x|h_{ij} = 1] = \int xp(x|h_{ij} = 1)dx$. Unfortunately, just like the optimization problem for $\arg\max p(h_{ij} = 1|x)$ is con-convex in high layers, this problem is intractable as well. However, if we consider a extreme case where the distribution of the samples concentrates around at x^+, the the $p(x|h_{ji} = 1) \approx \delta_{x^+}(x)$, thus the expectation as well being $E[x|h_{ij} = 1] = \delta_{x^+}(x)$. Although this concentration assumption is not the case in practice, we can still use this notion to show an interesting link between maximization of the activation and $E[x|h_{ij} = 1]$ by using Bayes' theorem to show that

$$\begin{aligned} p(h_{ji} = 1|x) &= \frac{p(x|h_{ji} = 1)p(h_{ji} = 1)}{p(x)} \\ &= \frac{\delta_{x^+}(x)p(h_{ji} = 1)}{p(x)}. \end{aligned} \tag{2}$$

This assumption yield zero everywhere except at x^+. Thus, under this assumption $\arg\max p(h_{ij} = 1|x) = x^+$, which means the expected value over samples and activation maximization should produce very similar results. However, generally speaking, concentration assumption is never the case in real-life, and that is the reason why images acquired by activation maximization looks more like image parts rather than a complete image. Erhan et al. (2009) suggests this phenomenon may be key for a more accurate representation of what particular filter is responsible for in the model.

The suggested optimization technique - activation maximization, was originally implemented on DBN, however, it can easily be extended to any network. In this project, we applied it to ConvNets making few minor adjustments such as up-scaling of pixel space. Like any gradient descent optimization technique, this technique too does require careful adjustment of the hyper-parameters such as a number of descending steps, learning rate, which is the only weakness as it is slightly more flexible to the choice of the values selected.

2.2 Scaling up pixel space

We begin the visualization of ConvNets filters with relatively small size. However, when we try to project the reconstruction to a small space, the formed pattern does not achieve a good resolution in terms of its generalization ability. So the problem arises here as the size output space of which filter is projected is so small for ensuring appropriate visualization for complex image models. Furthermore, if we do the opposite by increasing the image size, then this time the size of the filter relative to the image size will mitigate, but pattern frequency created by it will increases, eventually making it hard to do any inference. The reasoning behind this is that increasing output dimension leaves the filter with enough space to project its characteristics on large space with the more frequent occurrence. As it has depicted in Figure 1., the small feature space leaves us with low-frequency with low-resolution, whereas larger output dimensions result in high resolution with high-frequency in the projection space. In both cases, it is very hard to capture the essential pattern that the visual representation is associated with.

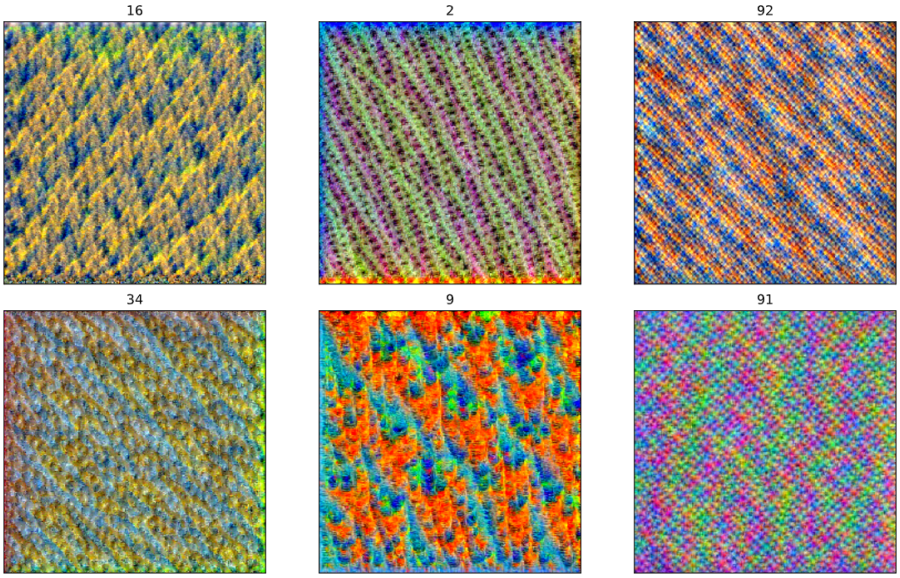

Figure 2: Visualization of the feature spaces reconstructed by the maximizing the activation of 2, 9, 16, 34, 91 and 92nd filters on the first convolutional function of the 3rd basic block of the 5th layer.

To remedy this disadvantage, we used the up-scaling method which starts with a low-resolution image and optimizes its pixel values for several gradient steps, then increases the image size by a certain factor and optimize it again, and repeat the same procedure for several steps. After scaling up the reconstructed image each time, the up-scaled pattern becomes less frequent than what the optimization steps would have produced if we directly started with large image size. This method not only beneficial for interpretability but also sidewalk the non-convexity nature of the optimization problem to some extend. That is because optimizing filter activation with the up-scaling technique leaves the gradient at the better starting point in the next scaling iteration, thus eventually avoiding poor local minimum. Moreover, to reduce the high-frequency pattern formation further, images are blurred after each up-scaling steps which eventually affects the high-frequency pattern more than it does that of low-frequency. This strategy ensures to make reconstruction characterizes the activation h_{ij} with high-frequency and high-resolution, thus more interpretable.

3 Model

The approach we are utilizing make it possible to be easily adapted to any ConvNets model. For this project, we consider the deep residual learning model (He et al., 2016). The main reason for that decision lies in the skip connection that residual models have. The skip connections in residual blocks add the output from an earlier layer to the output of the existing layer helping it to mitigate the vanishing gradient problem. Moreover, explicitly formulating layers by learning residual function with reference to the layer input, the input propagates forward faster through the residual blocks across layers.

The model we use is ResNet34 (He et al., 2016) architecture, which obtained with a residual network of 34 parameter layers (3.5 billion FLOPs). The convolution layers have 3×3 filters with a stride of 2 up to 512 filters, except the first one which has 7×7 filters. The model was trained on ImageNet 2012 classification dataset (Russakovsky et al., 2014), which consist of 1.3 million image samples spread over 1000 classes. Each image is preprocessed by resizing the shorter dimension to randomly sampled scale in [256, 480], randomly cropping 224×224 region from the image or its horizontal flip, subtracting the per-pixel mean (across all images). Right after each convolution just before the linear activation, the Batch Normalization (BN) (Ioffe and Szegedy, 2015) is applied. Stochastic descent up to 60×10^4 iterations with a mini-batch size of 256 is used to update the parameters, starting with the learning rate from 10^{-2} and is divided by 10 each time when the error plateaus. The weight decay of 10^{-5} and a momentum of 0.9 is used. However, the model does not use dropout (Hinton et al., 2012) following the same practice in

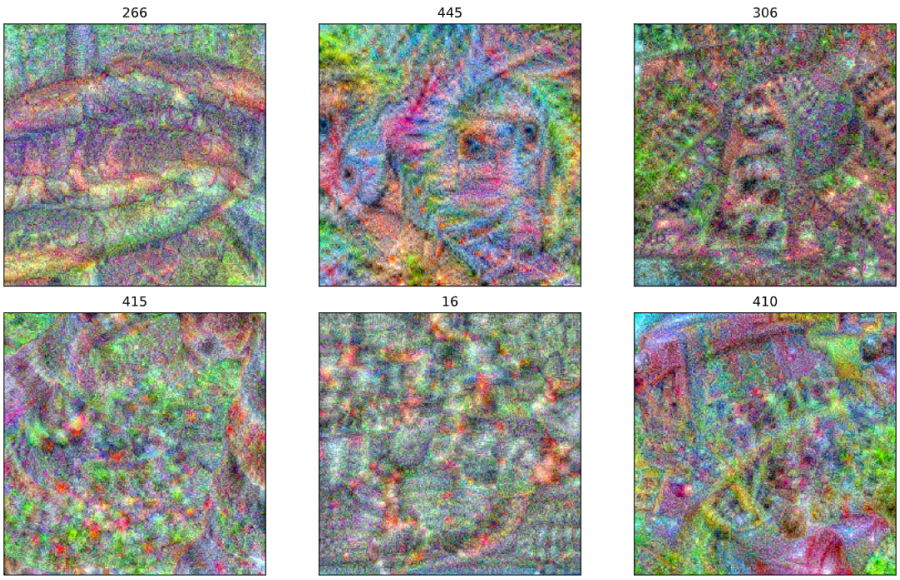

Figure 3: Visualization of the feature spaces reconstructed by the maximizing the activation of 16, 266, 306, 410, 415, and 445th filters on the relu function of the 3rd basic block of the 7th layer.

(Ioffe and Szegedy, 2015).

In this project, as we are interested in optimizing the noise image by maximizing the filter activation in convolution blocks, we excluded the final average pool and fully connected dense layer from the model. Moreover, as we scrutinize what input space that the filters correspond to by taking a gradient with respect to the input image, we use a pre-trained ResNet34 model and freeze its parameters.

4 Experiments

Setup

As we mentioned Section 2.1, we intended to visualize the feature representation that the activation of the given filter at any level can be characterized with. For that, we, initially create a random noise sample with dimension 56×56. Later we upscale this image using the approach in Section 2.2. Input sample is upscaled for 12 iterations with the up-scaling factor of 1.2. For each iteration of upscaling, we transform it in the same way originally proposed for ResNet34 architecture. Then we update the upscaled and transformed sample in the opposite direction of the gradient of the selected filter's activation with respect to it. What we do is to push noise image to form a representation characterized by the pattern of the filter of our interest by maximizing activation of that filter at each gradient step. Hence, we keep the weights of the pre-trained model fixed to see how effective the learned parameters are for visualization as they're used in the forward propagation of the layer to obtain the activation value. So in essence, instead of optimizing the parameters of the model, we optimize the input sample in a direction that yields the highest activation. For the optimization step, we use Adaptive Moment Estimation (Adam) (Kingma and Ba, 2014) algorithm with a single batch, as we only have one input sample. Pixels are updated starting with a learning rate of 10^{-2}, in conjugation with a momentum term of 10^{-7}. The initial iteration for gradient ascent we use here is 20. However, after the half of scaling steps completed, we increase it by the factor of 1.3 to compensate the gradient steps for the increase in reconstruction space to have a better result. After completing the reconstruction we obtain 408 dimensional low-frequency and high-resolution visualization of feature space. Nevertheless, it is important to stress that hyperparameters of optimization are selected after the trial-and-error phase, as there is no quantitative way to evaluate the result of the reconstruction as a function of the selected hyperparameters. In fact, that is the main drawback of this method as requires the careful initialization of optimization parameters.

The full implementation of the term project together with the necessary instruction on how to run it is publicly available on our GitHub repository.[1].

[1]https://github.com/anaramirli/visualizing-cnn-features

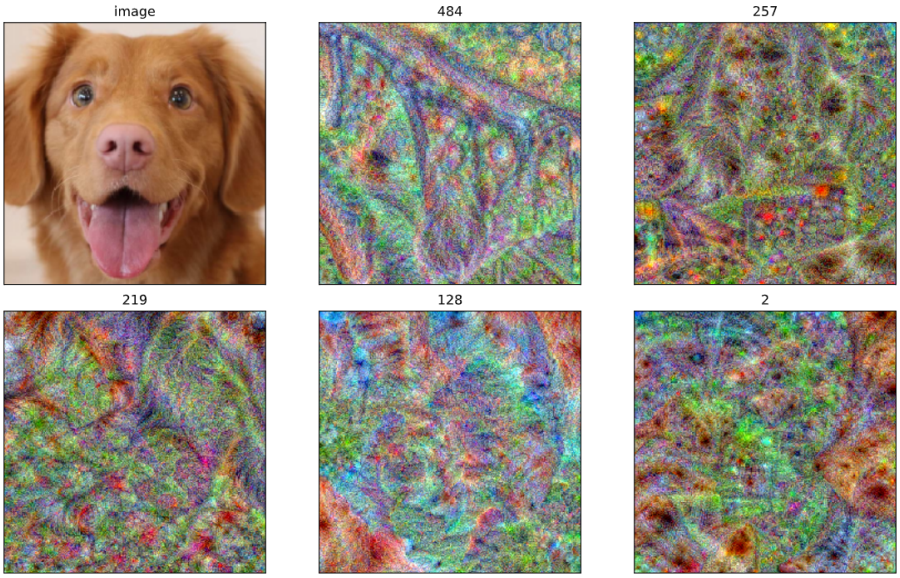

Figure 4: The constructed pixel spaces for dog image corresponding to the 5 most sensitive filters for relu function of the 3th basic block of the 7th layer of the model.

4.1 Maximizing Filters

We being the experiments by the analysis of the activation maximization method. Figure 2. and Figure 3. show the reconstruction for randomly selected 6 filters respectively in the first convolutional function of the 3rd basic block of the 5th layer and the Rectified Linear Activation (relu) function of the 3rd basic block of the 7th layer. This experiment shows the importance of the depth of the model. As the depth increase, the complexity of feature space accounts for the filters in that depths get complex and revealing more information for inference. Moreover, in shallow levels the random initialization yield roughly a similar reconstruction, most of which appears to be random and to have less variability in the pattern. However, in the higher layers, noise sample leads to various prominent invariances, each is far from being random and resembles some part we would observe in natural images. Such good results for high layers are especially surprising, given that, the activation function of high layer filters is a non-convex function of their input. Overall, qualitatively speaking, filters from deep layers are distinguished with meaningful and complex invariances which in its turn is essential to make valid reasoning about the ability of the layers. For the very same reason, in the later experiments, we use the high later filters for visualization and sensitivity analysis.

4.2 Most Activated Filters

Now, to further explore the ability and robustness of the feature activation maximization, we perform sensitivity analysis in order to determent whether these patterns reconstructed are indeed a good representation of a sample space of each class that the model is trained with. Generally speaking, we will test whether the filters are corresponding to these patterns found by the optimization routine. To do this, we choose an image from one of the 1000 image classes that the model is trained. We compute the forward pass for that image up until the layer we're interested in. As the activation function of a filter is a function of input space, we assume the filters that are sensitive to the fitted image must capture most of the invariances in that image. Accordingly, we pick out the most strongly activated filters and reconstruct the pixel space for them. Later we use these reconstructions to compare the original image to see how the filters are generalizing the input space for a given image. In this experiment, for the 3rd block of the 7th layer of the model, we applied the same procedure to the dog image, then selected the 5 most sensitive filters based on their mean activation. The mean activation of filters is given in Figure 5. Although we only use one sample to find out the sensitivity of filters, It worth mentioning that subset of samples, possibly with random augmentations applied, would be better to have a more generalized view of filters activation. Later, we reconstruct the pixel

space associated, this time for each of the 5 most sensitive filters. As we can see in Figure 4., the interpretability of pixel space decrease along with the sensitivity of the filter. Especially, the visualization for the filers 484th and 257th almost resembles a real-life dog, at least with a reasonable resemblance to eyes and nose parts we might expect in natural image.

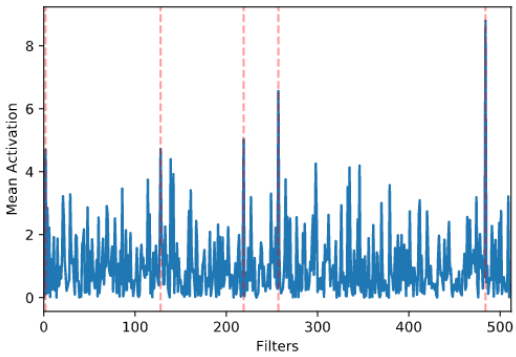

Figure 5: Mean activation of forward pass for filters on relu function of the 3th basic block of the 7th layer computed for god image. We can see that some filters have higher spike form the rest, especially the filter 484 showing significant difference.

Here three scenarios are possible: (i) activation of one or a few of the filers is considerably large from the rest, (ii) some groups of filters are larger from the rest but they have similar activation within-group, or (iii) there is no significant difference for the activation of the filters. These scenarios also give us good insight into what kind of feature reconstruction we might expect. If the forward propagation for the filters produces results similar to that of (i) and quality of interpretability deduced from the reconstructed visualizations yield similar to that as well (i.e., one or a few reconstructions captures the most of the invariances of the feature space associated with given sample image), it means that for the selected image the features space are explained by a few filters. This might be an indicator of the fact that the feature space of that sample is learned well. If the sensitivity of the filters and the reconstructions correspond to the scenario indicated in (ii). Then, we might conclude that for this sample space, the model is still doing reasonably well, however, this time the feature space is shared by many filters. For that kind of cases, we might expect the occlusion of the image parts to cause a reduction in the quality of the prediction. Finally, if the sensitivity of activation after a forward pass is similar to that of (iii) and non of the visualizations achieved with the maximization technique bears any resemblance to the image, it is safe to deduce that the filter has not been able to capture the representation of feature space for that images. Generally speaking, we can easily observe the phenomenon of (iii) quantitively while checking the evaluation scores for the classification. However for the first two cases - (i) and (ii), that is not the same as they both generalized by the model well, but on different levels. Of course, in the case of (ii), although we don't need to do activation maximization to know which occluded part of the image cause reduction in quality and what filters are responsible for that (we can simply analyze it by simple forward pass), the reconstruction of pixel space corresponding to the sensitive filters might still give us very significant insight into what we have to fix. For that, the visualization of reconstruction along with the filter sensitivity analysis is important to capture these inferences about the model performance.

4.3 Activation of Occluded Image

The maximization of filter activation alone is not informative in indicating which part of the image receives high stimuli for the activation of the filters. Moreover, to prove the deduction that we draw in Section 4.2 about the activation of the filters and the corresponding reconstruction for that filters, we perform a very simple sensitivity analysis for occluded dog image and compare it with the reconstruction results we obtained. In this experiment we use the same dog image that we used in activation maximization. The only difference here we will be analyzing the first 10 strongest filters to achieve a more generalized assumption. We occluded several parts of the image that we thought might be important, moreover in one example, we covered the face of the dog completely, to see how the filter activation would react to it. The example of the occlusion can be seen in Figure 6.

Our observations in this experiment indeed justifies the deduction we made earlier about the relation of the activation of filters and the corresponding visualization in regard to the parts of the image stimuli. We can see from Figure 4. that the reconstructions suggest that the most prominent filters are 484 and 257 . If we look closely, we can see that reconstitution space for these two filters resembles a complete dog's face, with nose and both eyes being slightly more 'dominant', rather than the

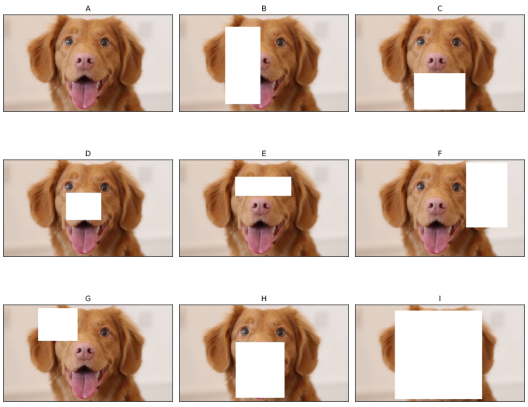

Figure 6: Example of occluded dog images used in our experiment.

image	most sensitive filters	# common
A	[484, 257, 219, 128, 2, 139, 298, 346, 335, 142]	
B	[128, 257, 484, 219, 114, 420, 2, 139, 142, 361]	7/10
C	[484, 139, 257, 219, 128, 2, 335, 114, 142, 265]	8/10
D	[484, 257, 128, 219, 249, 335, 298, 86, 114, 464]	6/10
E	[484, 257, 128, 219, 139, 114, 265, 464, 142, 2]	7/10
F	[128, 257, 2, 484, 139, 335, 114, 219, 379, 142]	8/10
G	[484, 257, 114, 2, 128, 139, 219, 335, 142, 265]	8/10
H	[484, 219, 128, 257, 114, 335, 298, 474, 29, 412]	6/10
I	[153, 420, 352, 92, 361, 88, 270, 20, 148, 114]	0/10

Table 1: Depicts the 10 filters with largest mean activation of forward pass in descending order for relu function of the 3th basic block of the 7th layer and compares how many filters that the occluded images have with the unoccluded image A.

other parts of the dog's face. This also supported by the filter sensitivity results we got for occluded images. For instance, for the occluded images B, D, E, H, the set of sensitive filters found still contains these two filters, however, they contain less common filters with that of what achieves the highest sensitivity for the original image, each having 7, 6, 7, and 6 filters in common respectively. Moreover, we also found a slight decrease in the mean activation for these images. These facts not only suggest the deductions we made are in the right direction but also implies another finding. That is, although the nose and eyes are crucial stimulus parts for feature space of the dog samples, when they are occluded activations in that layers still reasonably good. That is because most strong filters associated with dog sample concentrated on whole dog face, rather than the parts of it. When we carry out the procedure to occluded images of the less important parts such as ears or one of the eye, we still obtain the same result but less sensitive to occlusion than that of the 'dominant parts. Another interesting phenomenon happens when we completely cover the dog's face. We see that non of the first 10 most active filters found are in common with that of the original image. Plus, the activation of the filters at that level all are around the same number (excluding the one filters which seems to be associated with rectangular shapes, as we did occlusion in a rectangular shape), point out the fact that if filters have no idea about the image it is not familiar.

5 Conclusion

In this term project, we used the activation maximization technique to visualize the high layer filters of ConvNets. The nature of that approach is straightforward and can easily be applied to any vision model. We build visualization from random noise image by updating it along the opposite direction of the gradient of the selected activation with respect to the reconstructed image. Doing that we acquired visualization in which the selected filters any layer is characterized by maximization the activation of that filter. Even though the nature of the gradient problem is non-convex, this methods still manages to produce sensible visualization for the filter its associated with. The only drawback is that the gradient problems are more sensitive to the choice of hyperparameters and they have to be chosen carefully in order to obtain proper visualization. We found that the deep layer learn more complex structures than shallow layers and moreover there are three main scenarios that the filters that can capture the feature space explained by trained samples. In addition to visualization of filters, we also conduct simple sensitivity analysis of the occluded images to reveal which parts of the sample vector is important for the activation of filters. Overall, using a very simple method to visualize what filters are doing at each layer, we showed that we can gain essential insight into what our vision model is actually doing. That in its turns can be quite helpful when designing a new architecture, for example, judging the quality of extended layers from their visualization, or the effect of the altering kernel dimension.

References

Dumitru Erhan, Y. Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing higher-layer features of a deep network. *Technical Report, Univeristé de Montréal.*

Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. 2016. Compact bilinear pooling. In *2016 IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 317–326.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Lisa Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. volume 9908.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580. Cite arxiv:1207.0580.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei. 2014. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

M. D. Zeiler, G. W. Taylor, and R. Fergus. 2011. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham. Springer International Publishing.