CNN Fine-tuned on Blurry Images Captures Blur Structure

Zhenyang Sun, Xinyi Chen, Yanling Hu

Abstract

Due to the imperfection in sensory organs, data collected by the brain are inherently noisy. Similarly, noise is ubiquitous to training data and applications in machine learning. However, while the human brain makes use of noise to enhance signal processing, noise is generally frowned upon in machine learning, hence the plethora of denoising techniques and networks. Given the similarity of image categorization networks and human visual cortex, it is curious whether noise could potentially do more than just interfere with network performance. Here we investigate how image classification network can learn noise structure and shift its "focus" accordingly, using gradient visualization.

1 Introduction

Suppose you are presented with a clear image of a cat, the defining features for which you use to identify it as a cat may be the face. However, if you are presented with the same image but blurred, the features for which you use to identify it may now be the legs or the tail. The difference in what we focus on in image classification decisions reflects how we think. Do similar mechanisms apply to machine learning image classification algorithms?

Figure 1: clear images and noisy images

Visual processing in the biological brain reacts intriguingly with noise, sometimes even with improved accuracy with the addition of noise (5). Considering the parallel between artificial neural and biological neuronal networks (brain-score), and the distant goal of creating human-level intelligence, this is one more reason in investigating networks' reaction in relation to noise/blur.

Currently, none of the published works focus on interpreting the features of classifying blurry images (16). For example, Generative Adversarial Networks are used as a discriminator that predicts whether a given image is a clear image or a blurry image with random noise added (13). However, not much analysis beyond visualization is done on the filters used to differentiate the clear v.s. blurry images.

Our analysis aims to answer question: How does image classification differ for blurry images and their clear counterparts? Specifically, we will use visualization on gradients to see what

Preprint. Under review.

parts of the image do the network "focus" on, comparing between a pre-trained network fine-tuned on clear images, and the same network fine-tuned on blurry images.

2 Related Works

Starting from the Denoising Convolutional Neural Network (DnCNN), CNN becomes popular in denoising images.(18) Generally, a blur image B is modeled as

$$B = K * L + N$$

where K is a blur kernel and L is a latent image or clear image, N is noise and $*$ is the convolution operator.(2)
When the kernel K is given, the problem is known as non-blind deblurring. The previous researches use deconvolution or apply deep networks directly on blurred images(14; 19; 15; 3; 17). When the kernel K is not given, the problems is then called blind deblurring. Earlier researches assume the kernel K is applied uniformly on images(2; 8; 4; 11) while the later studies focus more on images with different blur kernels applied to different positions(10; 17).

Discover the "focus" of the network:
This idea stems from the desire to interpret the inner workings of neural networks, especially how and what does the network focus on given in a given image. Essentially, what does the network "see", and whether that is human interpretable. To achieve this, various techniques are developed to analyze the gradients of the network with respect to an image, or a class of image (1; 6). Notable endeavors in this aspect include DeepDream and SmoothGrad, and the latter will be used in this work (12; 9) .

3 Methods and Algorithm

Our goal is to observe how noise affects what an image classifier learns. We aim to justify our hypothesis that an expert classifier that is fine-tuned on blurry images will focus on different pixels during image classification compared to a naive classifier that is fine-tuned on clear images.

3.1 Data Loading and Model Set-Up

We used the cats and dogs dataset from Kaggle to fine-tune two separate classifiers from the pre-trained Alexnet instant (available in PyTorch) – one naive classifier fine-tuned on clear images and one expert classifier fine-tuned on blurry versions of the clear images.

3.2 SmoothGrad

We consider the forward of an neural network as a classification function S which takes in an image and outputs the predicted class, which in our case is either dog or cat. In order to locate "important" pixels in input images, we create sensitivity maps for each input image by computing the gradient of function S (ie: $\frac{\partial S}{\partial x}$) and plotting gradients. Intuitively, $\frac{\partial S}{\partial x}$ is the change to the predicted class with respect to small change in pixel value x. As mentioned by Smilkov et al., S may fluctuate sharply at small scale and thus need to be smoothed. He suggests to apply an Gaussian kernel on images which can be approximated by $\frac{1}{n}\sum_{i=1}^{n}\frac{\partial S}{\partial(x+\theta)}$, where n is the number of samples and θ is sampled independently from $N(0,\sigma^2)$ with standard deviation σ. The method of computing gradient and smoothing it by adding noises is called SmoothGrad. To save training time, we compute $\frac{\partial S}{\partial(x+\theta)}$, which is an estimate of the approximation that Smilkov proposed.(12) Since functions S predicts two class, we obtain two sensitivity map for one input image. We plot the sum of two sensitivity maps to capture all pixels that CNN takes into account when making a prediction.

3.3 Computing the distances

To understand how the presence of noise and the different classifiers affect the focus of the neural network, we compute the following distances:

1. The distance between the focus of the **naive classifier** on clear images and on noisy images.
2. The distance between the focus of the **expert classifier** on clear images and on noisy images.
3. The distance between the focus of the **naive classifier** and the focus of **expert classifier** both on noisy images.

The focus of a network is defined to be the top 20 pixels with the most intensity after running SmoothGrad, and the distances are computed using Euclidean norms.

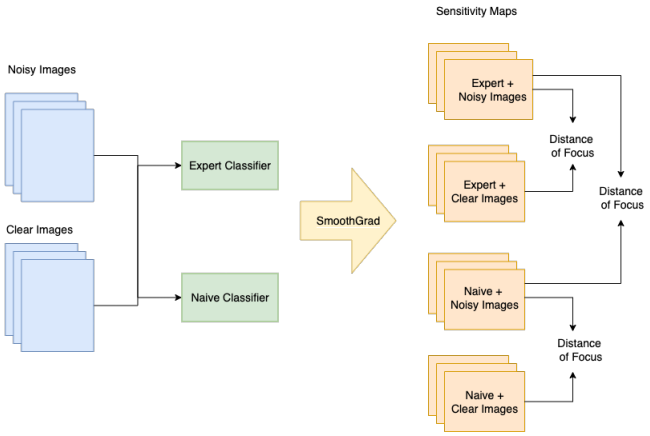


Figure 2: the procedure to compute the distances

3.4 T-test for Significance

We use a Student's t-test to test if the distances mentioned above are statistically significant. We assume that the distances follow a Gaussian distribution and are sampled independently and randomly from the population. To perform a t-test on a multidimensional matrix of distances like this, we will run a one-sample t-test for each pixel (with sample size = number of images), combine the 20 obtained p values to form adjusted p-values and conclude our results by comparing the adjusted p-values with the significance level (0.05). Our null hypothesis is that the distances have a mean of 0, meaning that there is no significance in the distances observed.

4 Results and Discussion

4.1 Visualizations of intermediate convolution layers

From plotting the activation maps in the intermediate concolution layers of the classifiers after feeding in a noisy image, we see that fine-tuning changed the weights and activations of the models (made the outlines less apparent as we are now only accounting for 2 classes instead of many classes). Also, we observe that compared to the naive classifier, the expert classifier has areas with high activation that closest resemble a rough outline of the object. To be specific, the expert classifier shows clear evidence of highlighting the tail and head of the cat and the limbs of the dog in the blurry images, supporting our hypothesis that the expert classifier focuses on different parts of an image compared to the naive classifier. (Figures and more details in the Appendix)

4.2 Qualitative Comparisons of the Sensitivity Map

Comparing vertically in Figure 2, we observe that the naive classifier has a more spread out focus on a noisy image than its clear counterpart, while the expert classifier shows little difference in focus when faced with the clear and noisy cat images. Also, comparing horizontally, the expert classifier is more condensed than the naive classifier on both clear and noisy images.

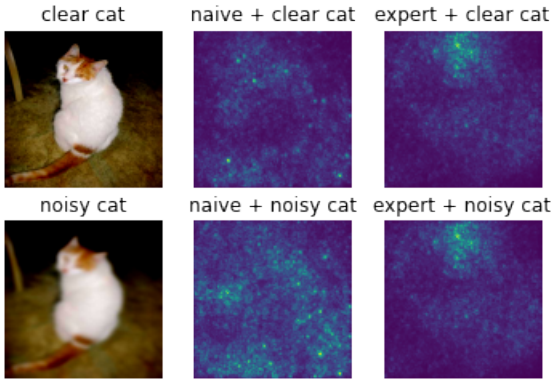


Figure 3: sensitivity maps of naive and expert classifier on clear and noisy cat image

4.3 T-test results

With the naive classifier, we ran a t-test on the computed distances between the clear images and the blurry images for each of the top 20 pixels. The results showed that the adjusted p values for all 20 pixels were 0.00. Hence, we have strong evidence against the fact that the naive classifier with noisy inputs focuses on the same pixels as the naive classifier with clear images. This suggests that the presence of noise itself affects the attention of the naive classifier.

We performed the same test on the computed distances using the expert classifier. The adjusted p values are again, 0.00 and hence we can conclude that the presence of noise indeed affects where the model focuses.

With this knowledge, we further test if the different classifiers focus on different pixels when given the same noisy inputs. We ran a t-test on the computed distances between the naive classifier and the expert classifier on noisy inputs for each of the top 20 pixels. The results show that we have strong evidence against the fact that the naive classifier with noisy inputs focuses on the same pixels as the expert classifier with noisy inputs. (See Appendix)

To examine if the difference due to the classifier was bigger than the difference due to the presence of noise, we conducted a one-sided t-test. The results indicate adjusted p-values that are all lower than (8.13e-16 < 0.05) for both the naive and expert classifier; hence, we can conclude that the effects on focus pixels due to different classifiers are larger than the effects on focus pixels due to noise.

The results from the T-test, as well as examples of visualization in Figure 3 show that the expert network, fine-tuned on noisy images, has learned to to focus on specific features to overcome the effect of noise. Therefore, it could be said that the network has learned the structure of the noise.

4.4 Strengths and limitations

Since no prior work was performed on visualizing the focus of networks when faced with noise, our qualitative and quantitative analysis are quite novel but experimental. We have utilized a comprehensive approach before carefully evaluating our results. However, some limitations include the potential bias in the interpretation of the saliency, the overly optimistic assumption for the distribution of the distances for the t-test, and the potential p-hacking effects of combining multiple p-values.

4.5 Future Work

After showing that the network shifts focus after learning the structure of blur, a future direction could be analyzing the features that network is focusing on. A useful and time-tested technique is Scale Invariant Feature Transform that analyzes image properties of a pixel's neighborhood (7). We could analyze features of the foci within and across different type of blur to find out invariant features with

respect to different blurs. Further understanding of blurs would help with development of debluring algorithms.

5 Conclusion

Through fine-tuning of existing architecture, we attempted to let a classification network learn the structure of blur. We demonstrated that said network has adapted to blur by comparing the foci of networks fine-tuned or not by noisy images. We showed that the network was able to change its foci according to blur, signifying its understanding of the blur structure. The present work represents a step towards assessing network's understanding of noisy inputs, with future implication of developing better debluring methods.

6 Code Citation

Our code about the sensitivity map consults some ideas implemented by kazuto1011 in Github (link). Also, we consults the implementation of visualizing intermediate layers by Neuromatch Academy (link)

References

[1] ADEBAYO, J., GILMER, J., MUELLY, M., GOODFELLOW, I., HARDT, M., AND KIM, B. Sanity checks for saliency maps. *Advances in neural information processing systems 31* (2018).

[2] CHO, S., AND LEE, S. Fast motion deblurring. In *ACM SIGGRAPH Asia 2009 papers*. 2009, pp. 1–8.

[3] DONG, J., ROTH, S., AND SCHIELE, B. Deep wiener deconvolution: Wiener meets deep learning for image deblurring. *Advances in Neural Information Processing Systems 33* (2020), 1048–1059.

[4] FERGUS, R., SINGH, B., HERTZMANN, A., ROWEIS, S. T., AND FREEMAN, W. T. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*. 2006, pp. 787–794.

[5] FUNKE, K., KERSCHER, N. J., AND WÖRGÖTTER, F. Noise-improved signal detection in cat primary visual cortex via a well-balanced stochastic resonance-like procedure. *European Journal of Neuroscience 26*, 5 (2007), 1322–1332.

[6] KIM, B., WATTENBERG, M., GILMER, J., CAI, C., WEXLER, J., VIEGAS, F., ET AL. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning* (2018), PMLR, pp. 2668–2677.

[7] LOWE, D. G. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision* (1999), vol. 2, Ieee, pp. 1150–1157.

[8] MICHAELI, T., AND IRANI, M. Blind deblurring using internal patch recurrence. In *European conference on computer vision* (2014), Springer, pp. 783–798.

[9] OLAH, C., MORDVINTSEV, A., AND SCHUBERT, L. Feature visualization. *Distill 2*, 11 (2017), e7.

[10] RIM, J., LEE, H., WON, J., AND CHO, S. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision* (2020), Springer, pp. 184–201.

[11] SCHULER, C. J., HIRSCH, M., HARMELING, S., AND SCHÖLKOPF, B. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence 38*, 7 (2015), 1439–1451.

[12] SMILKOV, D., THORAT, N., KIM, B., VIÉGAS, F., AND WATTENBERG, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

[13] WANG, X., YU, K., WU, S., GU, J., LIU, Y., DONG, C., LOY, C. C., QIAO, Y., AND TANG, X. ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR abs/1809.00219* (2018).

[14] XU, L., REN, J. S., LIU, C., AND JIA, J. Deep convolutional neural network for image deconvolution. *Advances in neural information processing systems 27* (2014).

[15] ZHANG, J., PAN, J., LAI, W.-S., LAU, R. W., AND YANG, M.-H. Learning fully convolutional networks for iterative non-blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 3817–3825.

[16] ZHANG, K., REN, W., LUO, W., LAI, W., STENGER, B., YANG, M., AND LI, H. Deep image deblurring: A survey. *CoRR abs/2201.10700* (2022).

[17] ZHANG, K., REN, W., LUO, W., LAI, W.-S., STENGER, B., YANG, M.-H., AND LI, H. Deep image deblurring: A survey. *arXiv preprint arXiv:2201.10700* (2022).

[18] ZHANG, K., ZUO, W., CHEN, Y., MENG, D., AND ZHANG, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing 26*, 7 (2017), 3142–3155.

[19] ZHANG, K., ZUO, W., GU, S., AND ZHANG, L. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 3929–3938.

7 Appendix

7.1 Visualizations of the Intermediate Layers

To get a sense of the general discrepancies between the two classifiers from the same image input, we plotted the activations of the 6th, 8th, and 10th convolution layers of AlexNet for the pre-trained model, the naive classifier, and the expert classifier. (More Details in the Appendix)

7.2 Visualizations of the intermediate activation maps for different classifiers

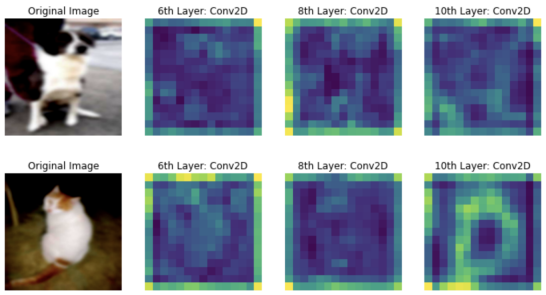


Figure 4: Activation Maps for the 6th, 8th, and 10th layer of the Pre-trained AlexNet Model

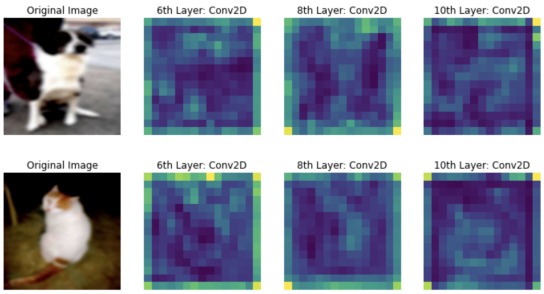


Figure 5: Activation Maps for the 6th, 8th, and 10th layer of the Naive Classifier

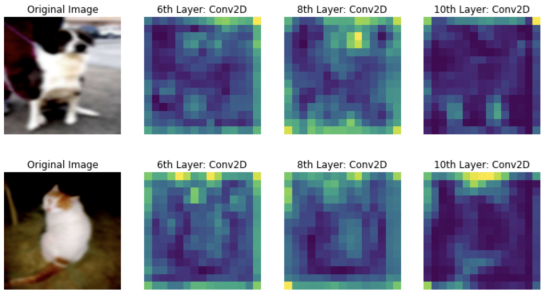


Figure 6: Activation Maps for the 6th, 8th, and 10th layer of the Expert Classifier

7.3 Results of T-test

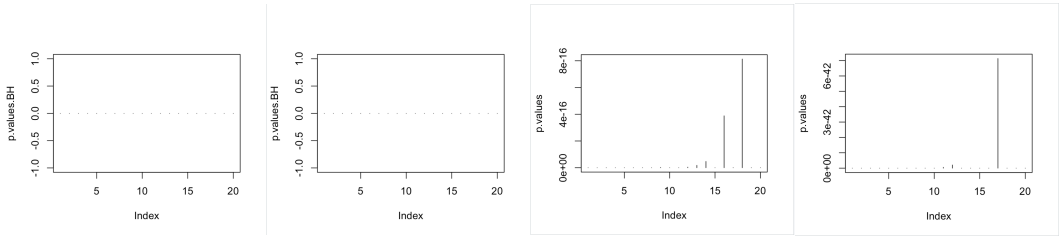
X-axis is the index of the top 20 focus pixels (smaller meaning higher focus).

[a] Adjusted p values for t-test with (top 20 pixels from naive classifier with clear images minus top 20 pixels from naive classifier with noisy images)

[b] Adjusted p values for t-test with (top 20 pixels from expert classifier with clear images minus top 20 pixels from expert classifier with noisy images)

[c] Adjusted p values for t-test with (distance resulting from noise minus distance resulting from network) for the naive classifier

[d] Adjusted p values for t-test with (distance resulting from noise minus distance resulting from network) for the expert classifier



[a] [b] [c] [d]

7.4 Contribution

Zhenyang Sun: Trained networks, oversaw visualization and smoothgrad implementation, wrote 1/3 of the report
Xinyi Chen: Coded the visualizations of activation maps, conducted the t-tests, wrote 1/3 of the report
Yanling Hu: Computed smoothed gradients, plotted sensitivity maps, found focus of CNN, wrote 1/3 of the report