Training Deep Neural Networks on Occluded Datasets

Luke Bun
UW Bioengineering
lukebun@uw.edu

Shao-Jung Kan
UW Electrical and Computer Engineering
kanjames@uw.edu

Cameron McCarty
UW Electrical and Computer Engineering
cmmc6@uw.edu

Abstract

A common weakness of convolutional neural networks (CNNs) trained for object recognition is their poor performance on classifying images that are distorted compared to images present in the original dataset. One type of distortion is occlusion, wherein a part of an image has been masked by an object in the foreground. While a human observer may be able to still recognize an object even when large parts have been occluded, a CNN may struggle. Here train a CNN to be robust to occluded images. We modified the CIFAR-10 dataset with occluding masks (black squares and Gaussian noise). We found that CNNs trained on occluded data are more accurate than models trained on unoccluded data on both unoccluded and occluded test sets.

1 Introduction

Convolutional neural networks (CNNs) trained for object recognition often fail to classify images that have been distorted in ways that weren't present in the training set. This is partially because CNNs tend to learn only a small number of features of each class. For example, a CNN may learn that a person is defined by having a face. But in an image where the face is occluded an object in the foreground, then the CNN may not recognize it as a person. In contrast, the human visual system can use a variety of cues to recognize objects despite occlusion. A human observer would be able to recognize other parts of the body such as arms and legs and their relative positions to recognize an occluded person.

How can this same capability be incorporated into CNNs? How can they be forced to learn to recognize objects using multiple features? We hypothesized that attribute can be accomplished by training a CNN with a modified image data set that includes occluded images. In doing so, the CNN will be forced to learn a wider variety of features as no single feature can reliably describe an object.

In line with this hypothesis, we find that CNNs trained to recognize objects in occluded datasets are robust to occluded data and even perform better on unoccluded data than a model trained solely on unoccluded data. The results of this work may act a powerful framework for training CNNs to function robustly in realistic environments, where occluded objects are common.

2 Related work

One area that uses occluded images heavily is the field of explainable artificial intelligence (XAI). For example with the RISE method [1] randomised masks are used to remove areas from inputs before running the model and analyzing the outputs. This method gives importance values to every pixel, and in this paper, this logic is used to spread out importance to reduce the reliance on certain pixels. Another insight from XAI is the data manifold [2], where XAI methods suffer due to the models not being well defined in the regions of the high dimensional input space that have not been trained on, this can cause undefined behavior of the model when occluded samples are tested. By training the model on already occluded images, the manifold space is increased to cover these regions commonly used by XAI feature occlusion methods. This increases the interpretability of models without limiting their architecture.

3 Technical description

3.1 Dataset

We use a modified version of CIFAR-10 [5]. CIFAR-10 images are sufficiently large (32-by-32 pixels) and diverse (50,000 training images evenly distributed among 10 classes) that they would be suitable for our occlusion study (Fig. 1). Other relatively small datasets, such as Tiny ImageNet [6], which use 64-by-64 pixel images, were thought to be excessive for our needs and using them would be an inefficient use of computation time. We also opted to use CIFAR-10 over CIFAR-100, which has the same number of images with the same size (32-by-32) but with more diverse classes. This was because we hypothesize that training on a dataset with a large number of classes classes would make performance worse than training with fewer classes. This may be further exacerbated by training on occluded images which could result in performance for all models being degraded to the point where they all perform so poorly that it becomes difficult to discriminate their performances.

Figure 1: Example CIFAR-10 image of a horse. Every CIFAR image is only 32-by-32 pixels, making it computationally inexpensive to train over the dataset.

We then modify the CIFAR-10 database by occluding different parts of each image. Each 32-by-32 pixel CIFAR-10 image is divided into a 3-by-3 grid of super pixels, where the superpixel in the upper left corner is 10-by-10 pixels, while the other 8 superpixels are 11-by-11 pixels. Next, for each superpixel, we create a new image with one of two occlusion methods. The first occlusion method replaces each superpixel with a black square of equal size (Fig. 2). This results in a region where none of the original information in the black superpixel region remains. This would be analogous to a person standing behind a tree which blocks their entire face. We hypothesize that this will force the network to learn features from other regions. The second method adds Gaussian noise (mean 0 and standard deviation 30) over the superpixel region (Fig. 3) resticting values to be within the display gamut between 0 and 255. Some details are still visible, such as object boundaries but finer details are obscured. This may be analogous to having a person's face hidden by the leaves of a tree, where some details are still available, but are harder to discern. We expect that this may allow the model to still learn features from occluded regions, but would discourage it from doing so. Each occlusion method produces a dataset that is nine times the size of the initial CIFAR-10 dataset.

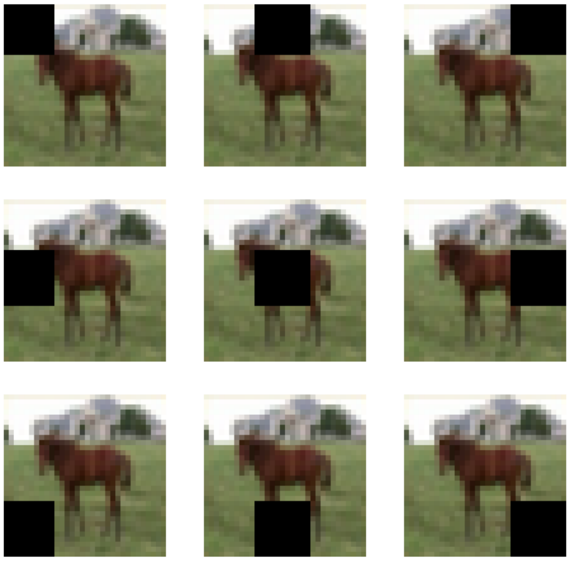

Figure 2: CIFAR image with black occlusion. For 9 superpixels in a 3-by-3 grid, all pixels are set to be black with RGB = [0,0,0]. In doing so, all information in that area is lost, as if blocked by a physical object.

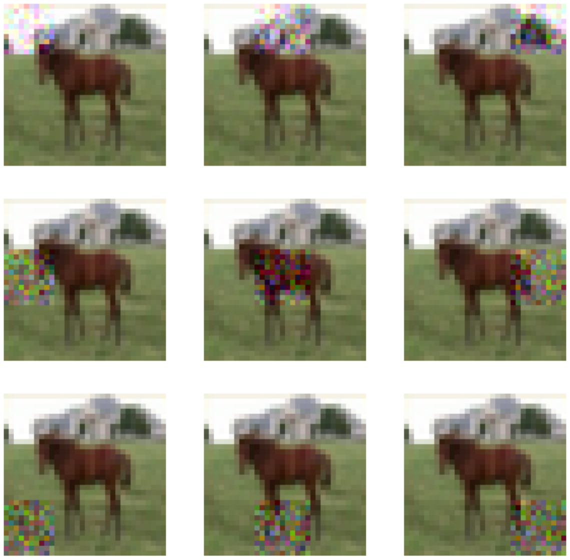

Figure 3: CIFAR image with Gaussian noise. For 9 superpixels in a 3-by-3 grid, all pixels had Gaussian noise added. In doing so, some information is retained, as if blocked by a scattering of smaller objects.

3.2 Model

We use a slightly modified version of VGG-11 [7]. VGG-11 was originally trained on the ImageNet dataset [8] which has 1000 classes, therefore the last layer is slightly modified to have fewer output classes (10 rather than 1000).

Compared to larger datasets, such as ImageNet, CIFAR-10 is relatively simple. Therefore, deeper networks, such as VGG-19 or ResNet-18 [9], were deemed excessive and would greatly increase computation time. AlexNet [10], an even shallower network was not used because of large kernel sizes in its early layers (11-by-11 pixels in Conv1), which would cover a large amount of the input image with each convolutional step. Such large kernels may not be able to capture high spatial frequency information in the CIFAR-10 dataset. While images could be resized to be larger, doing so would needlessly increase computation time.

3.3 Training

We train three models: one on standard CIFAR-10, one on CIFAR-10 with black superpixels, and one on CIFAR-10 with Gaussian noise superpixels. During training, images are Z-scored using the mean and standard deviation of pixel values in the unoccluded CIFAR-10 dataset. Each model has its training hyperparameters (momentum, learning rate and weight decay) optimized. Data are randomly split into training (80%) and validation (20%) sets and loaded into batches with 64 images each. Then a greedy algorithm is used to optimize each hyperparameter one at a time. Momentum is optimized first by testing five values uniformly spaced between 0.5 and 1.3, keeping both learning rate and weight decay constant at 1e-3 over two epochs. The momentum that resulted in the model with the best accuracy on the validation set is used going forwards. Similarly, learning rate is optimized first by testing a range of five values logrithmically spaced between 1e-1 and 1e-5, using the previously optimized momentum while weight decay is still 1e-3 over two epochs. Much like momentum, the learning rate that resulted in the best performance on the validation set is used going forwards. Finally, we repeat the process for weight decay, testing the same values as learning rate, using the optimized momentum and learning rate previously calculated. After optimizing training hyperparameters, each model is trained on its respective training set for 20 epochs.

3.4 Evaluation

After training, each model is evaluated on the standard CIFAR-10 test set, and black box occluded and Gaussian noise occluded versions of the same test set, manipulated using the same occlusion methods applied to the training set. The occluded test sets are nine times the size of the original test set. Performance for each model on each test set was compared to determine if training on occluded images significantly improves performance on occluded image classification and which occlusion strategy is more generalizable.

4 Experimental results

4.1 Optimized training parameters

We used a greedy algorithm to identify the optimal training parameters to train the CNN on each training set. We found that the training parameters for each are similar, with the training parameters for the black box and Gaussian noise occluded training sets to be identical (Table 1).

Table 1: Optimal training parameters

Parameter	**CIFAR-10 model**	**Black box model**	**Gaussian noise model**
Momentum	0.9	0.9	0.9
Learning rate	0.001	0.01	0.01
Weight decay	0.00001	0.0001	0.0001

4.2 Training on occluded datasets

We then evaluated each of the three trained models on the three equivalent test sets (Table 2). We found that all three models perform well on the unoccluded dataset, with the models trained on occluded data outperforming the model trained on unoccluded data, with the model trained on black box occluded data performing the best.

All models also perform similarly well on data occluded by Gaussian noise, with the model specifically trained on Gaussian noise performing the best. In contrast, both the model trained on unoccluded data and the model trained on Gaussian noise occluded data perform poorly on the black box occluded test set. The model trained on Gaussian noise actually performed worse than the model trained on unoccluded data. However, the model trained on black box occluded data still performed well.

Table 2: Test set accuracy

Test set	**CIFAR-10 model**	**Black box model**	**Gaussian noise model**
CIFAR-10	73.900000%	79.870000%	78.200000%
Black box	55.713333%	77.111111%	49.623333%
Gaussian noise	72.440000%	77.758889%	77.797778%

Note however, that the black box and Gaussian noise occluded models use training sets nine times the size as the model trained on unoccluded data. While each occluded image was unique, they still have very similar information between them and therefore may be akin to training on the same image multiple times or training for more epochs. To control for this, we also normalized the performances of each model by its performance on the unoccluded test set via percent difference (Table 3):

$$\%Difference = \frac{Occluded\% - \text{unoccluded}\,\%}{\text{unoccluded}\,\%} * 100$$

We observe the same pattern, with the CNN trained on black box occluded images consistently performing well compared to unoccluded images and the other two CNNs struggling on the black box occluded test set.

Table 3: Percent difference

Test set	**CIFAR-10 model**	**Black box model**	**Gaussian noise model**
Black box	-24.609833%	-3.454224%	-36.543052%
Gaussian noise	-1.975643%	-2.643184%	-0.514351%

5 Discussion

We trained the same CNN architecture on three variants of the CIFAR-10 dataset (unoccluded, occluded with black boxes, and occluded with Gaussian noise). We then tested those models on equivalent test sets. We found that the training on occluded images improves performance on unoccluded and occluded images, with models trained on black box occluded image performing consistently well on multiple test sets. This is consistent with our initial hypothesis that training on occluded images improves recognition of occluded objects.

The improved performance of occlusion trained models compared to unoccluded models may be the result of forcing them to learn a wider variety of features. Occlusion may act as a form of regularization, preventing overfitting to a handful of features for each object class. When an image is occluded, some of these features may be missing but because our occlusion trained models are trained to recognize a multitude of features, the absence of those features in some examples may have little impact. Furthermore, the consistent performance of the black box occlusion model may the the result of how extremely it is forced to learn multiple features, given the lack of information in some occluded images. However, an alternative explanation posits that this improved performance may simply be the result of using larger training sets for the occlusion models. Future work may more thoroughly control for this by training all models on an equal number of images, but randomly occluding parts of each image, using methods similar to those often used for random affine and mirrored transformations.

We also observed that all models perform well on Gaussian noise occluded data. One possibility is that is that the noise we added was insufficient to occlude details in the data, and thus all models were able to still identify those details. This may not be a shortcoming of Gaussian noise occlusion and may be a consequence of our choice of standard deviation. Using a larger standard deviation would increase the range of distortion and may result in greater occlusion. Varying the standard deviation of Gaussian noise is an important future direction.

Finally, we observed that the model trained on Gaussian noise occlusion performed worse on the black box occlusion test set than the other two models. While it is intuitive the black box model, which was explicitly trained on similar data, would perform the best, it is less clear why training on Gaussian noise occlusion degrades performance for these test data. One possibility is that instead of the Gaussian noise model learning a variety of features, it simply learned to extract features from the noise in occluded regions and in doing so allocating valuable computations for that procedure that may have been better used for extracting other features. This hypothesis could be tested by taking the early layers of the Gaussian noise occlusion model and transferring them to a new model and training the later layers for a de-noising task, attempting the reconstruct the original image after it has been occluded by noise. If the Gaussian noise occlusion model is already de-noising occluded regions, then the new model will out perform a similar one based on the unoccluded model.

5.1 Strengths and weaknesses

5.1.1 Strengths

This work provides a novel comparison of two occlusion methods: black boxes and Gaussian noise. Within the scope of our testing we have definitively shown that the Gaussian noise is a weak method for occlusion. This can be extended to any method that can be removed or lessened by a convolutional filter. Our work also shows promise in the field of XAI, where occlusion is a common method for testing feature importance. Training a model to be used on occlusion will make the model more explainable by these standardised methods. The greatest strength of this work is however as a template for another study that addresses this ones weaknesses.

5.1.2 Weaknesses

This study suffers from a lack of scale. CIFAR-10 is a computationally inexpensive dataset and is sufficient for preliminary work but is very small compared to other datasets. This study also only looked at two occlusion methods, there are a wide array more that can be tried. The method also suffered from a bias towards the occluded datasets as they contained more images and so where trained for longer. Due to the inability to gather data, this study also does not test on natural images of occlusion, like a person standing behind a tree, instead only testing on manufactured data.So much so that observations form this dataset may not be generalize to the wide diversity of natural images. Similarly, only one CNN architecture was used. Given the wide range of possible architectures it is entirely possible that other networks may arrive at different conclusions. Finally, the methods of occlusion were limited. Occluders were always the same size (10-by-10 or 11-by-11 pixels), while varying the area and shape may induce different results.

5.1.3 Suggested Changes

To solve the issue of large datasets, the occlusion should be done at random, like other input transformers. With this more methods of occlusion can be attempted. Shape can be altered by using a random binary mask. Mask operators could be used to further refine the images segmentation to either a solid style or a style with holes which could relate to trees and leaves. Color could also be randomly chosen to reduce the statistical relation of black to irrelevancy in the model. This can be extended with textures as well. Also for testing these methods should be compared with natural samples off occlusions as the main goal is improve classification performance in the real world. XAI methods should be used in the analysis of results. Methods like SAGE [3] allow for importance to be given to each superpixel for the models predictions. This can be used to visually see if the model is learning to consider a wide area of the image or just focusing on one spot.

5.2 Future work

Both the CNNs presented in this work and the primate visual system are adept at recognizing objects despite occlusion. Are they using similar methods to accomplish the same task? CNNs and the visual system have been observed to share other common mechanisms [11-13], and the same may apply with regards to overcoming challenges related to occluded images. A particular brain region of interest is inferior temporal cortex (IT). IT is a region late in the primate ventral stream which is closely associated with object categories. Occlusion studies in IT have found that some neurons respond more strongly to occluded objects than unoccluded objects [14]. Similar units may exist in CNNs trained on occluded images. This could be determined by presenting the same stimuli used to study IT to the CNNs that we trained and analyzing the responses of individual units. If units preferentially sensitive to occluded images exist in CNNs, then a common mechnaism for recognizing occluded objects may be shared between CNNs and IT.

One of the challenges faced by deep learning is understanding the computations of CNNs. After the first convolutional layer, extracted features become increasingly complex and difficult to describe succinctly. One method to determine what these features are is to present many images and identify which images are drive the strongest activation for a given unit [15]. Then an occluder can be applied to those images at various points to identify the region of the image which driving the largest portion of that activation. How do the results of this process differ for networks already trained on on occluded datasets? Are their activations still strong despite occlusion or are the features they detect entirely different? Answering these lingering questions will also help to determine the mechanisms underlying robust performance on occluded images.

Acknowledgments

The authors would like to thank Linda Shapiro, Kechun Liu, Meredith Wu, and Mehmet Saygin Seyfioğlu for teaching a wonderful and course and fruitful discussions.

References

References

[1] V. Petsiuk, A. Das, and K. Saenko, “RISE: randomized input sampling for explanation of black-box models,” *CoRR*, vol. abs/1806.07421, 2018.

[2] C. Frye, D. de Mijolla, L. Cowton, M. Stanley, and I. Feige, “Shapley-based explainability on the data manifold,” *CoRR*, vol. abs/2006.01272, 2020.

[3] I. Covert, S. M. Lundberg, and S. Lee, “Understanding global feature contributions through additive importance measures,” *CoRR*, vol. abs/2004.00668, 2020.

[5] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images.

[6] Le, Y., Yang, X.S. (2015). Tiny ImageNet Visual Recognition Challenge.

[7] Simonyan K & Zisserman A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. doi:10.48550/arxiv.1409.1556.

[8] Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg & L. Fei-Fei (2014). *ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision* **115: 211-252**.

[9] He, K., X. Zhang, S. Ren & J. Sun (2015). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **770-778**.

[10] Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. ArXiv abs/1404.5997.

[11] Nguyen A, Dosovitskiy A, Yosinski J, Brox T & Clune J. (2016) Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. arXiv.org.

[12] Pospisil DA, Pasupathy A & Bair W. (2018) “Artiphysiology” reveals V4-like shape tuning in a deep network trained for image classification. *eLife*. **7**. doi:10.7554/eLife.38242.

[13] Flachot A & Gegenfurtner KR. (2021) Color for object recognition: Hue and chroma sensitivity in the deep features of convolutional neural networks. *Vision research (Oxford)*. **182:89–100**. doi:10.1016/j.visres.2020.09.010.

[14] T. Kim, W. Bair & A. Pasupathy (2022) Perceptual Texture Dimensions Modulate Neuronal Response Dynamics in Visual Cortical Area V4. *Journal of Neuroscience*. **42(4), 631-642**.

[15] Zhou B, Khosla A, Lapedriza A, Oliva A & Torralba A. (2014). Object Detectors Emerge in Deep Scene CNNs. doi:10.48550/arxiv.1412.6856.