Radboud Universiteit Nijmegen

Faculty of Social Sciences - Artificial Intelligence

Using a Secondary Network for Foveation in Computer Vision

Bachelor Thesis Artificial Intelligence

Author:
Johan van den Heuvel,
s4770528

Supervisor:
Dr. T.C. Kietzmann

Second reader:
Dr. F.A. Grootjen

Foveation plays a important role in the human visual system, but is currently hardly used in computer vision. Even though the "input" and systems used in computer vision are quite different from human vision, it is plausible foveation provides some benefits. In order to investigate this I compared classification losses of foveated images and non-foveated images. In addition I trained a Deep-Q Network to learn which foveation locations are optimal for a given image. Results show that there is a benefit of foveation, and that it is possible for a computer vision model to learn which foveation locations are optimal.

January 31, 2020

1 Introduction

Computer vision is a classic Artificial Intelligence problem, and as the name suggests concerns itself with vision. Vision contains a number of sub-problems such as object detection, motion tracking, action recognition, human pose estimation, and semantic segmentation. However, in this thesis I will focus on image classification. The goal of image classification is for a given image decide which class it belongs to, e.g. when shown a picture of a dog the class is "dog" and not a "cat". One of the best performing models in image classification, and in computer vision in general, is the Convolutional Neural Network (CNN). A CNN is a type of Artificial Neural Network (ANN). The development towards ANNs started in the 1940s. In 1943 McCulloch and Pitts [10] tried to understand how basic units, i.e. neurons, could process complex features. A couple of years later Donald Hebb proposed the Hebbian Learning rule [3]. In the following decades there were a number of important breakthroughs, leading up to LeNet [7] in 1990. LeNet showed that the CNN performed well on a computer vision task. In 2006 Deep Belief Networks [4] were proposed by Hinton, which was a important step towards deeper ANN's. In 2012 AlexNet [6] was a very successful combination of the deeper ANN's and the CNN structure resulting in the Deep Convolutional Network (DCN). However, even though DCNs perform great on a lot of computer vision tasks, they still have great trouble with certain tasks that are easy for humans.

This difference is not surprising as CNN's are in some aspects similar to human vision, they are quite different in others. There are differences trough-out the whole visual system, but in this thesis I will only discuss a difference of input. The human visual system does not receive a number of pixels as input, but is dependent on the eyes. The eye has two parts of vision: peripheral vision which is blurry, and foveal vision which allows us to see in great detail at a fixation point. By using multiple fixation points humans piece together a clear image, like shown in Figure 1. In current computer vision this distinction is hardly used and images are given as input in full detail.

It is hard to tell to what extent applying foveation to computer vision is beneficial as humans are products of evolution and have much bigger computational constraints. In this thesis I would like to investigate if it is possible for a small DQN to learn what locations are best to foveate at. If this is possible it opens up other interesting investigations like figuring out what the DQN learns to focus on, or even try to improve classification scores.

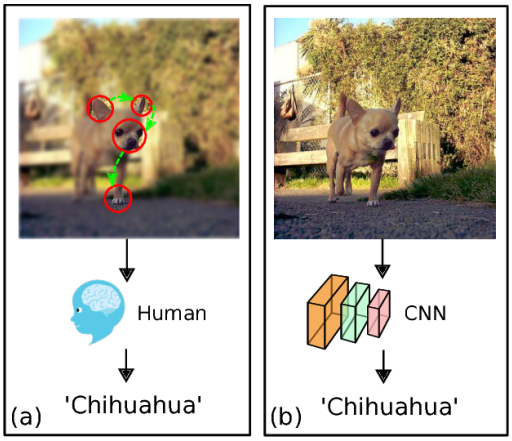
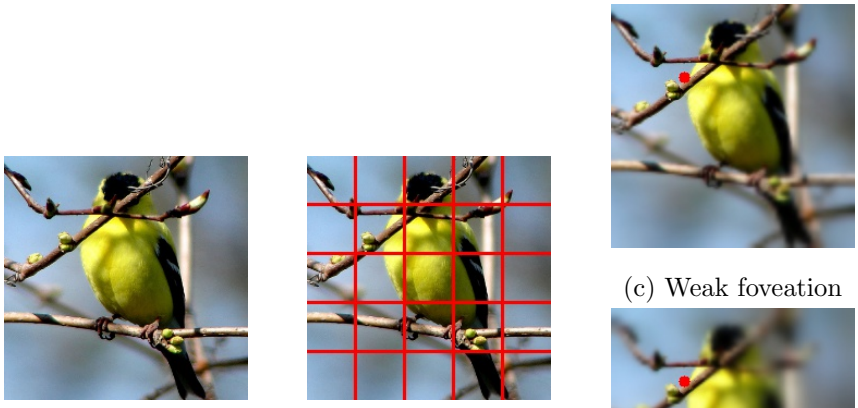


Figure 1: *Example of the difference between human and computer vision. On the left it shows how a human might use several foveation points to get a clear image of the dog. Figure from "Learning Where to Fixate on Foveated Images" [15].*

2 Methods

In the following section I will present the methods used to answer the question if a small DQN can learn to decide what the best foveation point is. The whole pipeline consists of two separate parts, one to generate the target data and another for the prediction. The images used in this research are from ImageNet [2], [12], [1], [9]. ImageNet consists of a 1000 classes, and each class has 1300 images. ImageNet also contains separate testing and validation images, those are not used. In this research a subset of classes is used, where the 1300 images of each class are split into 1200 training images, and 100 testing images. ImageNet is chosen because the images are high resolution (224*224), and commonly used which is useful for comparing results.



(a) No foveation

(b) Actions

(c) Weak foveation

(d) Strong foveation

Figure 2: *Different stages of pipeline. Figure 2a shows how images would normally look like. Figure 2b shows how each image is divided into actions, of which the DQN picks one. Figure 2c and Figure 2d show resulting foveated images for an action, with a difference between them being caused by different foveation parameters. These foveated images are fed into the image classifier which is used to score how well the DQN is doing.*

First the data is resized and cropped to a resolution of 224*224 using CV2 and interlinear interpolation. In order to generate the target data the images are first foveated and then forwarded into a pre-trained visual classifier. How the target data looks like can be seen in Figure 3. The foveation method is an implementation [1] of image retina transformation [5][11] based on Gaussian blurring. There are other foveation methods which offer a closer approximation of the human retina. However, staying closer to the original images makes it easier to understand what the network is doing. Another consideration was the ease of implementation. For the foveation two sets of parameters are used. The difference in the resulting foveation can be seen in Figure 2. Foveated images are forwarded into the pre-trained visual classifier, in this research MobileNet is used [13]. MobileNet is a computer vision network designed to perform on mobile, thus having low computational requirements. It seems that the performance of MobileNet

[1] https://github.com/ouyangzhibo/Image_Foveation_Python

is adequate for this research. MobileNet is chosen because of it's low computational requirements which is helpful for fast iteration during testing and needed because of limited access to computational resources.

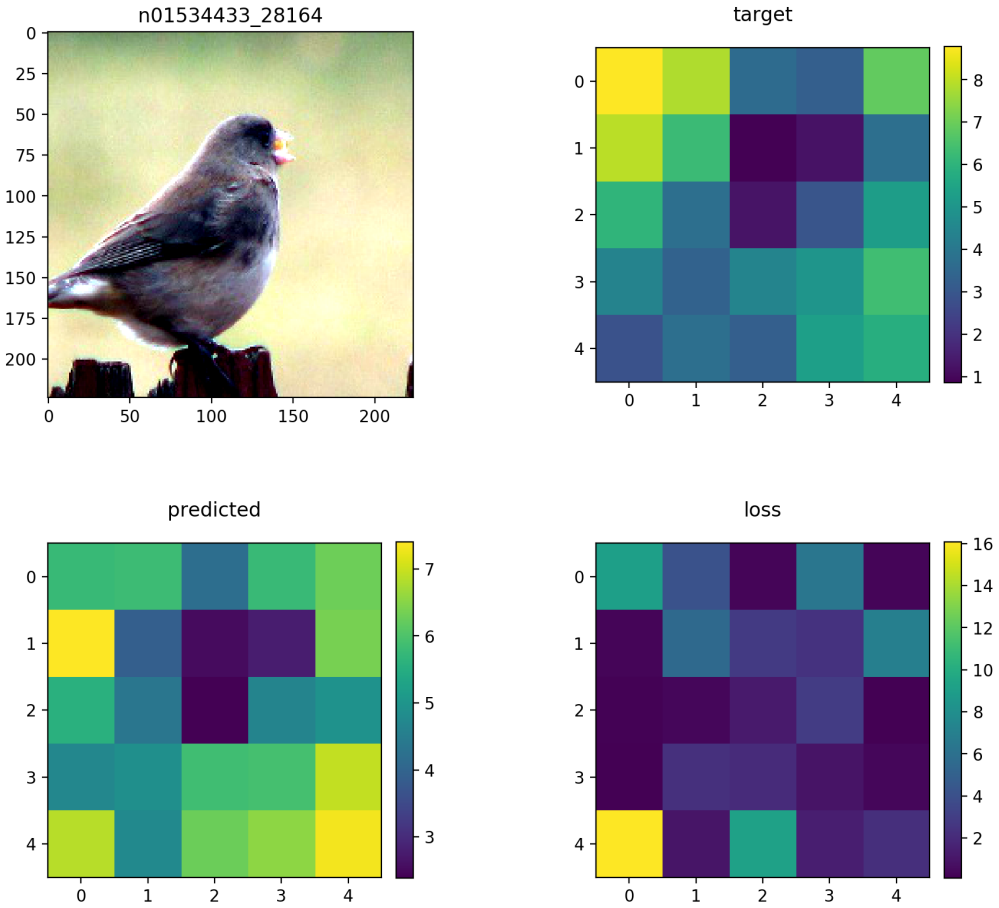


Figure 3: *Top left image is the input, the name above it is the sample name from ImageNet. Target matrix is the cross-entropy loss of MobileNet classification score at each foveation location. Predicted matrix is the output of the Deep-Q network. The loss matrix is the MSE between the target and predicted matrix. Colors are proportional to the MSE loss, i.e. darker is better. Note that during training only the MSE at a single foveation point is used, not the whole matrix.*

In order to generate predictions the images are fed into a Deep-Q Network (DQN). A DQN is used because this allows for the combination of a DCN and the reinforcement learning paradigm. This network predicts the cross-entropy loss for each of the 25 foveation locations. Then the action, i.e. foveation point, is decided by taking the minimum of these 25 losses. At this point it calculates the Mean Squared Error (MSE) between the predicted loss and actual loss at the selected foveation point, the MSE is used to optimize the DQN. In Figure 3 this would mean that the action would be the foveation point (2,1), and the MSE at that point can be seen in the loss matrix. The network architecture of the DQN can be found in Table 1. In order to assess if this network is of decent size for this task I used it to classify the training data of 5 classes. It reached 70% test accuracy on this task after 3 epochs, which indicates it should be able to process the images to some extent.

Layer (params)	Activation shape
Conv2d (k=3, s=1, p=1)	(244, 244, 3)
BatchNorm2d	-
ReLU	-
Conv2d (k=5, s=1, p=2)	(244, 244, 32)
BatchNorm2d	-
ReLU	-
Conv2d (k=5, s=0, p=2)	(244, 244, 64)
BatchNorm2d	-
ReLU	-
Conv2d (k=5, s=0, p=2)	(110, 110, 128)
BatchNorm2d	-
ReLU	-
Conv2d (k=5, s=0, p=2)	(53, 53, 256)
BatchNorm2d	-
ReLU	-
Dropout2d (p=0.2)	-
Linear	(25*25*512,1)
Linear	125

Table 1: *DQN architecture.*

3 Results

3.1 Foveation in general

In the following section I will compare foveated images against non-foveated images, using the cross-entropy loss. The losses for the non-foveated images are obtained by forwarding them trough MobileNet. For each non-foveated image there are 25 foveated images, one for each possible foveation location. These foveated images are also forwarded trough MobileNet. The end result is 26 losses for each image, 1 non-foveated loss and 25 foveated losses. The foveation used here is the weak foveation as shown in Figure 2c as strong foveation results in worse performance in general.

In order to show that foveation improve image classification I gathered the % of non-foveated images for which foveation reduces cross-entropy loss. The results can be seen in Figure 4. This shows that there is a negative relation between loss and % of images that benefit from foveation, i.e. the higher loss an image has the more likely that foveation will reduce this loss.

To get a better understanding of foveation I collected losses of 12 high loss images, the results of this can be seen in Figure 5. On almost all the images the best performing foveation location is better then non-foveated, especially on the images where the mean is already better. On these images the best performing foveation locations have down to half the loss of the non-foveated image. It seems unlikely that this increase in performance is due to blurring alone, as the worst performing locations are worse then the non-foveated images. So the distinction between clear and blurred seems to be crucial in performance, and there is quite a big decrease in loss when these images are foveated correctly.

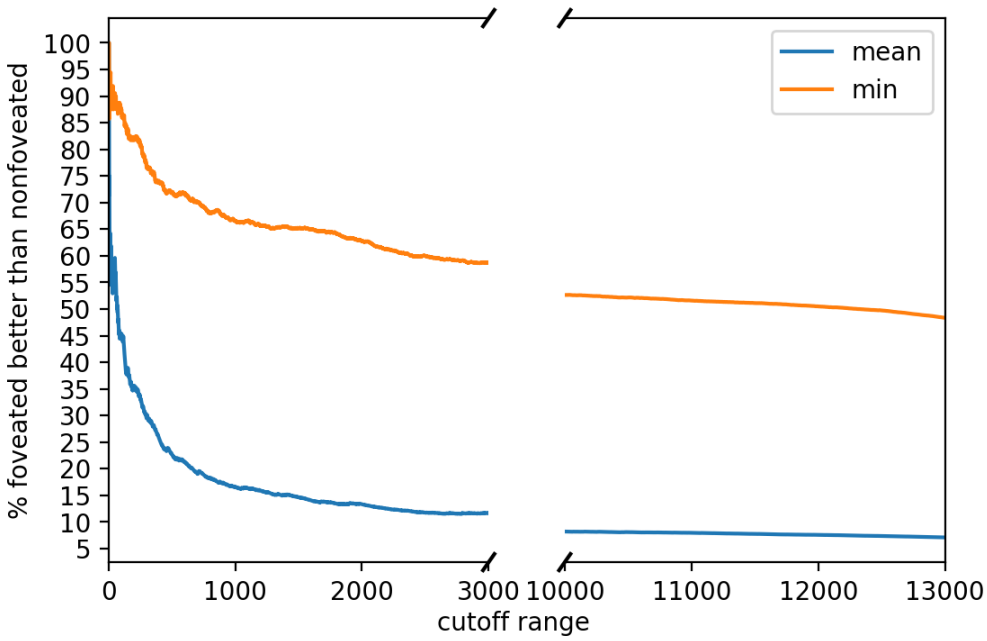
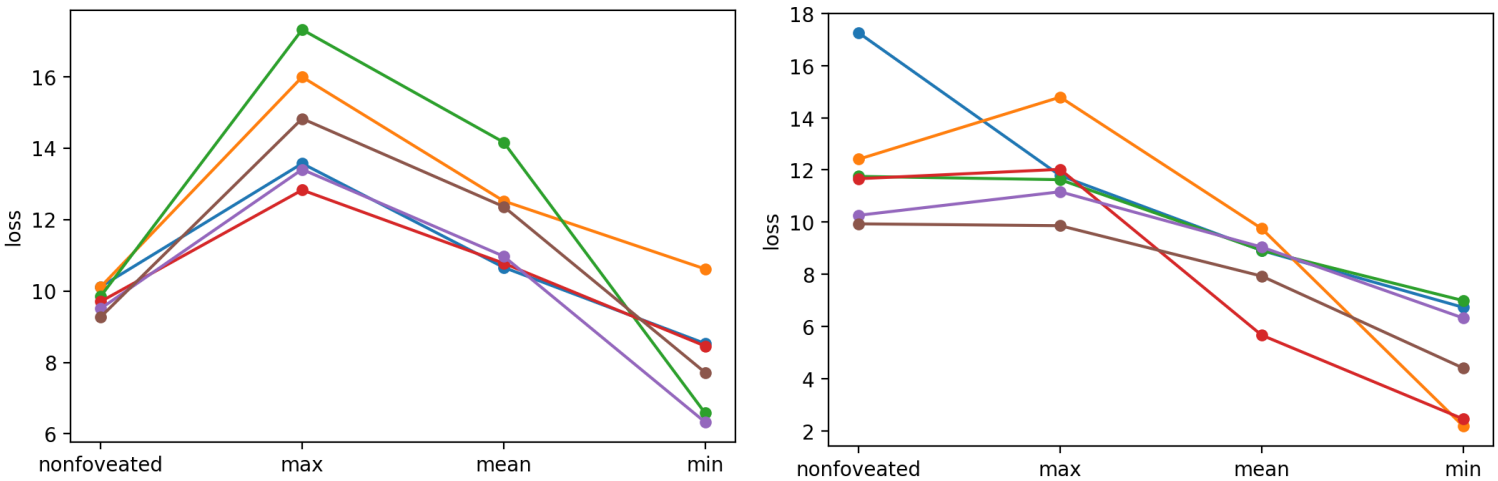


Figure 4: *This Figure show the % of foveated losses, either the mean or the minimum, that are better then non-foveated losses. The x-axis is sorted based on non-foveated image loss, from worse to best. E.g. for the 2000 worst performing non-foveated images around 15% of images foveated mean performance, i.e. average performance over the 25 location, is better then non-foveated performance.*



(a) Worse Mean performance

(b) Better Mean performance

Figure 5: *Some examples of ImageNet images with high cross-entropy loss. As can be seen in the Figures the losses for foveated images can be a lot lower then non-foveated images.*

3.2 Using an DQN to learn where to foveate

In order to show that the DQN is learning, the results of a trained DQN are compared with taking random actions. To test how well the network generalizes to different classes I also tested it on samples from 10 different classes then the 10 classes used in training. To further measure the performance I also compared it with taking the center action. This location is most often the best choice, or very close to it in both location and score. Note that when I mention random, center, or predicted values I mean the real losses associated with taking actions randomly, center, or according to lowest predicted loss respectively. Here strong foveation is used, as shown in Figure 2d. The network was trained for 100 epochs on 10 classes. For the random actions an average over 30 runs is used. Statistical analysis was done on 1000 test samples using the Wilcoxon signed-rank test. This test has the null-hypothesis that the median of the differences, $d_i = x_i - y_i$, is zero ($H_0 : d_M = 0$, $H_1 : d_M \neq 0$). with The results can be seen in Table 2.

Comparison	Classes	Statistic	Two-sided p-value
Predicted vs. random	Same as training	56887	<.001
Predicted vs. center	Same as training	180376	.0035
Predicted vs. random	Different from training	109685	<.001
Predicted vs. center	Different from training	172127	.0746

Table 2: *Results from statistical testing using the Wilcoxon signed-rank test. Note that for significance I used a threshold of .05.*

In both test cases, using either same or different classes then trained on, the predicted values are significantly different from the random values. I followed this up with a one-sided test which has the null-hypothesis that the difference has a positive median ($H_0 : d_M > 0$), and the alternative-hypothesis ($H_1 : d_M < 0$) that the difference has a negative median. The results are as follows for same-class data: (statistic=58030, p-value=<.001), and for different-class data: (statistic=109685, p-value=<.001). So for both we can reject the null-hypothesis that the difference has a positive median, and can conclude that the DQN does better then random.

When comparing the predicted- and center-values on same-class test data the two-sided p-value is significant, i.e. we can reject the null-hypothesis that the median of the differences between the predicted- and center-values is zero. To investigate if the median of differences is positive or negative I applied the same one-sided test as described earlier, the result of this test is: (statistic=180376, p-value=.0017). This has a significant p-value and thus the null-hypothesis, that the median of difference is positive, can be rejected. We can conclude that in this case the DQN is doing better then the center action. In the case of not using same-class test data, the Wilcoxon test on the predicted- and center-values has a non-significant p-value, so we cannot reject the null hypothesis that the median of differences between predicted- and center-values is zero. So it seems that the DQN doesn't perform either significantly better or worse then center on the different class test set.

4 Discussion

I found that in some cases foveation can improve classification on ImageNet, this shows that in principle foveation could improve computer vision. An interesting discovery is that there is a negative relation between classification loss and % of images that perform better when foveated. Images that on average perform better when foveated, do often perform worse at the worst foveation spot. This suggests that the improvement due to

foveation is because of the difference between blurred and non-blurred, and not as a result of blurring in general. I also found that it is possible for a DQN to learn which locations to foveate at. The predicted values are significantly better then random which indicates that the network learns. The predicted values are in general a result of taking actions around the center, which is reflected in that the predicted values are usually quite similar to the center values. There is also evidence that these results generalize to classes not trained on.

One of the limitations of these results is the simplicity of the task. In the task there are only 25 possible actions and the network doesn't take steps after the initial action. Only having 25 steps, and the fact that for a lot of images in ImageNet the class object is around the center, means that there isn't as much variability in what actions are good. This means that the network could perform quite well by just learning what actions are in general the best, and ignore the input image. I tried to counteract this problem by using strong foveation, and made sure the network doesn't do this to a problematic extend by keeping track of the distribution of actions. An additional benefit of doing this is that it allowed me to make sure the network wouldn't go for the center to much, as that is most often the best or at least a very good choice. Additionally, the network doesn't take multiple steps. Because I'm using the reinforcement learning paradigm for this thesis it would translate quite easy to a task where the network performs a sequence of actions. By not using sequences of foveation locations there are some comparisons and conclusions I cannot make. The way I would increase task difficulty can be seen in Figure 6. Increasing task difficulty by both increasing number of actions and taking sequences of tasks this research could also approximate similar research done with actions in a continuous domain, e.g. [15].

(a) More actions (b) Multiple actions

Figure 6: *These are two possible ways to make the task more difficult. On the left it shows 49 actions instead of 25, and on the right it shows how taking multiple actions after each other would look. In order to compare them to the task used in this thesis look at Figure 2. These changes to the task would go together very well, as having more possible actions allows for more "movement" around the image.*

I used MobileNet to get the classification loss for the images. This makes it harder to compare results to computational neuroscience research, where networks like VGG are often used. There are several reasons why I think MobileNet gives a very good indication of VGG results. First of all MobileNet [13] and VGG [14] have very similar top-1 error on ImageNet. Secondly the plots generated using VGG losses are extremely similar to the ones generated using MobileNet losses, e.g. look at Figure 4 and Figure

7. However, to be actually able to compare results to other research that uses VGG I would have to do the experiments using VGG loss.

Even though the network does learn to foveate on different locations for different images, there is a lack of understanding what the underlying mechanics of this are. There is active research in what kind of features are used in vision, e.g. [8], and analysing what the DQN in this research uses to decide foveation locations could be a contribution.

A limitation concerning the foveation, is that it is unclear what exactly causes foveation to reduce loss. Figure 8 shows 6 examples of images that on average benefit from foveation, and 6 images that don't. These examples show that it might be quite hard to get a firm grasp on the exact reasons why foveation sometimes works.

The main contribution of this thesis is to provide a proof of concept. It shows that for at least one foveation method on ImageNet, classified by MobileNet, some images perform better when foveated. It also shows that an DQN can learn to predict foveation locations better then random. Although these results are minor, it shows that in principle foveation can be used to improve performance in computer vision tasks. These results should encourage further research into what the exact mechanics are that result in foveation improving classification scores, and what kind of features the DQN uses to direct the foveation location.

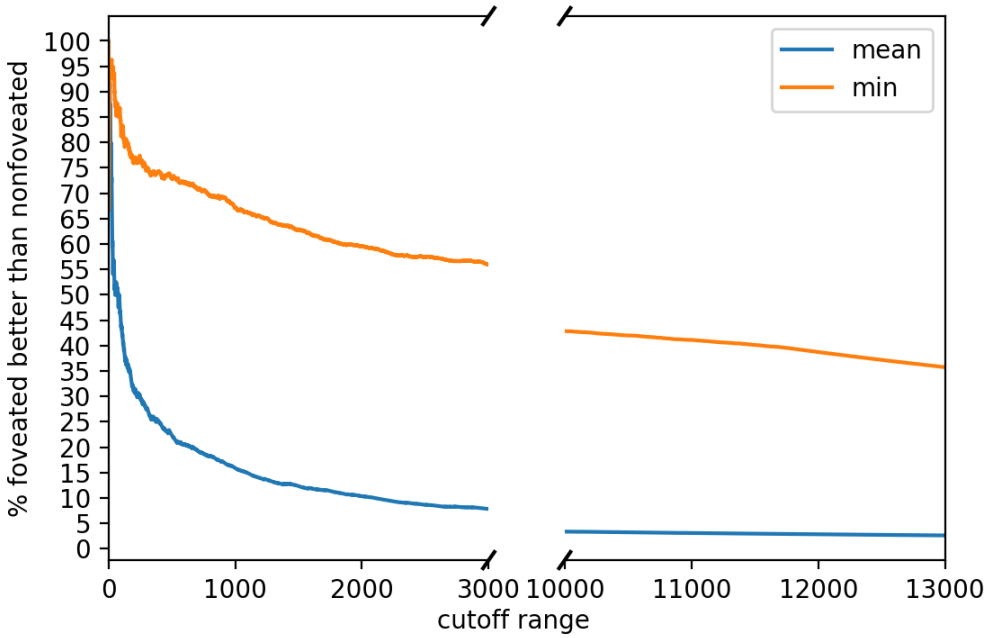


Figure 7: *The same as Figure 4, except using VGG instead of MobileNet losses.*

(a) Worse mean performance

(b) Better mean performance

Figure 8: *12 images with high loss, 6 of which benefit on average from foveation and 6 which don't. These are also the images for which scores are shown in Figure 5a and Figure 5b.*

5 Acknowledgements

I thank AcademicTorrents.com for making data available for my research.

References

[1] Joseph Paul Cohen and Henry Z. Lo. "Academic Torrents: A Community-Maintained Distributed Repository". In: *Annual Conference of the Extreme Science and Engineering Discovery Environment*. 2014. DOI: 10.1145/2616498.2616528. URL: http://doi.acm.org/10.1145/2616498.2616528.

[2] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[3] Donald Olding Hebb. *The organization of behavior: a neuropsychological theory*. Science Editions, 1962.

[4] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[5] Ming Jiang et al. "Salicon: Saliency in context". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1072–1080.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[7] Yann LeCun et al. “Handwritten digit recognition with a back-propagation network”. In: *Advances in neural information processing systems*. 1990, pp. 396–404.

[8] Drew Linsley et al. “What are the visual features underlying human versus machine vision?” In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 2706–2714.

[9] Henry Z. Lo and Joseph Paul Cohen. “Academic Torrents: Scalable Data Distribution”. In: *Neural Information Processing Systems Challenges in Machine Learning (CiML) workshop*. 2016. URL: http://arxiv.org/abs/1603.04395.

[10] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[11] Jeffrey S Perry and Wilson S Geisler. “Gaze-contingent real-time simulation of arbitrary visual fields”. In: *Human vision and electronic imaging VII*. Vol. 4662. International Society for Optics and Photonics. 2002, pp. 57–69.

[12] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge (V2017)”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[13] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4510–4520.

[14] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[15] Hanxiao Wang et al. “Learning Where to Fixate on Foveated Images”. In: *ArXiv* abs/1811.06868 (2018).