Foveated Convolutions: Improving Spatial Transformer Networks by Modelling the Retina

Ethan Harris* **Mahesan Niranjan*** **Jonathon Hare***

Abstract

Spatial Transformer Networks (STNs) have the potential to dramatically improve performance of convolutional neural networks in a range of tasks. By 'focusing' on the salient parts of the input using a differentiable affine transform, a network augmented with an STN should have increased performance, efficiency and interpretability. However, in practice, STNs rarely exhibit these desiderata, instead converging to a seemingly meaningless transformation of the input. We demonstrate and characterise this localisation problem as deriving from the spatial invariance of feature detection layers acting on extracted glimpses. Drawing on the neuroanatomy of the human eye we then motivate a solution: foveated convolutions. These parallel convolutions with a range of strides and dilations introduce specific translational variance into the model. In so doing, the foveated convolution presents an inductive bias, encouraging the subject of interest to be centred in the output of the attention mechanism, giving significantly improved performance. The code for all experiments is available at `https://github.com/ethanwharris/foveated-convolutions`.

1 Introduction

In nature, visual attention is the movement of the eyes [25], and often body [12], which enables us to process intractable quantities of information by restating the problem as one of time rather than bandwidth or functional capacity. In general, one can deal with a simplification by considering the eyes to act as one perceptive unit since stereopsis has only a limited affect [20]. The validity of this approach can be easily verified should the reader close one eye and note that depth perception broadly continues at the usual level. We will draw on a specific treatment of convolutional networks which supposes that each convolutional filter can be seen to have functional expressivity similar to that of a retinal ganglion cell. The tiling of this single filter over the image is accounted for by the spatial redundancy observable throughout the human visual system as a space preserving retinal mapping (retinotopy) [21]. This view is validated by recent work which showed that early convolutional filters can exhibit the same center surround receptive field as efferent fibers in the human optic nerve [15]. However, despite these similarities, humans still exhibit localisation performance that is vastly superior to convolutional models augmented with visual attention mechanisms.

In this work, we show that the Spatial Transformer Network (STN) [13], a popular deep model of visual attention [1], fails to perform in a visual search problem (scattered CIFAR-10: a CIFAR-10 [14] image is placed randomly on a large blank canvas and the network is required to classify it) that is trivial to a human observer. Although many components (for example, a visual memory [4]) contribute to the success of human visual attention we argue that one very simple change, foveation, can enable deep networks augmented with STNs to approach human-like localisation performance. Specifically, we introduce the foveated convolution, a layer which induces learning of powerful attention policies, solving the localisation problem. These policies are learned without any enhancement to

*Vision, Learning and Control Group, Electronics and Computer Science, University of Southampton, {ewah1g13, mn, jsh2}@ecs.soton.ac.uk

1st Workshop on Shared Visual Representations in Human and Machine Intelligence (SVRHM), NeurIPS 2019.

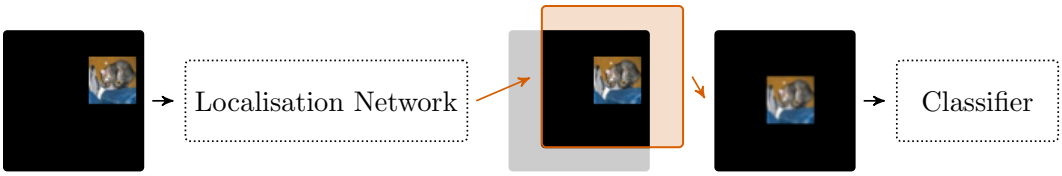

Figure 1: The basic network architecture for a translation-only STN. At each step, the output of the localisation network is used to construct a sampling grid. This is then interpolated over the image to obtain the classifier input.

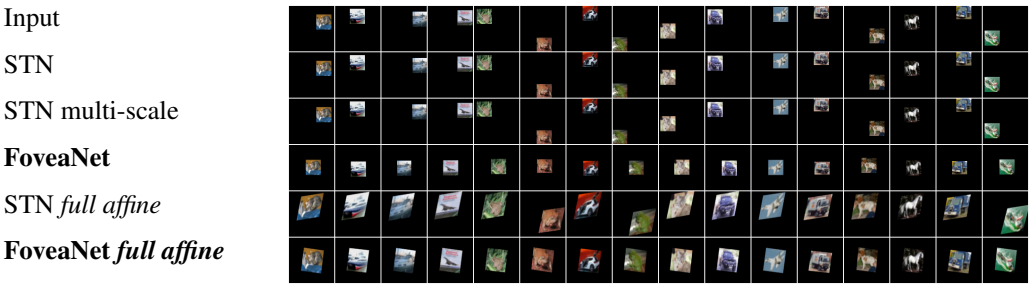

Figure 2: The transformed results, after training on scattered CIFAR-10.

or supervision of the localisation network (as in Ablavatski et al. [1]) or increase in the total number of parameters. Furthermore, we show that the foveated convolution encourages attention mechanisms to centre the region of interest in the output. Finally, we show that this unsupervised localisation network can be used in tandem with a PreAct ResNet-18 [10, 11] to completely solve scattered CIFAR-10, even marginally improving performance over the same network on vanilla CIFAR-10. Mean and standard error in all experiments are reported from 5 trials. Code, implemented using PyTorch [19] and Torchbearer [9] is available at `https://github.com/ethanwharris/foveated-convolutions`.

2 The Localisation Problem: Scattered CIFAR-10

Translation invariance is a property long thought desirable in image processing models [18]. A model can be considered translation invariant if the quality of its inference is not altered by translations of its inputs. However, translation invariance is not always desirable; if the feature extraction following a spatial attention mechanism is translation invariant, there will be no motivation for the model to learn an interesting policy. By virtue of their action over the input, convolutional networks exhibit a degree of translation invariance. This makes spatial transformer networks followed by standard convolutional layers struggle to localise the input in our problem. The architecture of a translational STN, where the attention mechanism is limited to only translating the input, is given in Figure 1. To perform well on scattered CIFAR-10, the attention mechanism will need to learn to align the images. If that alignment is made to a repeatable point, such as the centre of the image, the challenge becomes effectively solved. Specifically, one can envisage a two stage process where the attention policy is first learned using a simple classification network and then fixed to act as a pre-processor for a more complicated network. In this setting, if the localisation network (which decides the glimpse parameters) performs well, it should be possible to get the same accuracy as can be obtained on vanilla CIFAR-10 with any architecture.

The second row of Figure 2 shows the transformed images emitted by the attention mechanism of a simple CNN with a translational STN over the input (top row). The classification error for this model and a non-attentional baseline with the same classification architecture are given in Table 1a. This model uses a localisation network of four convolutional layers (kernel sizes of 7, 5, 3 and 3) followed by a linear layer with 32 neurons and a final linear layer which regresses the 2 translation parameters. Following the translational affine transformation, a three layer convolutional network (kernel size of 3 and stride of 2 in each layer), 128 neuron linear layer and final 10 neuron softmax layer perform the classification step. We use ReLU nonlinearity throughout and train for 50 epochs with the Adam optimiser (initial learning rate of 0.0001). From the figure we can see that the model

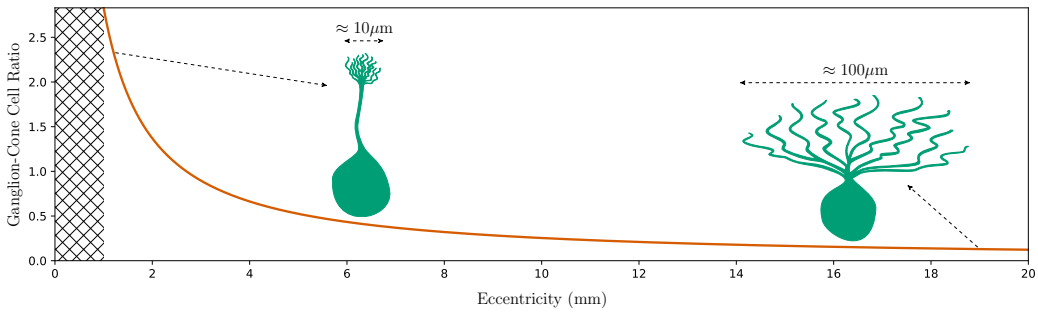

Figure 3: Ratio of ganglion to cone cell density as a function of eccentricity from the fovea. Approximate ganglion and cone cell densities taken from Curcio and Allen [6] and Curcio et al. [7] respectively. Dendritic field size is based on that of midget ganglion cells described by Dacey and Petersen [8]. The hatched region represents the foveal center which we do not address in this work.

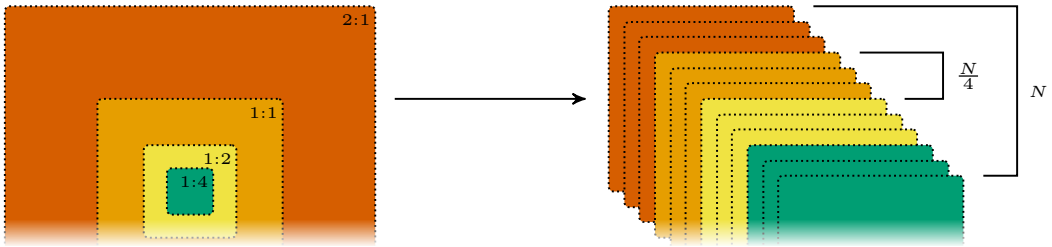

Figure 4: The foveated convolution operation. Left shows the crop area and 'input : output' pixel ratio for each convolution, right shows the output tensor for a foveated convolution with N channels. To add this component to the model from Figure 1 we replace the first layer of the classifier with a foveated convolution.

fails to appropriately localise the input, instead collapsing to zero translation in both directions. It could be argued that the STN performs poorly as a result of the constraint that it can only translate the input. However, as demonstrated in the fifth row of Figure 2, the network still fails to localise in this setting, whilst adding unnecessary rotation and scale. In order to approach a solution, we need to construct models with a counter-intuitive property: translation variance.

3 Foveated Convolutions

In this section we draw inspiration from the distribution of ganglion cells on the retina in order to construct a suitable foveation mechanism which solves the localisation problem. We can describe visual acuity in the retina in terms of the relative densities of cone and ganglion cells at varying degrees of eccentricity from the fovea. We will also need to consider the dendritic spread of ganglion cells in order to establish some understanding of their field of view. An approximation of these statistics with a visual demonstration of the dendritic spread is given in Figure 3. From the figure we can see that oversampling occurs near the fovea with a ratio of two or more ganglion cells for each cone cell. This falls off rapidly until, at the periphery, there are nearly ten cone cells for each ganglion cell. At the same time, dendritic spread increases by a factor of around ten. Following these observations, we now construct a convolutional layer, inspired by the retina, that bares a strong resemblance to image foveation; hence the name foveated convolution. We model oversampling with two transpose convolutions which operate only on the central regions of the input. We subsequently have a traditional convolutional layer with a stride of one which operates on a slightly larger region. Finally, we use layers with increased dilation and stride to model the undersampling that occurs towards the periphery. We take dilation and stride to be equal since the relative increase in dendritic spread is approximately inversely proportional to the change in ganglion-cone cell ratio. The input crop is chosen so that the output from each convolutional layer is equal and maximal (that is, the layer with the highest stride operates on the full image). Combining each of these design choices, we obtain a layer of the form depicted in Figure 4.

Table 1: Classification performance on scattered CIFAR-10

(a) Comparison of attention mechanisms

Model	Error
CNN	$58.77 \pm 0.29\%$
CNN with global pooling	$52.71 \pm 0.47\%$
STN	$60.02 \pm 2.18\%$
STN multi-scale	$57.55 \pm 2.72\%$
FoveaNet	$\mathbf{50.41 \pm 1.39}\%$
Pooled FoveaNet	$\mathbf{49.53 \pm 0.82}\%$
STN *full affine*	$47.70 \pm 1.70\%$
Pooled FoveaNet *full affine*	$\mathbf{45.83 \pm 0.61}\%$

(b) PreAct ResNet-18 with pre-processing

Model	Error
No crop	$\mathbf{6.17 \pm 0.08}\%$
STN + crop	$53.77 \pm 0.63\%$
STN multi-scale + crop	$38.66 \pm 9.08\%$
Pooled FoveaNet + crop	$\mathbf{6.13 \pm 0.08}\%$
Vanilla CIFAR-10	$9.28 \pm 0.09\%$

No crop: 10 hours, 12GB GPU memory
Others: 2 hours, 2GB GPU memory

Specifically, the output of a foveated convolution with 32 channels (the number used in our experiments) will contain 8 channels from each of four convolutional layers. The first acts on the whole image with a stride and dilation of 2. The second, acting on a centre crop of the image with half the width and height, has a stride and dilation of 1. The third layer, acting on a centre crop of the image with a quarter of the width and height, is a transpose layer with a stride of 2. The final layer, acting on a centre crop of the image with an eighth of the width and height, is a transpose layer with a stride of 4. By virtue of their stride, the dilated layers address only a fraction of the pixels in the image. Although this is appropriate since there are fewer cone cells towards the periphery, we can further model the increase in size of peripheral cone cells by swapping the dilated layer with an average pooling followed by a convolution with a stride and dilation of 1. All convolutional layers have a kernel size of 3 in our foveated layer. Example Python code for foveated and pooled foveated convolutions is given in Listings 1 and 2 from the Appendix.

The fourth row of Figure 2 shows the result of the model from Section 2 augmented with a foveated convolution layer. This has the same number of weights as the previous model but is now able to almost perfectly solve the localisation problem. Consequently, the terminal error of the model (given in Table 1a) has significantly improved. It could be suggested that this problem is better suited to a fully invariant network such as a CNN with global pooling. However, as Table 1a shows, the performance is still worse than that of a foveated convolution, and without any of the localisation benefits. The foveated layer with pooling instead of strided downsampling (Pooled FoveaNet) gives a small improvement in localisation performance and accuracy as shown in Table 1a, which also gives classification error for the STN and Pooled FoveaNet with full affine transformations enabled, again demonstrating an improvement with foveated convolutions. It should be noted that the low accuracy of the compared models is simply due to the use of a simplistic three layer classification network.

We also perform experiments where the central region is extracted from an image processed by a fixed localisation network and passed to a PreAct ResNet-18 for classification. In this setting, the full advantage of foveated convolutions becomes clear. The results, given in Table 1b, show that foveated convolutions can be used to completely solve scattered CIFAR-10, obtaining a superior result to the ResNet trained on vanilla CIFAR-10. We suspect that the reason for the increase is that the attention step acts as a small augmentation, similar to a padded random crop. As a further baseline, we measure the performance of the ResNet-18 trained on full scattered CIFAR-10 images. Although this performs very well, the increased resolution resulted in an approximately five fold increase in memory usage and training time. Strangely, the ResNet trained on scattered CIFAR-10 outperforms the same network on vanilla CIFAR-10; we again suspect that the scattering acts as an (extremely costly) augmentation.

4 Related Work

Several previous works have attempted to model the retina for various purposes. Image foveation has an established use in traditional image processing for the purpose of source coding [23]. Although these techniques bare little functional similarity to foveated convolutions, the core principle is the same; oversampling towards the center and undersampling towards the periphery. Furthermore, the notion of space variant image processing has been considered for processing data from spatially variant sources [22]. More recently, foveal mechanisms have attracted a following within deep learning for a range of tasks such as gaze prediction [26], object detection [2], video processing [24] and the automated discovery of discriminative visual elements [16]. Of particular relevance to our work are the multi-scale glimpses used by Mnih et al. [17] and the emergent foveation studied by Cheung et al. [5] which we will now consider in more detail.

Multi-scale glimpses: Early deep visual attention mechanisms used a primitive solution to the localisation problem which involves extracting the chosen region of the image at several scales before concatenating them together and passing them forward to subsequent layers [3, 17]. However, extracting a glimpse at various scales can be costly, especially when using STNs since this would require the image to be differentiably scaled multiple times on the forward pass. Furthermore, there are limitations to the localisation ability of this technique. The third row of Figure 2 shows the outcome when running our model from before, extracting glimpses at 3 scales. Clearly this is better than the naive approach, producing a lower test error than the simple STN. However, the terminal accuracy still falls short of that achieved by foveated convolutions, shown in Tables 1a and 1b.

Emergence of a fovea: In Cheung et al. [5], the authors show that an attention mechanism equipped with a learnable sampling grid exhibits emergent foveation. Specifically, the grid nodes learn to concentrate and reduce in size towards the centre of the field of view. The type of sampling this mechanism learns to perform is analogous to the anatomical inspiration for our work. The fact that this emerges in a learnable attention mechanism is a strong validation that this type of representation is desirable.

5 Summary and Future Work

To summarise, we have demonstrated a simple task, scattered CIFAR-10, which a traditional visual attention mechanism, the Spatial Transformer Network (STN), fails to solve. We have subsequently taken inspiration from biology to design the foveated convolution, a layer of stacked convolutions with a foveated receptive field. Through experimentation, we have shown that the foveated convolution induces the localisation mechanism to centre the image in the output, improving classification performance. We have then shown that this unsupervised localisation is sufficiently reliable that a PreAct ResNet-18 can be trained on top to completely solve scattered CIFAR-10 with an approximately five fold reduction in complexity over training on the full images. Future work will extend this result and find real-world visual search problems where the foveated convolution can provide an improvement and potentially a solution. Experiments will also seek to understand if the foveated convolution layer can be used to model biological phenomena that relate to visual attention.

References

[1] A. Ablavatski, S. Lu, and J. Cai. Enriched deep recurrent visual attention model for multiple object recognition. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 971–978. IEEE, 2017.

[2] E. Akbas and M. P. Eckstein. Object detection through search with a foveated visual system. *PLoS computational biology*, 13(10):e1005743, 2017.

[3] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[4] A. D. Baddeley and G. Hitch. Working memory. In *Psychology of learning and motivation*, volume 8, pages 47–89. Elsevier, 1974.

[5] B. Cheung, E. Weiss, and B. A. Olshausen. Emergence of foveal image sampling from learning to attend in visual scenes. In *International Conference on Learning Representations*, 2017.

[6] C. A. Curcio and K. A. Allen. Topography of ganglion cells in human retina. *Journal of comparative Neurology*, 300(1):5–25, 1990.

[7] C. A. Curcio, K. R. Sloan, R. E. Kalina, and A. E. Hendrickson. Human photoreceptor topography. *Journal of comparative neurology*, 292(4):497–523, 1990.

[8] D. M. Dacey and M. R. Petersen. Dendritic field size and morphology of midget and parasol ganglion cells of the human retina. *Proceedings of the National Academy of sciences*, 89(20): 9666–9670, 1992.

[9] E. Harris, M. Painter, and J. Hare. Torchbearer: A Model Fitting Library for PyTorch. *arXiv preprint arXiv:1809.03363*, 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. doi: 10.1109/cvpr.2016.90.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[12] R. Held and A. Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of comparative and physiological psychology*, 56(5):872, 1963.

[13] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[14] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[15] J. Lindsey, S. A. Ocko, S. Ganguli, and S. Deny. The effects of neural resource constraints on early visual representations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1xq3oR5tQ`.

[16] K. Matzen and N. Snavely. Bubblenet: Foveated imaging for visual discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1931–1939, 2015.

[17] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212. IEEE, may 2014. doi: 10.1109/ijcnn.2017.7965820.

[18] M. S. Nixon and A. S. Aguado. *Feature extraction & image processing for computer vision*. Academic Press, 2012.

[19] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. 2017.

[20] D. Purves and R. B. Lotto. *Why we see what we do: An empirical theory of vision.* Sinauer Associates, 2003.

[21] D. E. Purves, G. J. Augustine, D. E. Fitzpatrick, L. C. Katz, et al. *Neuroscience.* Sinauer Associates, 1997. doi: 10.4249/scholarpedia.7204.

[22] R. S. Wallace, P.-W. Ong, B. B. Bederson, and E. L. Schwartz. Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90, 1994.

[23] Z. Wang and A. C. Bovik. Embedded foveation image coding. *IEEE Transactions on image processing*, 10(10):1397–1410, 2001.

[24] J. Wu, S.-h. Zhong, Z. Ma, S. J. Heinen, and J. Jiang. Foveated convolutional neural networks for video summarization. *Multimedia Tools and Applications*, 77:29245–29267, 2018.

[25] A. L. Yarbus. Eye movements during perception of complex objects. In *Eye movements and vision*, pages 171–211. Springer, 1967.

[26] M. Zhang, K. T. Ma, J. H. Lim, and Q. Zhao. Foveated neural network: Gaze prediction on egocentric videos. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3720–3724. IEEE, 2017.

Appendices

A Example Code

Listing 1: Example PyTorch [19] implementation of a foveated convolution.

```
import torch
import torch.nn.functional as F


def foveated_convolution(image, w1, w2, w3, w4):
    h, w = image.size(3), image.size(2)

    x1 = F.conv2d(x1, w1, dilation=2, stride=2, padding=1)

    x2 = center_crop(image, (h//2, w//2))
    x2 = F.conv2d(x2, w2, stride=1, padding=1)

    x3 = center_crop(image, (h//4, w//4))
    x3 = F.conv_transpose2d(x3, w3, stride=2, padding=1,
            output_padding=1)

    x4 = center_crop(image, (h//8, w//8))
    x4 = F.conv_transpose2d(x4, w4, stride=4, padding=0,
            output_padding=1)

    return torch.cat((x1, x2, x3, x4), dim=1)
```

Listing 2: Example PyTorch [19] implementation of pooled foveated convolution.

```
import torch
import torch.nn.functional as F


def pooled_foveated_convolution(image, w1, w2, w3, w4):
    h, w = image.size(3), image.size(2)

    x1 = F.avg_pool2d(image, 2)
    x1 = F.conv2d(x1, w1, stride=1, padding=1)

    x2 = center_crop(image, (h//2, w//2))
    x2 = F.conv2d(x2, w2, stride=1, padding=1)

    x3 = center_crop(image, (h//4, w//4))
    x3 = F.conv_transpose2d(x3, w3, stride=2, padding=1,
            output_padding=1)

    x4 = center_crop(image, (h//8, w//8))
    x4 = F.conv_transpose2d(x4, w4, stride=4, padding=0,
            output_padding=1)

    return torch.cat((x1, x2, x3, x4), dim=1)
```