Egocentric Perception using a Biologically Inspired Software Retina Integrated with a Deep CNN

Piotr Ozimek Lorinc Balog Ryan Wong Tom Esparon
J. Paul Siebert
School of Computing Science
University of Glasgow, Lilybank Gardens, Scotland, UK G12 8RZ
paul.siebert@glasgow.ac.uk

Abstract

We presented the concept of of a software retina, capable of significant visual data reduction in combination with scale and rotation invariance, for applications in egocentric and robot vision at the first EPIC workshop in Amsterdam [9]. Our method is based on the mammalian retino-cortical transform: a mapping between a pseudo-randomly tessellated retina model (used to sample an input image) and a CNN. The aim of this first pilot study is to demonstrate a functional retina-integrated CNN implementation and this produced the following results: a network using the full retino-cortical transform yielded an F1 score of 0.80 on a test set during a 4-way classification task, while an identical network not using the proposed method yielded an F1 score of 0.86 on the same task. On a 40K node retina the method reduced the visual data by~×7, the input data to the CNN by 40% and the number of CNN training epochs by 36%. These results demonstrate the viability of our method and hint at the potential of exploiting functional traits of natural vision systems in CNNs. In addition, to the above study, we present further recent developments in porting the retina to an Apple iPhone, an implementation in CUDA C for NVIDIA GPU platforms and extensions of the retina model we have adopted.

1. Introduction

This paper updates progress in developing and implementing the concept of a combining a software retina with a DCNN for efficient visual processing as detailed in [7]. We argue that a key design issue for egocentric and robot vision is the ability to perform advanced computer vision while using only lightweight processing platforms. This requirement implies a need for a smart camera capable of transferring low bandwidth visual data streams and data efficient Deep CNN processing. Accordingly, as we noted at the EIPC Workshop in 2016, there is currently much work going on to adapt smartphones to provide complete portable vision systems, as the relatively low-cost and ubiquitously available smartphone is so exquisitely integrated by having camera(s), inertial sensing, sound input/output and excellent wireless connectivity. Mass market production makes this a very low-cost platform and manufacturers from quadrotor drone suppliers to childrens toys, such as the Meccanoid robot, employ a smartphone to provide a vision system/control system.

The key challenge of this work we proposed in [9] is how to transform the irregularly distributed retina samples to a matrix format that can be processed within conventional CNN environments and then to devise a CNN architecture which is compatible with this input. We demonstrate that our software retina-integrated CNN formulation is capable of learning object classes, affords a reduction in network size and also requires fewer training epochs to achieve a classification performance similar to that of a standard CNN formulation. We now outline a retina implementation running on the CPU, a GPU accelerated version and also a first demonstration e on an Apple iPhone.

We summarise a first study [7] into improving the efficiency of image analysis using CNNs by pre-processing using a software-based retina model, similar in structure to those of mammals and humans, to substantially reduce the visual input data size to the CNN and also simplify its learning requirements. Our retina model samples the input image using Gaussian receptive fields located on a space-variant (foveated) pseudo-random sampling tessellation generated by annealing. This data is then spatially transformed using a polar transform to generate a *cortical map*, similar to that observed in the human visual system. This mapping splits the visual field into two hemifields, as observed in the brain, and these are projected into a regular image suitable for processing by a Keras CNN model we formulated. This mapping not only reduces the visual data by a factor of ~×7 for a 4K node retina (~×16.7 for a 50K node retina), it also affords a degree of input image scale and rotation invariance and therefore has the potential to both reduce the network size substantially and also simplify its learning requirements.

2. Balasuriya's Retina

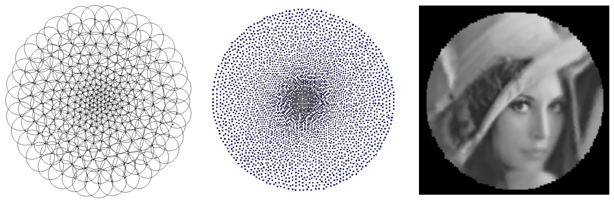


Figure 1. **Left:** Gaussian receptive fields on top of a retina tessellation, taken from [1]. **Centre:** The 4196 node tessellation used in this paper. **Right:** A backprojected retinal image.

The retina model that has been employed in this paper was developed by Balasuriya [1] whose work investigates the generation, sampling function, feature extraction and gaze control mechanism of a self-organized software retina.

To generate the retina tessellation without local discontinuities, distortions or other artefacts Balasuriya employs a self-similar neural network as described by Clippingdale & Wilson [2]. This method relies on a network of N nodes jointly undergoing random translations to produce a tessellation with a near-uniform dense foveal region that seamlessly transitions into a sparse periphery. Each node in the resultant tessellation defines the location of a receptive field's centre. The receptive fields somewhat follow the biological retina's architecture; they all have a Gaussian response profile the standard deviation of which scales linearly as a function of local node density, which in turn scales with eccentricity. This scaling balances between introducing aliasing at the sparsely sampled peripheries and super-Nyquist sampling at the densely sampled foveal region.

The values sampled by the receptive fields are then stored in an *imagevector*, which is a one-dimensional array of intensity values which supply the remainder of his visual processing chain and are also used to feed the processing pipeline in this work.

3. The Retino-Cortical Transform

The core idea behind cortical images is to first map the receptive field centres onto a new space and then project the associated imagevector intensities via Gaussian kernels centred on these locations, i.e. perform a *forward warp*. The approach taken eliminates the possibility of holes in the mapping as the size of the Gaussian projections can be increased to compensate.

The cortical images should ideally be *conformal*, i.e. preserve local angles and maintain a fairly uniform receptive field density while preserving local information captured by the retina without introducing any artefacts. These criteria must be satisfied to enable the convolution kernels of CNNs to extract features from the resultant cortical image. The literature reports sampling points at an adjusted log-polar space are the most appropriate retinocortical mapping, as it is believed they are employed in the primate visual cortex [8]. It is mathematically a plausible mapping for foveated images as it stretches out the fovea and compresses the peripheral field; it is also a conformal mapping.

4. 4K Node Retina Validation

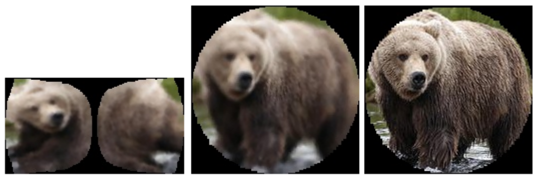
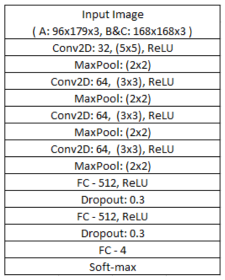


Figure 2. An example *Brown Bear* image from each of the three subsets (A, B and C).

A dataset suitable for training retina-integrated CNNs (RI-CNNs) was created by selecting and pre-processing the appropriate classes from ImageNet [3]. To prevent the classification task from being trivial, each object class in the dataset should share a subset of its visual features with at least one other class, meaning that the classes should be somewhat similar to each other.

In order to evaluate each part of the proposed retino-cortical transform in isolation, three validation subsets have been constructed: subset A is made up of cortical images (Fig. 2, left), subset B is retinal backprojected images (Fig. 2, centre) and subset C consists of the conventional images, masked with the retinal lens (Fig. 2, right).

	Training	*Eval.*	*Test*	**TOTAL**
Hoop	2560	727	372	**3659**
Brown Bear	2422	693	350	**3465**
Keybrd.	2490	711	360	**3561**
Racoon	2492	704	339	**3535**
TOTAL	**9964**	**2835**	**1421**	**14220**



Input Image (A: 96x179x3, B&C: 168x168x3)
Conv2D: 32, (5x5), ReLU
MaxPool: (2x2)
Conv2D: 64, (3x3), ReLU
MaxPool: (2x2)
Conv2D: 64, (3x3), ReLU
MaxPool: (2x2)
Conv2D: 64, (3x3), ReLU
MaxPool: (2x2)
FC - 512, ReLU
Dropout: 0.3
FC - 512, ReLU
Dropout: 0.3
FC - 4
Soft-max

Figure 3. **Left:** Per class and per split fixation image counts. The numbers are consistent across all 3 subsets of the dataset. **Right:** The CNN architecture used in this paper.

The object categories selected for the classification task are *Basketball Hoop*, *Brown Bear*, *Keyboard* and *Racoon*. The similarities between *Brown Bear* and *Racoon* (furry animal), *Basketball Hoop* and *Keyboard* (synthetic object with a grid-like key feature) helped ensure that the classification task is not trivial. The class objects were cropped out from their original images using the bounding boxes provided in ImageNet [3]. The resultant images passed automatic selection that ensured the images were not too small ($width, height > 75, 75$) or too long ($1/3 < width/height < 3$), and were then processed by appropriate parts of the retina pipeline to produce the three subsets.

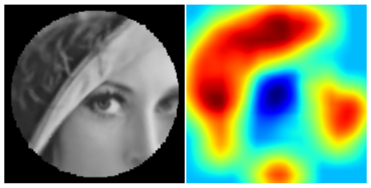


Figure 4. **Left:** A retinal backprojection image. **Right:** the equivalent saliency map. Note the inhibition near the foveal region.

The image locations fixated on by the retina were selected by a gaze control system that is both simple and sufficient to meet the needs of this study. The algorithm maintained a saliency map driven by SIFT features that was inhibited at locations of past fixations (Figure 4, right). In order to correct a large imbalance between the class frequencies the number of retina fixations was varied per class. The final image counts in the dataset can be seen in Figure 3.

5. Results and Discussion

In order to evaluate in isolation the performance contributions of the retinal subsampling mechanism and the cortical image representation to the overall pipeline, three CNNs were trained using Keras 2.0.2, each with the same architecture but each using a different subset of the dataset built in the previous section. The CNN architecture used (Figure 3) was chosen by trialling various architectures to maximise their performance over the cortical image dataset. A relatively simple architecture was chosen in accordance the objectives of this study.

Adam [6] was employed for training optimisation in combination with categorical cross-entropy as the loss function. Early stopping callbacks (used to monitor improvements in validation accuracy) were employed to prevent unproductive training. L2 regularisation of strength $\lambda = 0.02$ was applied to the internal fully connected layers to prevent over-fitting, however that value could have been increased as the model still displayed signs of over-fitting. The key figures from training are:

- **Network EVAL-A**, using (96x179) cortical images, reached its peak performance **(validation loss= 0.605, validation accuracy=82.26%)** after **16 epochs**.
- **Network EVAL-B**, using (168x168) retinal backprojected images, reached its peak performance **(validation loss= 0.493, validation accuracy=86.14%)** after **21 epochs**.
- **Network EVAL-C**, using (168x168) conventional images, reached its peak performance **(validation loss= 0.488, validation accuracy=87.51%)** after **25 epochs**.

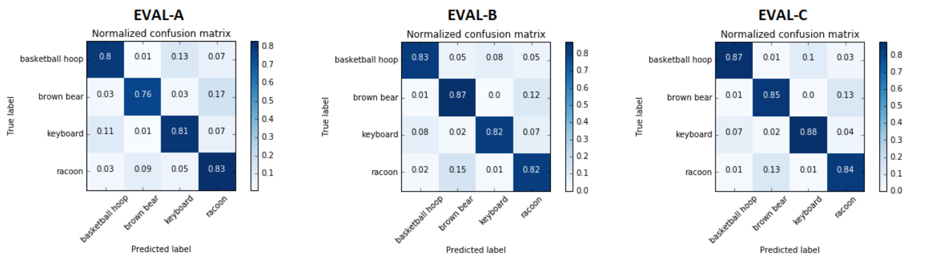
The results from evaluating the networks against the test set (Figure 5) show that both applying the full retinocortical transform and the retinal subsampling led to a small decrease in the CNNs' performance. The network trained on conventional images performed the best, with an average F1 score of 0.86; the network trained on retinal images landed an F1 score of 0.84 while the cortical images network had an F1 score of 0.80 showing that remapping the image from the retinal to the cortical space was the most damaging aspect of the retinocortical transform. As seen in the matrices in Figure 5, the majority of the networks' confusion is between the classes sharing similar key features. Although the retina has reduced classification performance, the gap between the performance scores of the different networks is modest, the required training epochs have been reduced significantly and the network EVAL-A has successfully demonstrated the learning capacity of convolutional neural networks for images in the cortical view.

6. Recent Work

Subsequent to validating the 4K node retina, as described above, we have implemented a 50K node retina capable of sampling a 930×930 pixel input image to produce a cortical image of around 150K pixels, yielding a visual data reduction of ~$\times 16.7$ and a network input reduction of ~$\times 5.8$. This 50K node retina generates a cortical image in ~13ms when executing on an NVIDIA GTX1080TI GPU, running CUDA C codes.

We have also demonstrated the 4K node retina running on an Apple iPhone. This comparatively small retina samples a patch in an image captured by iPhone's camera and SIFT descriptors are extracted from the cortical image to direct the next retinal fixation location of the next in conjunction with a simple inhibition of return algorithm. The iPhone can therefore serve as an autonomous cortical image capture device for tasks such as object appearance learning. Our next objective is to couple the retina input to a mobile Tensorflow model, trained using the iPhone retina, to support egocentric perception processing tasks.

Finally, in order to improve the overall data efficiency of the retina-DCNN combination, we have implemented a gamut of colour and intensity retinal ganglion P-cells. These include RG & BY difference cells, and RG & BY single and double opponent cells. We have also implemented the classical intensity difference of Gaussian response cells.



EVAL-A	precision	recall	f1-score	support
basketball hoop	0.83	0.80	0.81	372
brown bear	0.88	0.76	0.82	350
keyboard	0.80	0.81	0.80	360
racoon	0.72	0.83	0.77	339
avg / total	**0.81**	**0.80**	**0.80**	**1421**

EVAL-B	precision	recall	f1-score	support
basketball hoop	0.88	0.83	0.85	372
brown bear	0.80	0.87	0.83	350
keyboard	0.90	0.82	0.86	360
racoon	0.77	0.82	0.79	339
avg / total	**0.84**	**0.83**	**0.84**	**1421**

EVAL-C	precision	recall	f1-score	support
basketball hoop	0.90	0.87	0.88	372
brown bear	0.85	0.85	0.85	350
keyboard	0.88	0.88	0.88	360
racoon	0.80	0.84	0.82	339
avg / total	**0.86**	**0.86**	**0.86**	**1421**

Figure 5. Confusion matrices and different performance metrics of the three CNNs evaluated against the appropriate test sets.

These have been structured to generate the pairs of response outputs corresponding to rectified +ve & -ve responses. As consequence these cells appear to exhibit the first stages on figure-ground separation in there responses to appropriate intensity and colour differential inputs and we anticipate this will simplify learning segmentation and object boundaries.

7. Conclusions & Further Work

This work has presented a pilot study into a novel method for pre-processing images provided to CNNs. The method draws inspiration from the mammalian visual system by imitating the retinocortical transform to reduce the networks' memory requirements as well as affording a degree of scale and rotation invariance. The contributions of our work comprise: a specific retina model, a coarsely optimised cortical transform formulation and an image classification dataset comprising three subsets designed to probe the impact of the spatial sampling and transformation components of the pipeline. Evaluating a CNN architecture on the three data subsets has shown that the performance of the retina-integrated CNN is comparable to that of a CNN working with conventional images, while image data reduction, network size reduction and learning simplification has been confirmed. To the best of the authors' knowledge no prior attempts have been made at integrating a similar process to CNNs.

This work has laid the groundwork for further investigations into integrating the retinocortical transform with convolutional neural networks. The authors propose to develop a custom non-shared CNN layer which is fed directly by the retina image vector, appropriately transformed into a 2D polar retinotopic mapping. We then propose to investigate more elaborate CNN architectures which are best suited for retinocortical transformed input. Locally connected convolution layers, as well as streams of parallel convolutions each processing a separate portion of the cortical image, are both relevant features of CNN architectures that appear to be worth investigating.

Our most recent work includes modelling the known retinal ganglion cells, GPU and iPhone retina implementations, and a high-resolution 50K node retina. Our currently investigations include the wider range of low level computations that are essential to high-level visual reasoning tasks related to edge detection, motion and prediction [4] and more sophisticated gaze control algorithms, potentially based on learning and generating target specific saliency maps, as in Hong et a. [5]. We are also working towards more extensive training datasets based on both static and video imagery captured using the iPhone retina system and also a camera mounted on a robot arm to support scene exploration and hand-eye visual serving.

References

[1] S. Balasuriya. *A Computational Model of Space-Variant Vision Based on a Self-Organized Artifical Retina Tesselation*. PhD thesis, Department of Computing Science, University of Glasgow, March 2006. 2

[2] S. Clippingdale and R. Wilson. Self-similar neural networks based on a kohonen learning rule. *Neural Networks*, 9(5):747–763, 1996. 2

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 2, 3

[4] T. Gollisch and M. Meister. Eye smarter than scientists believed: neural computations in circuits of the retina. *Neuron*, 65(2):150–164, 2010. 4

[5] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, pages 597–606, 2015. 4

[6] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[7] P. Ozimek and J. Siebert. Integrating a Non-Uniformly Sampled Software Retina with a Deep CNN Model. In *BMVC 2017 Workshop on Deep Learning On Irregular Domains*, September 2017. 1

[8] E. L. Schwartz. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977. 2

[9] J. Siebert, A. Schmidt, G. Aragon-Camarasa, N. Hockings, X. Wang, and W. P. Cockshott. A Software Retina for Egocentric & Robotic Vision Applications on Mobile Platforms. In *ECCV 2016 Workshop on Egocentric Perception, Interaction and Computing*, October 2016. 1