Automatic detection and identification of retinal vessel junctions in colour fundus photography

Harry Pratt[1], Bryan M. Williams[1], Jae Ku[1,3], Frans Coenen[2], and Yalin Zheng[1,3]

[1] Dept of Eye and Vision Science, Institute of Ageing and Chronic Disease, University of Liverpool, L7 8TX,
[h.pratt,yzheng]@liverpool.ac.uk,
WWW home page: http://pcwww.liverpool.ac.uk/yzheng
[2] Dept of Computer Science, University of Liverpool, L69 3BX
[3] St Paul's Eye Unit, Liverpool Royal University Hospital, L7 8XP

Abstract. The quantitative analysis of retinal blood vessels is important for the management of vascular disease and tackling problems such as locating blood clots. Such tasks are hampered by the inability to accurately trace back problems along vessels to the source. This is due to the unresolved challenge of distinguishing automatically between vessel branchings and vessel crossings. In this paper, we present a new technique for tackling this challenging problem by developing a convolutional neural network approach for first locating vessel junctions and then classifying them as either branchings or crossings. We achieve a high accuracy of 94% for junction detection and 88% for classification. Combined with work in segmentation, this method has the potential to facilitate automated localisation of blood clots and other disease symptoms leading to improved management of eye disease through aiding or replacing a clinicians diagnosis.

Keywords: Convolutional neural networks, retinal imaging, retinal vessels fundus photography, vessel classification

1 Introduction

Vascular conditions present a challenging public health problem. They are often life-threatening and damage to blood vessels can lead to significant complications such diabetes, hypertension and stroke. The retina is the only inner organ which can be directly imaged and also serve as a window for the diagnosis of systematic diseases such as cerebral malaria, stroke, dementia and cardiovascular diseases [10]. It is therefore of great importance to better understand and be able to manage such conditions. It is also significant that pathologies can affect veins and arteries differently. For example, in diabetic retinopathy, there is veinus beading. With the availability of imaging techniques such as colour fundus photography, fundus angiography and optical coherence tomography angiography, there has been a significant need for automated vessel analysis techniques [15, 14].

There has been a considerable amount of work in recent years aimed at the effective segmentation of retinal blood vessels in fundus photography, which is a fundamental step for a vessel analysis system. Work such as [2, 14, 15] has been able to achieve increasingly improved segmentation of retinal vessels. However, a significant remaining challenge is to distinguish between vessel branchings and vessel crossings, where one blood vessel passes over another but does not connect to it. This is important for tracking vessels, separating veins from arteries and with occlusions.

When a blood clot needs to be located, we must be able to trace back along the vessel. The current inability to accurately identify vessel crossings in vessel segmentations hinders this. It is also important to monitor progress after vein and artery occlusions; being able to identify and distinguish vessel crossings and branchings facilitates this. Automating the detection and classification of vessel junctions also allows us to aid clinicians in detecting vascular abnormalities.

In this paper, we present a new hierarchical approach to first automatically determine the locations of blood vessel junctions in colour fundus images and then distinguish between vessel branchings and crossings. We employ an available segmentation of the vessel structure, although an automatic segmentation procedure could be incorporated, to identify points along blood vessels. We then develop a convolutional neural network which is trained on expert annotated data to identify vessel junctions. The same network architecture is then used and trained to learn new convolution filters to distinguish between vessel branchings and crossings. This results in a method which is capable of identifying and classifying vessel junctions without user intervention.

For applications in image analysis and classification, Convolutional Neural Networks (CNNs), a branch of deep learning, has achieved state of the art results for many problems. The 1970's saw the introduction of network architectures being used to analyse image data [4]. These had useful applications and allowed challenging tasks such as handwritten character recognition [3] to be achieved. Decades later, there were several breakthroughs in neural networks that lead to vast improvements in their implementation such as the introduction of dropout [13] and rectified linear units [11]. These theoretical enhancements and the accompanying increase in computing power through graphical processor units (GPUs) meant that CNNs became viable for more complex image recognition problems. Presently, large CNNs are used to successfully tackle highly complex image recognition tasks with many object classes to an impressive standard. CNNs are used in many of the current state-of-the-art image classification tasks including medical imaging. Hence, we use this method combined with expert segmented fundus images and skeletonisation [12][5] to detect and classify vessel junctions within fundus images.

There are many different architectures for neural networks. Recently residual networks have achieved impressive results on the highly competitive competition of ImageNet detection, ImageNet localisation, COCO detection, and COCO segmentation [6]. They were then widely used in the following 2016 ImageNet competition due to impressive performance on general large data sets of small images

such as the MNIST [9] dataset for handwritten digits 0-9 and CIFAR-10 [8] a dataset of 10 classes of color images. This makes the network ideal for our patch based method. Hence, the Res18 network structure containing 18 residual layers is used in the CNNs throughout this paper.

The rest of this paper is organised as follows. In §2, we present our new method for locating and identifying crossings and branchings of retinal vessels, in §3 we present our experimental results and in §4 and §5 we discuss this work and present our conclusions.

2 Methods

The images used to implement out method are from the Digital Retinal Images for Vessel Extraction (DRIVE) database with manual segmentations [14]. The images in the DRIVE dataset were obtained from a diabetic retinopathy screening program in The Netherlands. The images were acquired using a Canon CR5 non-mydriatic 3CCD camera with a 45 degree field of view (FOV) using 8 bits per color plane at 768 by 584 pixels.

Our framework consists of identifying patches of fundus images $z(\mathbf{x})$ and identifying those patches which include junctions, and then distinguishing the type of junction located. We make use of available vessel segmentations given as binary functions $\phi(\mathbf{x})$ defined on the domain $\mathbb{R} \times \mathbb{R}$. In practice, we are dealing with the discrete counterparts $\mathbf{z}$ and ϕ of the image and segmentation function respectively defined over the discrete domain $\Omega \subset \mathbb{Z} \times \mathbb{Z}$.

2.1 Skeletonisation and patch extraction

We consider patches of the fundus images centred along the segmented vessels. In order to restrict the number of patches for training to a manageable amount and reduce bias, we aim to reduce to segmentation of the vessels to a skeleton and consider regions centred only on these points. We achieve this by performing a skeletonisation of the level set function $\phi(\mathbf{x})$ for each image. We convolve the level set function with the kernels

$$\kappa_1^j(\alpha^1) = r_j \begin{pmatrix} 0 & 0 & 0 \\ \alpha_1^1 & 1 & \alpha_2^1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \kappa_2^j(\alpha^2) = r_j \begin{pmatrix} \alpha_1^2 & 0 & 0 \\ 1 & 1 & 0 \\ \alpha_2^2 & 1 & \alpha_3^2 \end{pmatrix}, \tag{1}$$

where r_j denotes rotation of the matrix by a multiple j of $\pi/2$ radians and $\alpha^1 = (\alpha_1^1, \alpha_2^1)^\top \in \Psi^2$, $\alpha^2 = (\alpha_1^2, \alpha_2^2, \alpha_3^2)^\top \in \Psi^3$ where $\Psi = \mathbb{Z} \cap [0, 1]$. We thin the segmentation of the vessels by removing the points which are centred on regions matching the above filters. That is, we set such points as background points. We achieve this by iterating

$$\varphi^{\ell+1} = \mathcal{F}_{i,j}\left(\varphi^\ell\right), \qquad \ell = 0, 1, \ldots, \qquad \varphi_0(\mathbf{x}) = \phi(\mathbf{x})$$

$$\mathcal{F}_{i,j}(\varphi) = \varphi - 1 + H\left(\left(\kappa_i^j(\alpha^i) * \varphi - \sum \kappa_i^j(\alpha^i)\right)^2\right), \tag{2}$$

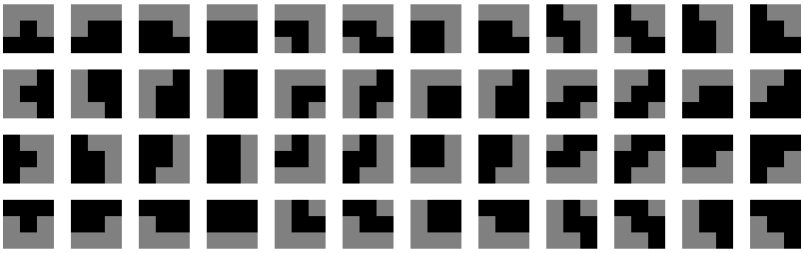


Fig. 1: Kernel functions for skeletonisation

beginning with $l_0^1 = 0$ and cycling through $i \in \{1, 2\}$, $j \in \{0, 1, 2, 3\}$.

Following this, we extract the patches by cropping the image $z(\mathbf{x})$ to 21×21 pixel windows $\Theta_{\mathbf{p}}$ centred on points $\mathbf{p}$ in the set Υ of points considered the foreground of the skeletonised vessel map. The patch size was selected so that junctions and branches in the vessel would fit within one patch. The patches are given by

$$\Theta_{\mathbf{p}} = \{\mathbf{q} \in \Omega \mid |\mathbf{p} - \mathbf{q}| \leq 10\}, \qquad \mathbf{p} \in \Upsilon = \{\mathbf{p} \in \Omega \mid \varphi(\mathbf{p}) = 1\}.$$

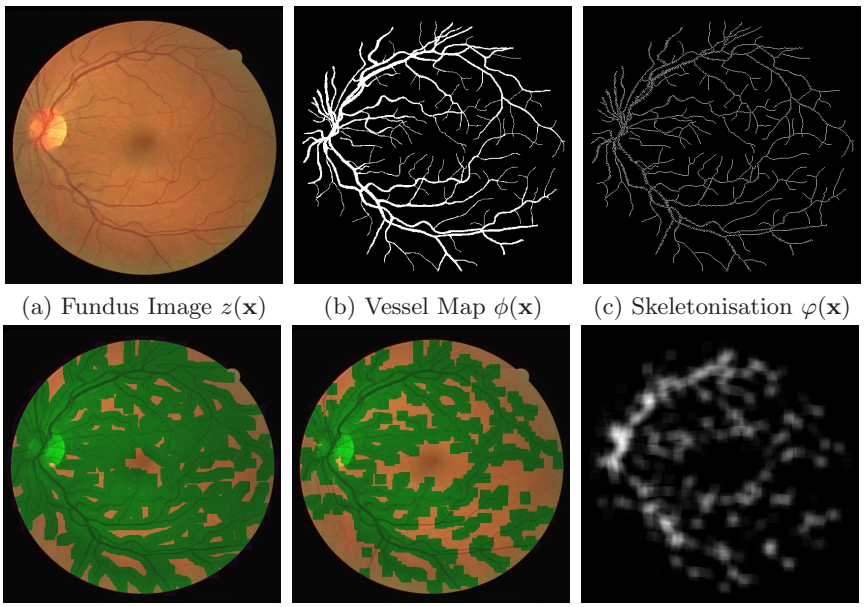
In the training stage, these patches Θ of the images in the training set are used to train the neural network to identify whether a branching or crossing is contained in the image patch. In the test stage, the trained CNN classifies the patches accordingly. This step is described below.

2.2 Junction Identification - CNN $\mathbb{C}_1$

To identify the vessel junctions within the patches created we train our CNN on a high-end graphics processer unit (GPU). The large random access memory of the Nvidia K40c means that we were able to train on the whole dataset of patches at once. The Nvidia K40c contains 2880 CUDA cores and comes with the Nvidia CUDA Deep Neural Network library (cuDNN) for GPU learning. The deep learning package Keras [1] was used alongside the Theano machine learning back end to implement the network. After training, the feed forward process of the CNN can classify the patches produced from a single image in under a second.

We use the Res18 network architecture [6] as deep levels of convolution were required to distinguish the vessel junction type in our small patches. The residual layers incorporate activation, batch normalisation, convolutional, dense and maxpooling layers. We also use L^2 regularisation to improve weight training. There were approximately 100,000 patches for training and 30,000 for testing in the junction identification problem. The classes were weighted as a ratio of junction to background due to the fact that junctions in the training and testing patches were sparse at a ratio of 1:39. The network was trained using Adam stochastic optimisation for backpropegation [7]. The network was trained to

classify the patches in to a binary classification of either vessel junction or background. Gaussian initialisation was used within the network to reduce initial training time. The loss function used to optimise was the widely used categorical cross-entropy function. Training was undertaken until reduction of the loss plateaued to obtain optimal results.



(a) Fundus Image $z(\mathbf{x})$ (b) Vessel Map $\phi(\mathbf{x})$ (c) Skeletonisation $\varphi(\mathbf{x})$

(d) Patch Boundaries (e) Patch Classification (f) Junction Location

Fig. 2: Example of first part of algorithm: locating junctions.

2.3 Locate the centres

Following the neural network classification, which tell us if a branching or crossing is contained within a patch, we aim to find the locations of of the points. We achieve this by forming the cumulative sum image

$$t(\mathbf{q}) = \sum_{\mathbf{p}\in\Upsilon} s^{\mathbf{p}}(\mathbf{q}, l), \qquad s^{\mathbf{p}}(\mathbf{q}, l) = \begin{cases} l^1_{\mathbf{p}} \text{ if } q \in \Theta(\mathbf{p}) \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

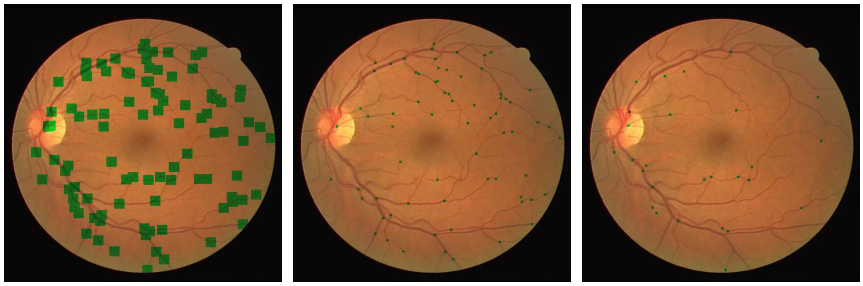
and taking the local maxima $\mathbf{r} \in \Upsilon$ as points of interest. We then aim to determine whether points are at crossings or branchings.

2.4 Junction Classification $\mathbb{C}_2$

We extract the patches $\Theta(\mathbf{r})$ and use these to train a neural network to distinguish between crossing and branchings. The second neural network was trained

with the Res18 architecture, like the first. Using a relatively small training set of patches, as from our images the majority of patches did not contain junctions, we trained our network in similar fashion to that used in the previous step. Weighted classes were introduced again to cater for the imbalance, in that images from the branching class were substantially more prominent than that of the cross class.

Depending on the patch method there were around 800-2500 patches containing a junction that were used for training. In all methods there were approximately twice as many junction patches containing branching vessels compared to patches containing vessels crossing. Training was performed until a plateau in the reduction of the loss function was reached indicating no further improvement.



(a) Identified Junctions (b) Branching Points (c) Vessel Crossings

Fig. 3: Example of second part of algorithm: classifying junctions as branchings and crossings.

3 Results

3.1 Overall

We test the ability of our algorithm using 40 images from the DRIVE database with manual segmentations [14]. This data was split to provide 30 images for training the neural networks, leaving 10 for testing. While this may seem a small number for a machine learning approach, is should be noted that the number of patches generated numbered more than 100,000 providing sufficient training data.

4 Discussion

We have produced a method that can learn to detect and classify vessel junctions using a very small dataset of 40 fundus images images that had been manually classified for junctions and their type. Using the CNN $\mathbb{C}_1$, we managed to detect the junctions to an impressive detection accuracy of over 94% due in part

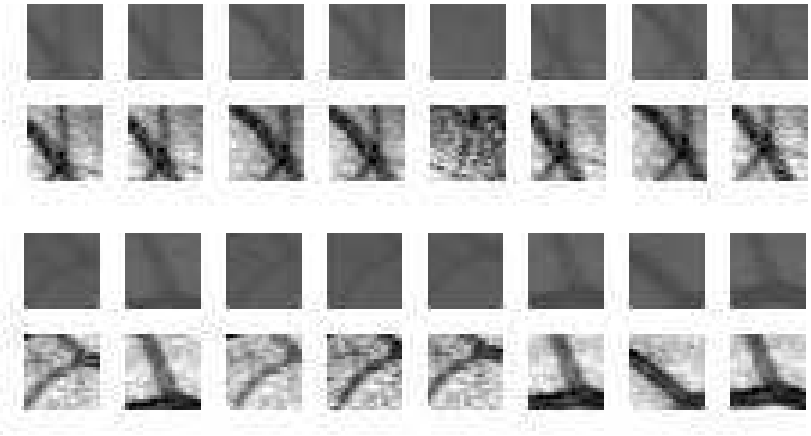


Fig. 4: Example of $\mathbb{C}_2$ input. Rows 1 and 2 (resp. 3 and 4): training patches with crossings (resp. branchings) and their enhanced counterparts for presentation. The neural networks were able to achieve good results using the patches without enhancement.

Algorithm 1 $l_p \leftarrow \mathbb{A}\left(z(\mathbf{x}), \phi(\mathbf{x}),\right)$

Skeletonisation of vessel map
Set $\varphi^0(\mathbf{x}) = \phi(\mathbf{x})$
for $\ell \leftarrow 0 : maxit$ **do**
    for $i \leftarrow 1 : 2$ **do**
        for $j \leftarrow 0 : 3$ **do**
            $\varphi^{\ell+1} \leftarrow \mathcal{F}_{i,j}\left(\varphi^{\ell}\right)$ using equation (2)
        end for
    end for
end for
Extract and classify the patches
Set $\Upsilon = \{\mathbf{p} \in \Omega \mid \varphi(\mathbf{p}) = 1\}$
for $\mathbf{p} \in \Upsilon$ **do**
    $\Theta_{\mathbf{p}} = \{\mathbf{q} \in \Omega \mid |\mathbf{p} - \mathbf{q}| \leq 10\}$
    $l^1_{\mathbf{p}} = \mathbb{C}_1\left(\Theta_{\mathbf{p}}\right)$
end for
Calculate the cumulative sum image $t(\mathbf{q})$ using (3) and determine the set of points of interest P.
$\forall\, \mathbf{p} \in P$, extract the patches $\Theta_{\mathbf{p}} = \{\mathbf{q} \in \Omega \mid |\mathbf{p} - \mathbf{q}| \leq 10\}$
Classify the extracted patches to obtain $l^2_{\mathbf{p}} = \mathbb{C}_2\left(\Theta_{\mathbf{p}}\right)$

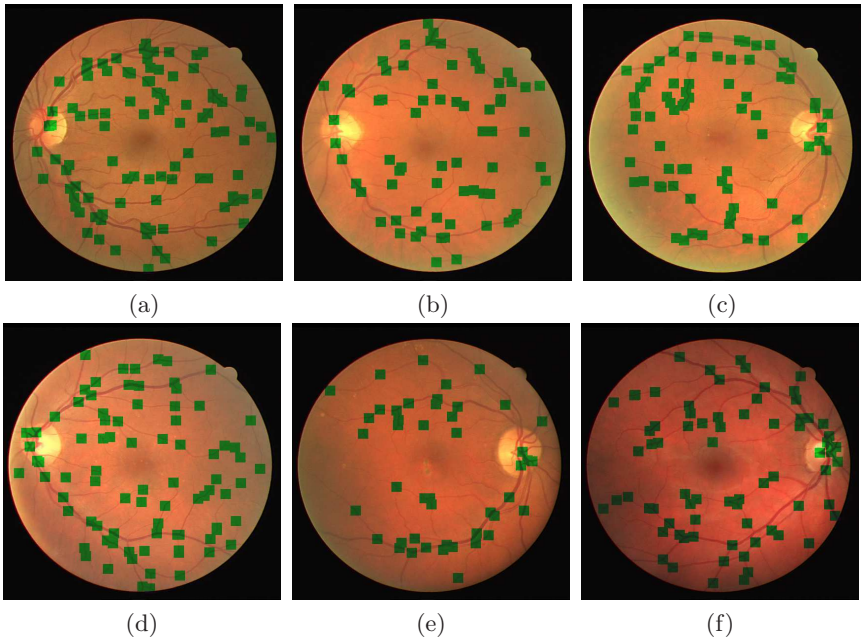


Fig. 5: Example of identifying junctions in fundus images.

to the relatively large amount of patches containing junctions. Along with the skeletonisation, our deep learning classification $\mathbb{C}_2$ for vessel type gave us an accuracy of 88%. Increasing the size of our dataset would allow better distinction in the classification of the vessel junctions. It is worth noting that junction type training was undertaken on a couple of thousand patches and tested on around 800. Through training on more images the model could be fine tuned to refine the filters and increase identification accuracy.

The current algorithm works well for images which have been manually segmented but this time-consuming task could be further extended to incorporate automatic segmentation techniques [15]. A further very useful extension would be to automatically determine whether the artery or vein is in front with arteriovenous crossings along with consideration of intra and inter-observer variability. In order to better identify and classify junctions with other nearby junctions, it would be useful to consider extending our method to a multi-scale approach.

5 Conclusion

The challenging task of detecting and classifying vessel junctions in fundus images is shown to be possible using our method. The ability to expand on this method to make the detection both quicker and more accurate than manual classification is possible. These preliminary results demonstrate that the overall framework including the deep learning approach proposed is a viable technique

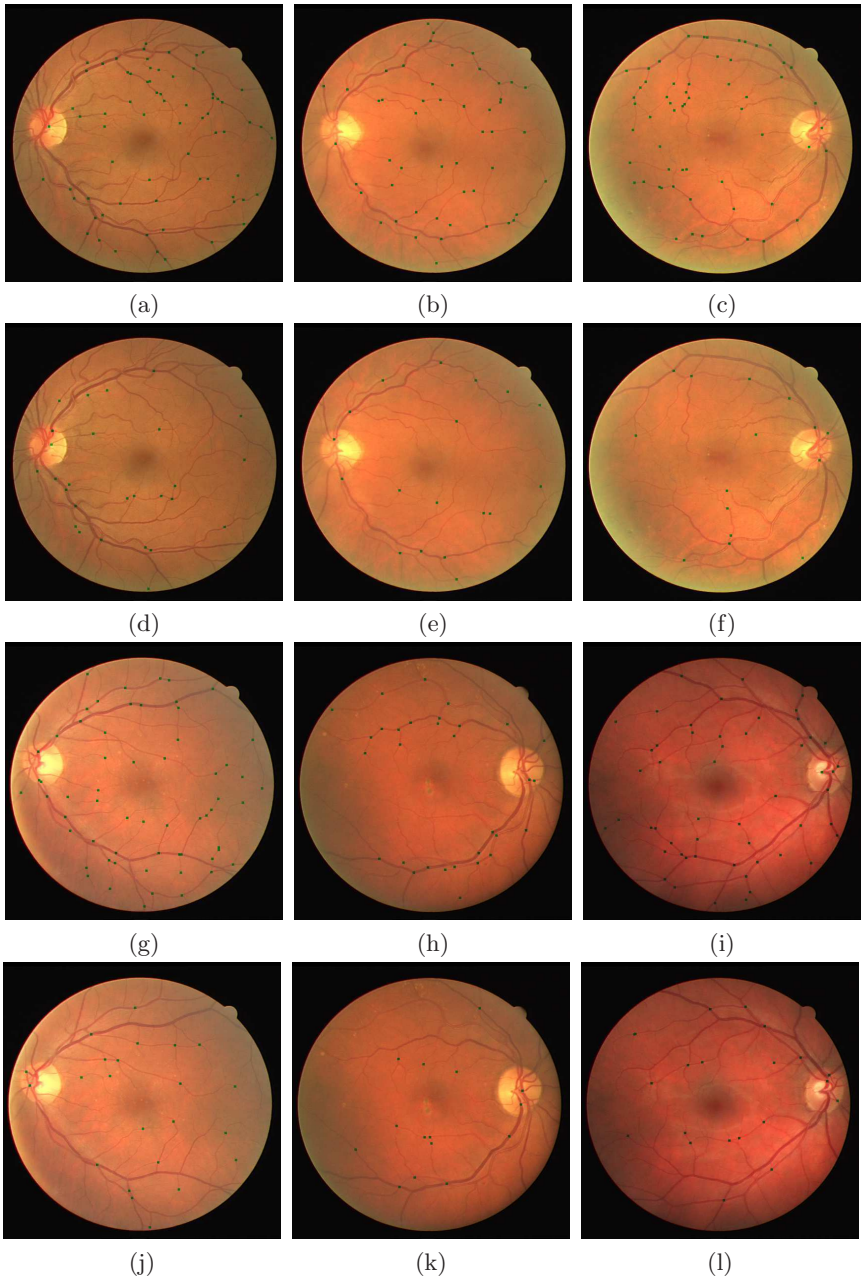


Fig. 6: Example of distinguishing between crossings and branchings in fundus images. In each column, rows one and three show branchings and rows two and four show the crossings for the respective examples.

to accurately find and identifying vessel junctions with little training data. More extensive testing of this framework could be undertaken to assess the transferability of these results to different size images and different datasets. However, there is no reason why this framework would not be directly applicable to another dataset.

Acknowledgement

H. Pratt acknowledges PhD funding from Fight for Sight charity. This project is funded in part by the National Institute for Health Researchs i4i Programme. This paper summarises independent research funded by the National Institute for Health Research (NIHR) under its i4i Programme (Grant Reference Number II-LA-0813-20005). The views expressed are those of the authors and not necessarily those of the NHS, the NIHR or the Department of Health.

References

1. Chollet, F.: keras. https://github.com/fchollet/keras (2015)
2. Chutatape, O., Zheng, L., Krishnan, S.M.: Retinal blood vessel detection and tracking by matched gaussian and kalman filters. In: Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE. vol. 6, pp. 3144–3149. IEEE (1998)
3. Cun, Y.L., Boser, B., Denker, J.S., Howard, R.E., Habbard, W., Jackel, L.D., Henderson, D.: Advances in neural information processing systems 2. pp. 396–404. Citeseer (1990)
4. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 36(4), 193–202 (1980)
5. Gonzalez, R., Wintz, P.: Digital image processing (1977)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015), http://arxiv.org/abs/1512.03385
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR abs/1412.6980 (2014), http://arxiv.org/abs/1412.6980
8. Krizhevsky, A.: Learning multiple layers of features from tiny images. https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf
9. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
10. MacGillivray, T., Trucco, E., Cameron, J., Dhillon, B., Houston, J., Van Beek, E.: Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions. The British journal of radiology 87(1040), 20130832 (2014)
11. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). pp. 807–814 (2010)
12. Sonka, M., Hlavac, V., Boyle, R.: Image processing, analysis, and machine vision. Cengage Learning (2014)

13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15(1), 1929–1958 (2014)
14. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. IEEE transactions on medical imaging 23(4), 501–509 (2004)
15. Zhao, Y., Rada, L., Chen, K., Harding, S.P., Zheng, Y.: Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images. IEEE transactions on medical imaging 34(9), 1797–1807 (2015)