MEDINFO 2017: Precision Healthcare through Informatics
A.V. Gundlapalli et al. (Eds.)

doi:10.3233/978-1-61499-830-3-559

Development of a Deep Learning Algorithm for Automatic Diagnosis of Diabetic Retinopathy

Manoj Raju, Venkatesh Pagidimarri, Ryan Barreto, Amrit Kadam, Vamsichandra Kasivajjala, Arun Aswath

Enlightiks Business Solutions Private Limited – a Practo Company, Bangalore, Karnataka, India

Abstract

This paper mainly focuses on the deep learning application in classifying the stage of diabetic retinopathy and detecting the laterality of the eye using funduscopic images. Diabetic retinopathy is a chronic, progressive, sight-threatening disease of the retinal blood vessels. Ophthalmologists diagnose diabetic retinopathy through early funduscopic screening. Normally, there is a time delay in reporting and intervention, apart from the financial cost and risk of blindness associated with it. Using a convolutional neural network based approach for automatic diagnosis of diabetic retinopathy, we trained the prediction network on the publicly available Kaggle dataset. Approximately 35,000 images were used to train the network, which observed a sensitivity of 80.28% and a specificity of 92.29% on the validation dataset of ~53,000 images. Using 8,810 images, the network was trained for detecting the laterality of the eye and observed an accuracy of 93.28% on the validation set of 8,816 images.

Keywords:

Neural Networks (Computer); Diabetic Retinopathy; Artificial Intelligence.

Introduction

Diabetic Retinopathy (DR) is one of the avoidable causes of blindness. Approximately 33% of people in the USA living with diabetes develop some stage of DR, out of which 10% evolve to a vision-threatening form of the disease [1]. Detecting DR is a time-consuming procedure that requires a trained clinician to evaluate digital funduscopic retinal images. Hence, early and quick diagnosis is a critical aspect in treatment of DR.

Computer Vision (CV) works around building Artificial Intelligence (AI) systems by interpreting information from digital images. Deep learning algorithms have the potential to produce such systems, since they do not revolve around handcrafted features. Convolutional Neural Networks (CNNs), a branch of Deep Learning, is the state of the art algorithm for image classification [7, 9]. From the 1970s and onwards, neural network based approaches were built for pattern recognition. They did not gain prominence due to their ineffectiveness to handle images of a high resolution.

CNN architecture is designed to take advantage of the 2D structure of an input image. It uses two main concepts, namely, local receptive fields and pooling. In CNN, the connections are localized to small regions called local receptive fields, and each input pixel, in turn, is not connected to all the hidden neurons. CNNs also contain pooling layers which help generate translation of invariant features and simplifies the information from the convolutional layer. Despite having many attractive features, CNNs are still expensive to apply in high-resolution colored images. Current Graphical Processing Units (GPUs), paired with the highly-optimized implementation of 2D convolution are powerful enough to train large datasets of high-resolution images [5].

The hypothesis that CNNs have the potential to improve the efficiency and speed of DR screening, and thereby help prevent visual loss and blindness from this destructive disease, is the basis of this work. The other interesting work isthe building of a second convolutional net which detects the laterality of the eye. We were unable to locate any other paper which identifies the laterality of the eye using fundus images along with the detection of the DR stage. Hence, our system is complete in a way that, when multiple images of a patient are uploaded, the system:

- Detects the laterality of the eye (Left vs Right)
- Detects the stage of DR for each eye
- Provides a diagnosis report for each eye

Few research projects have been conducted on the automatic detection of DR, but most of them were carried out with Support Vector Machines (SVMs) or fully connected neural networks. Gardner et al [2] used a neural network to detect diabetic features in fundus images. A fully connected neural network was trained on 147 diabetic and 32 normal fundus eye images. The blood vessels, exudates and hemorrhages were detected with an accuracy of 91.7%, 93.1% and 73.8% respectively.

Usher et al [3] in their work built a system using macula centered color retinal images from 1,273 patients, out of which 500, were used for training the model. Image pre-processing, identification of normal structures and candidate lesions, extraction of candidate lesion features were performed to identify if a patient was normal/abnormal. The trained system was evaluated using the remaining images and observed a sensitivity of 95.1% and a specificity of 46.3%.

Pratt et al [4] in their work managed to use a CNN based approach for automatic detection of DR on the Kaggle dataset (www.kaggle.com) comprising 80,000 fundus eye images. They observed a sensitivity of 30%, specificity of 95% and a 75% accuracy. This is claimed as one of the first papers on classifying DR into five stages.

Google (Gulshan et al [6]), in its vision of deep learning augmented healthcare, has trained a CNN indetecting referable DR using 128,175 training images which were labelled by 54 licensed ophthalmologists and trainees in a massive exercise organized for annotating the dataset. They achieved an average sensitivity of 92.72% and specificity of 95.97%. While the numbers achieved are better here, it requires massive datasets and considerable investments.

This paper provides an overview of the dataset, hardware and software configuration used, architecture of the CNN, the way the model was trained and the observed results with a note on future possibilities.

Methods

DR progresses through four stages, namely:, mild, moderate, severe nonproliferative DR (NPDR) and the advanced stages of proliferative DR (PDR). During mild NPDR, small areas of balloon-like swelling in the blood vessels of the retina called microaneurysms occur. As the disease progresses, multiple microaneurysms, hemorrhages, venous beading and cotton wool spots (CWS) occur which make the patients lose their ability to transport blood to the retina, termed as moderate NPDR. The diagnosis is severe NPDR when the abnormalities intensify and are seen on multiple quadrants of the eye and the patient has either of the following: diffuse intra-retinal hemorrhages and microaneurysms in 4 quadrants, venous beading in ≥2 quadrants, or intraretinal microvascular abnormalities (IRMA) in ≥1 quadrant. As the disease progresses, new blood vessels grow and this stage is termed PDR. The new blood vessels are fragile and are prone to bleeding and cause retinal destruction.

Dataset, Hardware and Software

EyePacs, a telemedicine platform which helps prevent vision impairment from DR (http://www.eyepacs.com/), provided the datasets used for training and model validation. Eyepacs uses telemedicine to build sustainable DR screening programs in community clinics. It links primary care providers with ophthalmologists, helping early detection of sight-threatening cases of DR. The data was accessible via the Kaggle website. The distribution of each type of image used for training from the dataset is shown in Table 1. A few of the eye images can be seen in Figure 1.

Table 1 - Training Dataset

Type	Number	Percentage
No DR (Normal)	25,810	73.48%
Mild NPDR	2,443	6.96%
Moderate NPDR	5,292	15.07%
Severe NPDR	873	2.48%
Proliferative DR	708	2.01%

Due to the challenges faced during the CNN model training, namely, long training duration and the daunting hyper-parameter tuning, only 35,126 out of 88,702 images are used. The images were of different resolution, cameras, angles, lighting, noise and artifact levels (Figure 2). The biggest challenge the algorithm had to encounter was to generalize all these cases and avoid overfitting.

The images range up to 4500 x 3500 pixels having 3 channels (R, G, B). Processing such huge images on a CNN required a high-end GPU. Two machines were used to train our model.

- NVIDIA GTX980Ti GPU containing 2,816 cores with a video memory of 6GB. The CPU contains an intel core i7 processor with 16GB memory and an Ubuntu OS.
- Amazon EC2 instance containing NVIDIA GPU with 1,536 cores, a video memory of 4GB and an Intel Xeon E5-2670 processor with 15GB memory.

The software used were, Python, NumPy and Theano (www.deeplearning.net/software/theano/), in combination with the cuDNN library. For convenience, we used Lasagne (www.lasagne.readthedocs.io/), a library built on top of Theano to build and train CNN. Preprocessing of the images wasdone using the Python Imaging Library (PIL) and the scikit-image was used for the augmentation process.

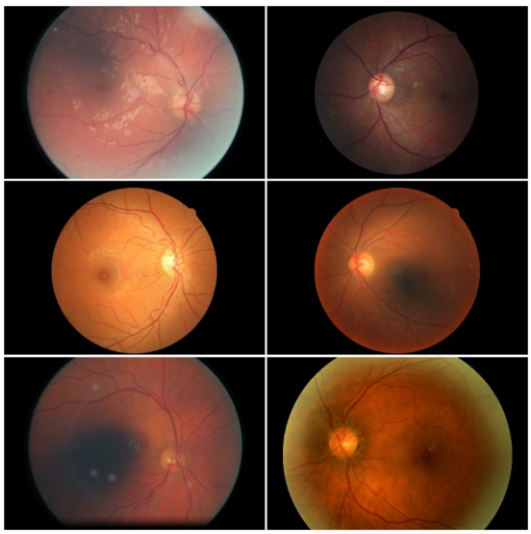


Figure 1 – Sample Fundus Eye Images

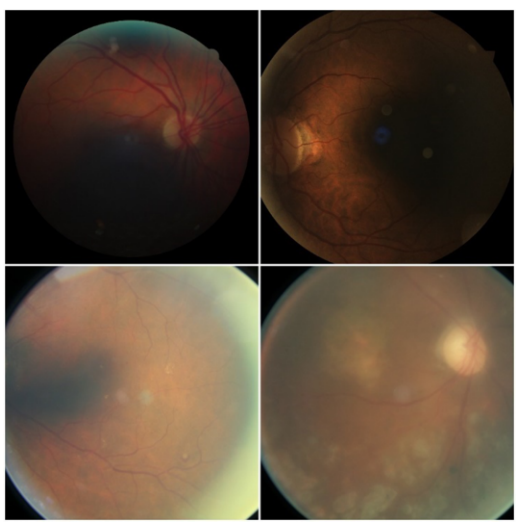


Figure 2 – Noisy Fundus Images

Diabetic Retinopathy Prediction

Data Preprocessing

The images were captured using different variants of fundus cameras, each having their unique recording method, due to which the exposure and lighting varied. In addition, images were also too large to be trained with CNN directly. The following pre-processing steps werefollowed:

- Cropping the image – The images were cropped by distinguishing the foreground from background.
- Resizing the image – The images were resized to 128x128, 256x256, 512x512, 768x768 and 1024x1024 pixels.

- Normalizing the channels – ZMUV (Zero Mean Unit Variance) was performed on each channel.

Data Augmentation

Data augmentation was performed randomly on every image over all epochs. The data augmentation steps performed were:

- Rotation – Images randomly rotated between 0 and 360 degrees
- Translation – Randomly shifted between -40 and 40 pixels
- Zoom – Randomly stretched between (1/1.2 , 1.2)
- Flip – Randomly flipping of images
- Color augmentation – Krizhevsky [5] color augmentation
- Image centering

The output size after data augmentation were 112x112, 224x224, 448x448, 672x672, 896x896 pixels for 128x128, 256x256, 512x512, 768x768 and 1024x1024 pixel images respectively.

Network Training

Deep learning is the process of training a neural network to perform a given task [6]. The deep learning algorithm (CNN) computes DR severity by analyzing the pixel intensities over each channel separately. Although the algorithm does not detect the lesions in the eye images explicitly, it learns to recognize them using the filter weights.

The training set comprised of 35,126 images for which the labels were provided by the Kaggle team. The problem was treated as a regression problem (with network output threshold at 0.5, 1.5, 2.5, 3.5) to calculate the accuracy, sensitivity and specificity. The network architecture for DR prediction is shown in Table 2. The network used in this work is inspired byOxfordnet [7].

Table 2 – Network Architecture for DR Prediction

Layer	No of Filters	Filter Size	Stride	Pad	Output Size
Input					448x448
Convolution	**32**	**4x4**	**2**		**224x224**
Convolution	32	4x4	1	2	225x225
Maxpool		**3x3**	**2**		**112x112**
Convolution	64	4x4	2		56x56
Convolution	**64**	**4x4**	**1**	**2**	**57x57**
Convolution	64	4x4	1		56x56
Maxpool		**3x3**	**2**		**27x27**
Convolution	128	4x4	1	2	28x28
Convolution	**128**	**4x4**	**1**		**27x27**
Convolution	128	4x4	1	2	28x28
Maxpool		**3x3**	**2**		**13x13**
Convolution	256	4x4	1	2	14x14
Convolution	**256**	**4x4**	**1**		**13x13**
Convolution	256	4x4	1	2	14x14
Maxpool		**3x3**	**2**		**6x6**
Convolution	512	4x4	1		5x5
RMSpool		**3x3**	**3**		**2x2**
Dropout					
Fully connected	**1024**				
Feature pool	512				
Dropout					
Fully connected	1024				
Feature pool	**512**				
Fully connected	1				

- Input - holds the raw pixel values of the image. Here, an image having the width of 448, height of 448, and three color channels R, G, B (depth = 3).
- Convolution – computes a dot product between the filter weights and a small region they are connected to in the input volume.
- Maxpool – performs a down sampling operation along the width and height. A maxpool of 3x3 will obtain the maximum of the local region of size 3x3.
- RMSpool – also performs a down sampling operation. It calculates the root mean square (RMS) value of the local region.
- Dropout – used to prevent overfitting. Nodes were randomly dropped with a probability, p= 0.5.
- Fully connected – each neuron in this layer will be connected to all the inputs from the previous layer.
- Feature pool – It is an activation function which is the max of the inputs. In our case, we use a pool size of 2.

There are about 12 million learning parameters for the network. The parameters are randomly initialized using orthogonal weight initialization. The mean squared error objective between the known label values and the network output for each image was calculated and in turn used to modify the weighting parameters during a back propagation step to reduce the training loss. This process was repeated for every image in a training set over 250 epochs with data augmentation at each step. The parameters calculated were general enough to work on new funduscopic retinal images. The classes in the dataset werehighly imbalanced (Table 1) due to which the CNN tends to bias towards the majority class. For training, a resampling of the images was done in a way that all classes were present in equal proportion. The weights of the rare classes were gradually reduced that left us with a final resampling of weights of 1, 2, 2, 2, 2 for normal, mild, moderate, severe NPDR and PDR respectively. The network training started with a batch size of 32, and a learning rate of 0.005. The learning rate was constantly decreased as the number of epochs grew and ended up with a learning rate of 0.00005. A leaky rectified linear unit (ReLU) of 0.01 was used after every convolutional and fully connected layer. An L2 regularization with a factor of 0.0005 wasapplied on every layer.

For spatial invariance, RMS pooling was used at the last pooling stage of the network. The output of the RMS pool layer was also used as features for the blending network (Table 3) which wastrained to improve the prediction results. To increase the quality of the features, feature extraction was repeated up to 50 times with different augmentations per image, and the mean and standard deviation of each feature was used as input to the blending network.

Total Features after RMS pool layer = (Number of filters * Output size after RMS pool) * 2 = (512 * 4) * 2 = 4096

Table 3 – Blending Network

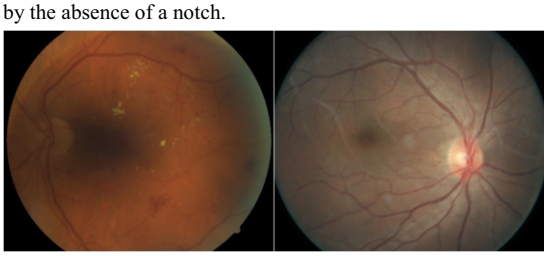
No	Layer	Output size
1	Input	4096
2	Fully connected	32
3	Feature pool	16
4	Fully connected	32
5	Maxout	16
6	Fully connected	1

All features were normalized to have a zero mean and unit variance and were used to train a simple fully connected network. A ReLU of 0.01 was introduced after each fully connected layer and an L2 regularization with a factor of 0.005 was applied to every layer. The batch size used was 128 with a mean squared error objective.

Laterality Detection

Data Preprocessing

Laterality detection prediction used the same preprocessing techniques with uncropped images. The images in the dataset were at times inverted, which needed to be identified to detect the laterality (Figure 3). The inversion of the image is detected by the absence of a notch.



(a) (b)

Figure 3 – (a) Non-inverted image (b) Inverted image

Data Augmentation

The same data augmentation process, described for DR prediction was performed, except that the images were not flipped.

Network Training

The training set comprised of 8,810 images for which the labels were provided by the Kaggle team. The laterality detection challenge was treated as a regression problem with a network output threshold at 0.5 to calculate the accuracy.

Table 4 – Laterality Detection Network

Layer	No of Filters	Filter Size	Stride	Pad	Output Size
Input					448x448
Convolution	32	7x7	1		442x442
Maxpool	32	3x3	2		221x221
Convolution	64	7x7	1		215x215
Convolution	64	7x7	1	2	213x213
Convolution	64	7x7	1		207x207
Maxpool		3x3	2		103x103
Convolution	128	7x7	2		49x49
Convolution	128	7x7	1	2	47x47
Convolution	128	7x7	1		41x41
Maxpool		3x3	2		20x20
Convolution	256	7x7	1	2	18x18
Convolution	256	7x7	1		12x12
Convolution	256	7x7	1	2	10x10
Maxpool		3x3	2		4x4
Convolution	512	7x7	1		4x4
RMSpool		3x3	3		2x2
Dropout					
Fully connected	1024				
Feature pool	512				
Dropout					
Fully connected	1024				
Feature pool	512				
Fully connected	1				

The network architecture for laterality detection is shown in Table 4. The number of learning parameters for this network are about 31 million.

The training process for the network remained similar to the DR prediction network in the weight initialization, loss function definition, regularization technique, data augmentation steps at each epoch and the number of epochs for training. The images used for training contain 50% for the left eye and 50% for the right eye and thus did not require resampling strategies.

A batch size of 32 was used to train the network. The training of the network was started with a learning rate of 0.003 (reduced as the number of epochs grew). Usage of ReLU remained the same as the DR prediction network.

RMS pooling was used for spatial invariance and feature extraction for the blending network (Table 3). The feature extraction was repeated up to 5 times with different augmentations per image and used the mean and standard deviation of each feature as input to our blending network. The total number of features used for this network remained 4096.

Application Overview

An authenticated Java application was built, which accepts patient info (patient name and patient id) and multiple images of the patient and provides a downloadable diagnosis report. The report contains the laterality of the eye, the stage of DR for each image, the follow-up advice and provides an overall diagnosis for each eye.

Results

Image characteristics used for training the network for DR prediction have been summarized in Table 1. Out of the 35,126 images used for training, 9,316 images (26.52%) had DR. The validation set consisted of 53,126 images (Table 5), of which 13,593 images (25.59%) had DR and graded between 0 and 4 (0 –no DR, 1 –mild NPDR, 2 –moderate NPDR, 3 –severe NPDR, 4 –proliferative DR). To train the network, 4x4 filters were used, which observed a specificity of 92.29% and a sensitivity of 80.28%. The confusion matrix of the classification results can be seen in Table 6.

Table 5 – Validation Dataset

Type	Number	Percentage
No DR	39533	73.79%
Mild NPDR	3762	7.02%
Moderate NPDR	7861	14.67%
Severe NPDR	1214	2.27%
Proliferative DR	1206	2.25%

Table 6 – Confusion Matrix (DR Prediction)

		Predicted				
		0	**1**	**2**	**3**	**4**
Actual	**0**	36484	2488	472	66	23
	1	1725	1271	753	13	0
	2	1016	1301	3934	1513	97
	3	18	26	232	786	152
	4	10	27	131	402	636

From Table 6, we can see that there are 10 images which have been totally rejected by the system and classified as normal. Also, 23 images which were normal have been classified as proliferative DR. The main cause for these misclassifications are the noise in the images (Fig 4). These images skew the prediction results.

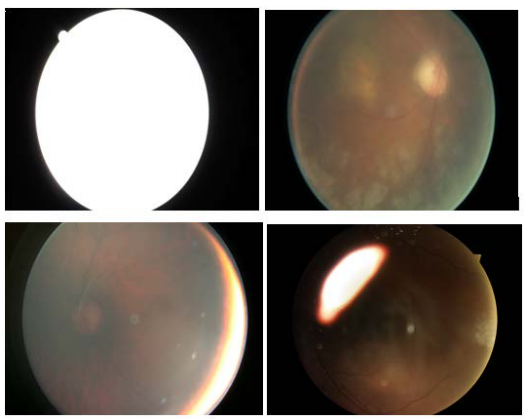


Figure 4 – Noisy Image Labeling

The confusion matrix for laterality detection is shown in Table 7. A total of 8,810 images were used to train the model which contains 4,405 right eye and 4,405 left eye images. The laterality of the eye was detected with an accuracy rate of 93.3% in a validation set containing 4,408 right and 4,408 left images.

Table 7 – Confusion Matrix (Laterality Detection)

		Predicted	
		Left	**Right**
Actual	**Left**	4156	252
	Right	340	4068

The exciting aspect of the network is its ability to classify thousands of images every minute which helps in providing a real-time diagnosis for DR.

Discussion

The network was trained using filters of size 3x3, 4x4, 5x5 and 7x7 for DR prediction and laterality detection. The best results were observed with 4x4 filters for DR prediction and 7x7 filters for laterality detection. The smaller filter size was better suited for DR prediction due to the lesions in the images which were of smaller size, such as, microaneurysms. In contrast, laterality detection requires identification of the direction of the blood vessels from the disk for which larger filter sizes suited better. Categorical cross entropy, the objective function used for classification, followed a diverging trend as the epochs progressed. Hence, regression was a better suited option for training the model. The features generated from multiple networks were ensembled but did not observe substantial change in the prediction results. Several image pre-processing strategies, such as edge enhancement, image smoothing, etc. were performed to see if learning could be improved, but neither of them provided better prediction results. Laterality detection was also observed to be inaccurate when the inverted images were mixed with non-inverted images. A dip in the laterality detection accuracy was also seen for images with no disk and macula.

The classification of Diabetic Macular Edema (DME) is an important addition required in the system. We have collaborated with ophthalmologists in getting DME annotated on the image. Deeper neural networks such as Residual Networks [8] or a variant of GoogLeNet [9] needs to be analyzed on the dataset.

Conclusion

We observed high sensitivity and specificity in the detection of DR staging, as well as laterality of the eye from funduscopic retinal images, using a CNN based algorithm. This demonstrates the potential of CNNs to automatically classify fundus images based on laterality and severity in real time. Further research is warranted on automatic classification of DME. Other parameters such as patient demographics, past history, family history etc. need to be analyzed to improve the prediction accuracy. Criteria need to be incorporated to filter out images with poor quality. Lesions on fundal images must be identified to help ophthalmologists with their diagnoses. With the incorporation of the improvement areas identified, detection of DR staging and laterality detection of the eye from funduscopic retinal images would be more robust.

References

[1] J.W. Yau, S.L. Rogers, R. Kawasaki, E.L. Lamoureux, J.W. Kowalski, T. Bek, S.J. Chen, J.M. Dekker, A. Fletcher, and J. Grauslund, Global prevalence and major risk factors of diabetic retinopathy, *Diabetes Care* **35** (2012), 556-564.

[2] G.G. Gardner, D. Keating, T.H. Williamson, and A.T. Elliott, Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool, British *Journal of Ophthalmology* **80** (1996), 940-944.

[3] D. Usher, M. Dumskyj, M. Himaga, T.H. Williamson, S. Nussey, and J. Boyce, Automated detection of diabetic retinopathy in digital retinal images: a tool for diabetic retinopathy screening, *Diabetic Medicine* **21** (2004), 84-90.

[4] H. Pratt, F. Coenen, D.M. Broadbent, S.P. Harding, and Y. Zheng, Convolutional Neural Networks for Diabetic Retinopathy, *Procedia Computer Science* **90** (2016), 200-205.

[5] A. Krizhevsky, I. Sutskever, and G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[6] V. Gulshan, L. Peng, M. Coram, M.C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams and J. Cuadros, Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs, *JAMA* **316** (2016), 2402.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[8] K. He, X. Zhang, S. Ren and Jian Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015).

[9] C. Szegedy, L. Wei, J. Yangqing, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, in: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015.

Addresses for Correspondence

Manoj Raju manoj.raju@enlightiks.com

Venkatesh Pagidimarri venkatesh.pagidimarri@enlightiks.com