Automatic Detection and Classification of Diabetic Retinopathy stages using CNN

Ratul Ghosh
Department of Electronics
and Communication Technology
Indian Institute of Information Technology
Allahabad, Uttar Pradesh 211011
Email: imm2014007@iiita.ac.in

Kuntal Ghosh
Machine Intelligence Unit
Indian Statistical Institute
Kolkata, West Bengal 700108
Email: kuntal@isical.ac.in

Sanjit Maitra
Indian Statistical Institute
North-East Centre
Tezpur, Assam 784028
Email: sanjit@isine.ac.in

***Abstract*—A Convolutional Neural Networks (CNNs) approach is proposed to automate the method of Diabetic Retinopathy(DR) screening using color fundus retinal photography as input. Our network uses CNN along with denoising to identify features like micro-aneurysms and haemorrhages on the retina. Our models were developed leveraging Theano, an open source numerical computation library for Python. We trained this network using a high-end GPU on the publicly available Kaggle dataset. On the data set of over 30,000 images our proposed model achieves around 95% accuracy for the two class classification and around 85% accuracy for the five class classification on around 3,000 validation images.**

Keywords*—*Convolutional Neural Network, Retinopathy, Fundus photography, Image Classification, Deep Learning

I. Introduction

Diabetic Retinopathy (DR) is an eye disease that damages the retina of patients with long-standing diabetes. This is an ocular complication of the eye that affects 75% of diabetic patients leading to blindness in the age group of 20-64 [1]. There are different ways to diagnose DR. The World Health Organization reports that about 347 million people in the world are affected by DR. About 366 million adults with diabetes is estimated by International Diabetes Federation. This figure is expected to rise to 552 million by 2030. Estimated occurrence of type 2 diabetes mellitus and diabetic retinopathy is quite high in India, according to the studies that have been conducted so far. Based on a survey in 2000, the top three countries with highest number of diabetes mellitus are India (31.7 million), China (20.8 million) and USA (17.7 million) [2]. Trained clinicians are required to examine the color fundus photographs of retina and detect DR. The process of identifying DR involves detection of lesions with vascular abnormalities. This is an effective way of detection but requires the service of experienced clinicians for analysis of the photographs manually, which is time-consuming. Rural areas, where the rate of diabetes is usually high, lack the expertise of well-trained clinicians and sophisticated equipment that are necessary for detection of DR. Better infrastructure with automated detection techniques are now required to tackle the growing number of individuals with diabetes. An early detection can help to avert or decrease the spread of DR which otherwise might cause blindness [3]. Previous research work for identification of the stages of DR using automated techniques includes support vector machines [4] and k-NN classifiers [5]. Most of the methods treat this as a two-class classification problem for detection of DR.

II. RELATED WORK

Numerous techniques are tested by researchers in the area for DR classification with encouraging results. Recent work for addressing blood vessel segmentation includes the application of CNN (LeNet-5 architecture) as feature extractor [6].Three heads are used in this model at different layers of the convnent which are then fed into three random forests. The final classifier achieved an accuracy of 0.97 and 0.98 on the DRIVE [7] and STARE dataset. An automatic segmentation of blood vessels in color fundus images is implemented by M.Melinscak et al [8] using deep max-pooling convnet to separate the blood vessels. The model contains a deep max-pooling convolutional neural networks to segment blood vessels. It deployed 10-layer architecture for achieving a maximum accuracy of around 0.94. It was carried around 4-convolutional and 4-max pooling layer with 2 additional fully connected layers for vessel segmentation. Automated analysis of DR using images processing techniques are introduced by Adarsh et al [9]. In this approach, extraction of retinal blood vessels, exudate, micro-aneurysms, haemorrhages and texture features takes place, followed by construction of Multiclass SVM using area of lesions and texture features. Impressive results are reported using the publicly available datasets DIARETDB0 and DIARETDB1 with accuracy of 0.96 and 0.946 respectively.

III. DATASET

A. Overview

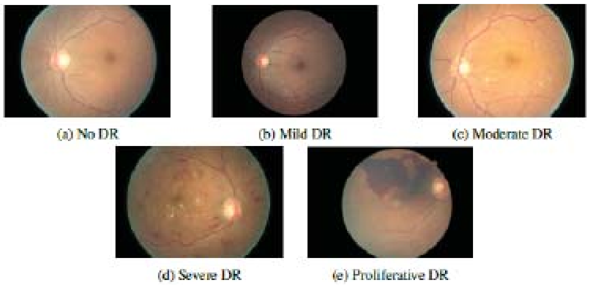
Data is collected from the dataset provide by the Kaggle coding website and maintained by EyePacs. The dataset consists of colour fundus photographs collected from various sources. The images are classified based on the severity of DR, where each image was assigned to a class by a trained clinician[1]. The figure below shows the various stages of diabetic retinopathy(DR)

B. Class Imbalance

The class labels of the dataset are highly imbalanced i.e more than 73% of the class are negative, which makes our

[1] https://www.kaggle.com/c/diabetic-retinopathy-detection/data

978-1-5090-2797-2/17/$31.00 ©2017 IEEE


(a) No DR

(b) Mild DR

(c) Moderate DR

(d) Severe DR

(e) Proliferative DR

Fig. 1. Diabetic Retinopathy(DR) stages

model difficult to train. Table I below shows the class proportion statistics, where PDR and NPDR refers to proliferative and Non-proliferative DR respectively.

Class	Number	Percentage
Negative	25810	73.5%
Mild NPDR	2443	6.90%
Moderate NPDR	5292	15.10%
Severe NPDR	873	2.50%
PDR	708	2.00%

TABLE I. CLASS IMBALANCE.

IV. FEATURES SELECTION

A. Pre-processing

The dimension of images in the dataset is 3000x2000 pixels. For convenient use of the CNN using the resources at our disposal, the images are cropped and resized to squares of 512 pixels.

B. Data Augmentation

Augmented images were created to increase the class size as there were limited number of training samples for some of the classes. Brightness of each of the images created after pre-processing were adjusted by converting the RGB image to float representation followed by converting into the original data type. This is done by adding a delta value to all the components of the image. The images are scaled appropriately and both the image and delta are converted to float prior to addition. As the addition to the image is performed in floating point representation, the delta must be in the range [0,1) whereas the pixel values are in [0,1). The original and the brightness adjusted images are then rotated by 90 and 180 degree which inherently increase the class size 6 times. This makes our model immune to different orientations and lighting conditions.

C. Normalization and Denoising

Normalization is performed to make bring the images on the same scale. The data is zero-centered followed by dividing each dimension by its corresponding standard deviation.

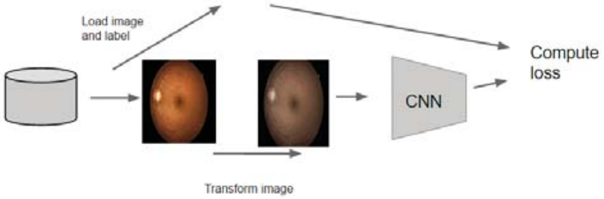


Fig. 2. Data Augmentation

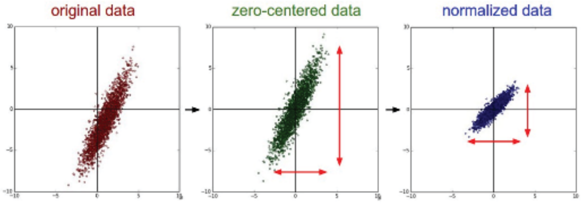


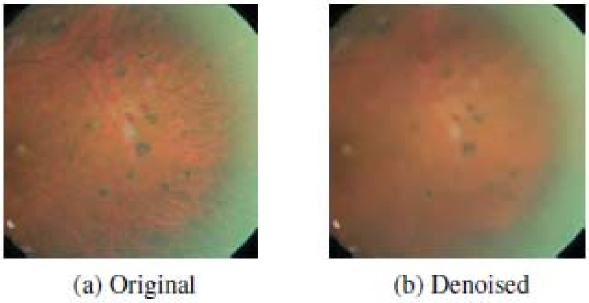
Fig. 3. Normalizing

Non-Local Means Denoising (NLMD) is implemented as the prepossessing step [10]. The denoising of an image x = $(x1;\, x2;\, x3)$ at pixel j on channel i is given as:

$$x_i(j) = \frac{\sum_{k\epsilon B(j,r)} x(j)w(j,k)}{C(j)} \tag{1}$$

$$C(j) = \sum_{k\epsilon B(j,r)} w(j,k) \tag{2}$$

where $B(j;r)$ is a neighborhood around pixel j with radius r, and the weight $w(j;\, k)$ is the square of Frobenius norm distance between color patches centered at j and k that decays under a Gaussian kernel. The figure given below shows a comparison between the original image and the corresponding denoised image in the Proliferative DR class.


(a) Original

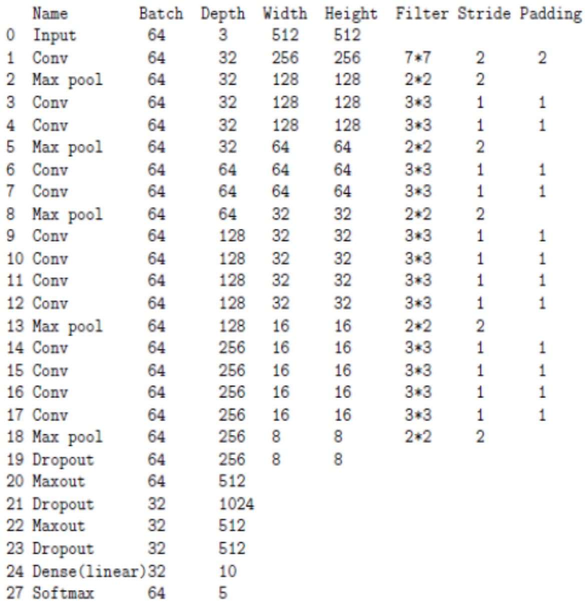
(b) Denoised

Fig. 4. Denoising

V. MODELING TECHNIQUES

A. Summary of Network Architecture

The network architectures of our conv nets is shown below



	Name	Batch	Depth	Width	Height	Filter	Stride	Padding
0	Input	64	3	512	512			
1	Conv	64	32	256	256	7*7	2	2
2	Max pool	64	32	128	128	2*2	2	
3	Conv	64	32	128	128	3*3	1	1
4	Conv	64	32	128	128	3*3	1	1
5	Max pool	64	32	64	64	2*2	2	
6	Conv	64	64	64	64	3*3	1	1
7	Conv	64	64	64	64	3*3	1	1
8	Max pool	64	64	32	32	2*2	2	
9	Conv	64	128	32	32	3*3	1	1
10	Conv	64	128	32	32	3*3	1	1
11	Conv	64	128	32	32	3*3	1	1
12	Conv	64	128	32	32	3*3	1	1
13	Max pool	64	128	16	16	2*2	2	
14	Conv	64	256	16	16	3*3	1	1
15	Conv	64	256	16	16	3*3	1	1
16	Conv	64	256	16	16	3*3	1	1
17	Conv	64	256	16	16	3*3	1	1
18	Max pool	64	256	8	8	2*2	2	
19	Dropout	64	256	8	8			
20	Maxout	64	512					
21	Dropout	32	1024					
22	Maxout	32	512					
23	Dropout	32	512					
24	Dense(linear)	32	10					
27	Softmax	64	5					

The model has six layers :

1) Convolutional Layer: This layer consists of set of filters(kernel). Each filter is convolved against the input image and features are extracted by forming a new layer. Each layer represent some significant characteristic or features of the input image.

2) Maxpooling Layer: Max pooling is a sample-based discretization process. It is used to down-sample an input representation (image, hidden-layer output matrix, etc.) and reducing it's dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. It reduces the number of parameters to learn and provides basic translation invariance to the internal representation, thus it also reduces the computational cost. Kernel size of 2x2 was used in our model for the Maxpooling process. The network is flattened to one dimension after the final convolutional block.

3) Activation Layer: This layer add nonlinearity after each layer, without which the whole network act as a simple linear transformation, which does not have so much power for complicated task such as image classification. We used leaky (0.01) rectifier units after each convolutional and fully connected layer to add nonlinearity in our model. In comparison to sigmoid and regression activation function, PReLU was used as to efficiently train the deep neural network [11].

4) Dropout Layer: This layer is used to prevent overfitting by preventing the network from relying on one node in the layer too much. Here, while updating our neural net layer, we update each node with probability p, and leave it unchanged with probability 1-p.

5) Fully Connected Layer: The high level reasoning in the neural network after numerous convolutional and max pooling layers, is performed using fully connected layers. This layer takes all the neurons from the previous pooling, convolutional or fully connected and connects to each and every neuron. They are not spatially located therefore there will be no convolutional layer after a fully connected layer.

6) Classification Layer: After the multiple layers, the final layer is a classifier layer which is stacked at the end for classifying the fundus image.

B. Weight Initialization

We have use Xavier initialization [12] as it makes sure the weights are just right while keeping the signal in a reasonable range of values through many layers.

$$W = random(fan_{in}, fan_{out})/(sqrt(fan_{in})/2) \tag{3}$$

It automatically determines the scale of initialization based on the number of input and output neurons which in this case are fan_{in} and fan_{out} respectively.

C. Loss Function

A softmax layer is used for the final prediction. Therefore, our loss function is given as:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \tag{4}$$

where f_j is the j-th element of the vector of class scores f. A proper prediction probability is ensured by the softmax in the log of the equation.

D. Regularization

A simple, effective regularization technique known as dropout is used. This approach is recently introduced by Srivastava et al [13]. During the training phase, dropout is performed by keeping a neuron active with some probability p (a hyperparameter), or setting it to zero. We have set the hyperparameter as 0.50 because it results in the maximum amount of regularization [14].

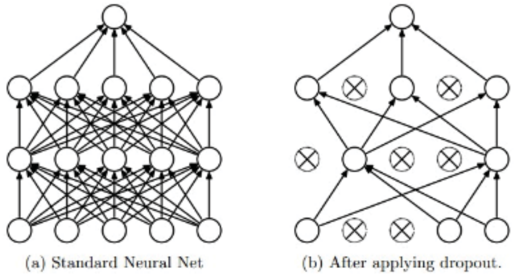


Fig. 5. Dropout

E. Parameter updates

Nesterov momentum, with fixed number of 250 epochs is used to train the networks. Nesterov momentum [15] is a slightly different version of the momentum update which has recently been gaining popularity. It enjoys stronger theoretical convergence for convex functions and in practice, it consistently works slightly better than standard momentum.

- epoch 0: 0.003
- epoch 150: 0.0003
- epoch 220: 0.00003

F. Addressing Class Imbalance

The class imbalance problem was tackled by training the algorithm on a class-balanced subset. The dataset was preprocessed to increase the class size by producing augmentation of underrepresented classes and on the other hand subsample the overrepresented classes. This creates a uniformly balanced training set.

VI. Result

The model presented in this paper were developed using Theano, an open source numerical computation library for Python, developed by the Universit de Montral[2]. ImageMagick's convert tool was used to trim off the blank spaces to the sides of the images and the training was performed using a high-end GPU.

A. Visualizing Hidden Layers

The figure below shows the activations for both eyes for the first layer.

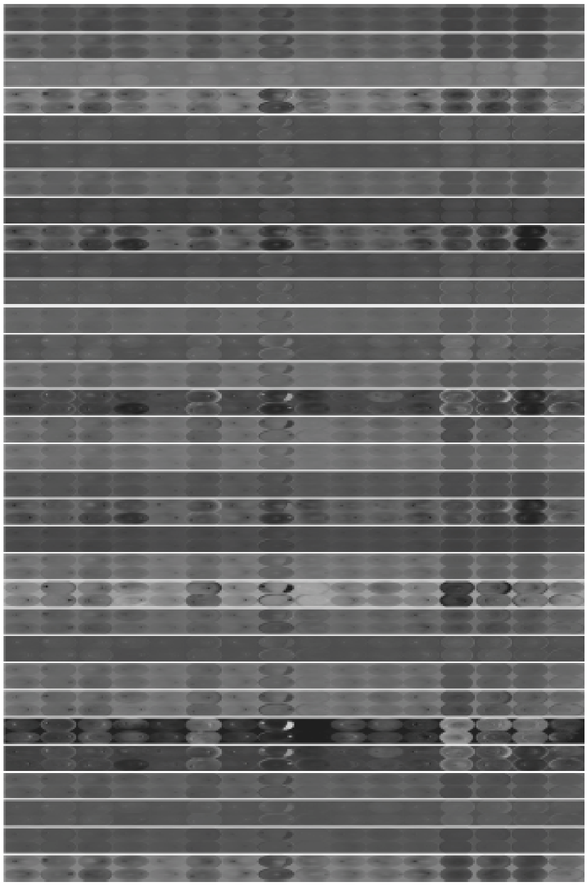


Fig. 6. First layer, 7 — 2 convolutional layer activations

True Label	0	1	2	3	4
0	**2186**	47	29	31	7
1	84	**79**	34	3	0
2	197	68	**210**	18	7
3	5	2	17	**51**	15
4	6	0	8	37	**49**

Predicted Label

TABLE II. Confusion Matrix

B. Evaluation Metrics

Four metrics are computed for performance evaluation of our models. The proportion of samples that were correctly classified gives us the accuracy of the model. As the dataset is very highly imbalanced the accuracy will not be a good measure for the model. The other metrics are recall which is the proportion of positives correctly predicted and precision which is the proportion of positive predictions that were correct. Quadratic weighted kappa is the fourth metric that is computed for performance evaluation. This metric gives an measure of the agreement between predicted and actual assignment of class. There are five possible ratings of 0,1,2,3,4. Scores by actual labels marked by clinician A and predicted labels by our model B characterizes each image by a tuple (e_a, e_b). Quadratic weighted kappa is computed using N x N histogram matrix O where each element of the matrix $O_{i,j}$ is the number of images which received i rating by A and j rating by B. This is followed by constructing an N-by-N matrix of weights, w, using the difference of scores between A and B. The weights of each of the elements of w are given as

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{5}$$

It is assumed that the rating scores are uncorrelated while computing the N-by-N histogram matrix of expected scores E. The normalization is done such that E and O have equal sum where the computation is performed as the outer product between the ratings by A and B.

Quadratic weighted kappa is calculated using the matrices using the relation:

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \tag{6}$$

C. Performance

About 10% of the images were used for validation purpose. From the confusion matrix given in Table II, we find that comparatively more images of class 1 (Mild NPDR) and class 2 (Moderate NPDR) are misclassified into class 0 (Negative class).

The final trained model achieved around 85% of accuracy for the five class classification and 95% accuracy for the two class classification(DR or no DR).

The precision and recall for each class are given in TABLE III. We observe higher values for class 0 (Negative class) whereas low values of both precision and recall for class 1 (Mild NPDR).

The final model achieved a kappa score of 0.74 which was further increased to 0.754 using ensemble.

VII. Conclusion and Future Work

A model is presented for classification of DR stages based on the severity using color fundus images. The performance

[2] http://arxiv.org/abs/1605.02688

Class Label	Precision	Recall
Class0	0.882	0.950
Class1	0.403	0.395
Class2	0.704	0.420
Class3	0.365	0.567
Class4	0.628	0.490

TABLE III. PRECISION & RECALL FOR EACH CLASS

of the model is assessed using different metrics. Considering the heterogeneity of the dataset, the performance of the proposed model is satisfactory. The accuracy of the model can be increased by using other complex denoising techniques. Incorporating experimental errors during image capture will be helpful in developing more efficient normalization methods.

ACKNOWLEDGMENT

The authors would like to thank California Healthcare Foundation and EyePACS for maintaining the dataset and Kaggle for making it available. Also many thanks to the wonderful developers and contributors of Theano for their continuous effort on these libraries.

REFERENCES

[1] Michael M. Engelgau, Linda S. Geiss, Jinan B. Saaddine, James P. Boyle, Stephanie M. Benjamin, Edward W. Gregg, Edward F. Tierney, Nilka Rios-Burrows, Ali H. Mokdad, Earl S. Ford, Giuseppina Imperatore, and K. M. Venkat Narayan. The evolving diabetes burden in the united states. *Annals of Internal Medicine*, 140(11):945–950, 2004.

[2] Cornwall J. Kaveeshwar SA. The current state of diabetes mellitus in India. *The Australasian Medical Journal*, pages 45–48, 2014.

[3] R. Williams, M. Airey, H. Baxter, J. Forrester, T. Kennedy-Martin, and A. Girach. Epidemiology of diabetic retinopathy and macular oedema: a systematic review. *Eye*, 18(10):963–983, Jul 2004.

[4] Shantala Giraddi, Jagadeesh Pujari, and Shivanand Seeri. Article: Identifying abnormalities in the retinal images using svm classifiers. *International Journal of Computer Applications*, 111(6):5–8, February 2015. Full text available.

[5] Muthu Rama Krishnan Mookiah, U. Rajendra Acharya, Chua Kuang Chua, Choo Min Lim, E.Y.K. Ng, and Augustinus Laude. Computer-aided diagnosis of diabetic retinopathy: A review. *Computers in Biology and Medicine*, 43(12):2136 – 2155, 2013.

[6] Shuangling Wang, Yilong Yin, Guibao Cao, Benzheng Wei, Yuanjie Zheng, and Gongping Yang. Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. *Neurocomputing*, 149, Part B:708 – 717, 2015.

[7] J.J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509, 2004.

[8] Martina Melinscak, Pavle Prentasic, and Sven Loncaric. Retinal vessel segmentation using deep neural networks. pages 577–582, 2015.

[9] P. Adarsh and D. Jeyakumari. Multiclass svm-based automated diagnosis of diabetic retinopathy. In *Communications and Signal Processing (ICCSP), 2013 International Conference on*, pages 206–210, April 2013.

[10] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 60–65, Washington, DC, USA, 2005. IEEE Computer Society.

[11] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *ARXIV*, 2015.

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[14] Pierre Baldi and Peter J Sadowski. Understanding dropout. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2814–2822. Curran Associates, Inc., 2013.

[15] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1139–1147, 2013.