Diagnosis of Diabetic Retinopathy with Deep Convolutional Neural Networks

Shankar Adhikari, Dasun Athukoralage
Florida International University
sadhi003@fiu.edu, dwick008@fiu.edu

Abstract

Diabetic retinopathy (DR) is most common eye diseases occur the large population as a result of diabetes. Almost 80% of diabetics will develop retinal damage after a decade with diabetes and is one of the main problems for developed countries. Retinal damage occurs when the blood vessel that supplies blood to retina has been broke down, and which is preventable through the laser treatments. For this, diabetics need to have frequent check-ups with trained doctors. But to predict DR is time-consuming and difficult to get a doctor on time. To make things easier we can apply machine learning to predict diabetes retinopathy. In this paper, we have modeled an automated detection system with realistic clinical potential to determine the presence of diabetic retinopathy on a scale of 0 to 4 using color fundus photography. We have used a feed forward Convolution Neural Network (CNN) for classification of images.

1 Introduction

Diabetic retinopathy is the leading cause of blindness in the working-age population of the developed world. It is estimated to affect over 93 million people [1]. It is estimated that 29.1 million people in the US have diabetes according to the US Center for Disease Control and Prevention. Also World Health Organization estimates that 347 million people have the disease worldwide. Diabetic Retinopathy (DR) is an eye disease associated with long-standing diabetes. Around 40% to 45% of Americans with diabetes have some stage of the disease. Progression to vision impairment can be slowed or averted if DR is detected in time, however this can be difficult as the disease often shows few symptoms until it is too late to provide effective treatment. As illustrated in Figure 1, images of patients with diabetic retinopathy can exhibit red and yellow spots which are problematic areas indicative of hemorrhages and exudates [2].

Currently, detecting DR is a time-consuming and manual process that requires a trained clinician to examine and evaluate digital color fundus photographs of the retina. By the time human readers submit their reviews, often a day or two later, the delayed results lead to lost follow up, miscommunication, and delayed treatment.

2 Related Work

The quantification of diabetic retinopathy and detection of features such as exudates and blood vessels on fundus images were studied [3][4]. Image processing techniques such as image preprocessing then segmentation after that 2D matched filters and image thresholding techniques were widely used for more clarify the image.

During the recent years, there have been many studies on detection of diabetic retinopathy using several machine learning techniques. SVM for the detection of diabetic retinopathy stages using color fundus images has been studied in [5]. In [6], authors have proposed an automatic method

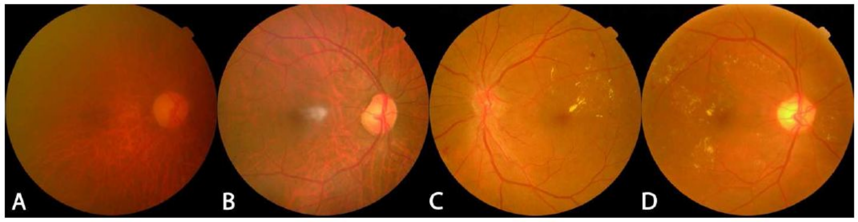

Figure 1: Examples of color fundus images. The first two images show healthy eyes while the last two images contain exudates, manifestations of retinopathy.

to detect exudates from lowcontrast digital images of retinopathy patients with non-dilated pupils using a Fuzzy CMeans (FCM) clustering. Exudates are also among the preliminary signs of diabetic retinopathy, a major cause of vision loss in diabetic patients. For this purpose, naive Bayes and SVM classifiers have been used and the performance has been compared with the nearest neighbor classifier as a baseline method [7]. Their results show that the naive Bayes and SVM classifiers perform substantially better than the nearest neighbor classifier.

Application of multilayer feed forward neural networks (NN) in the same context, has attracted the research community extensively. Accurate feature extractions and accurate grading of DR lesions are not required in NN. Its robustness as the network can also classify DR effectively in noisy environments is an another strengths of this technique. A multilayer feed forward neural network consisted of an input layer (257 nodes), one hidden layer (11 nodes), and an output layer (7 nodes) has been proposed in [8] for the classification of DR. The neural network has been trained using error back propagation technique.

3 Data and Resources

3.1 Data Description

Data has been taken from California Health Care Foundation [9]. Data contains images of right and left eyes retinal view under a variety of conditions. The image has a different scale of DR ranges between 0 to 4 based on the earlier rating done by the clinician. Where 0 is assigned for no DR and 4 is for proliferative DR. Our job is to build a model to predict right score based on these scale ranges.

Different models and types of cameras have been used for obtaining the images in the dataset which can affect the visual appearance of left vs. right. Some images are shown as one would see the retina anatomically (macula on the left, optic nerve on the right for the right eye). Others are shown as one would see through a microscope condensing lens (i.e. inverted, as one sees in a typical live eye exam). The one way to tell whether an image is inverted is if the macula (the small dark central area) is slightly higher than the midline through the optic nerve. Also, if there is no notch on the side of the image (square, triangle, or circle) then it's inverted.

3.2 Tensorflow

Tensorflow is an open source library for machine learning developed by Google Brain team [10] and released on November 9, 2015. Tensorflow was the architect in such a way that, the numerical computation would follow a graph where a node of the graph gives the mathematical operation and edge of the graph represents the multidimensional data array [11]. The tensorflow was already been used by google research team in machine learning application such as speech recognition, smart messaging and more [12].

In this project, we choose tensorflow because of better performance in the similar type of problems [13]. Moreover, it is easy to implement and have plenty of libraries mostly define for convolution neural network.

4 Features Selection/ Extraction

The images for this analysis are too large to train with convolutional neural network and suitable for our hardware system. We therefore crop the images to take away all the background, and resizes the images to the squares of 300 pixel. We don't apply any other preprocessing.

5 Modeling Techniques and Training

5.1 Convolution Neural Network

The convolution neural network is similar to the ordinary neural network where a network is made by the neurons that have learnable weights and biases. These neurons receive inputs(for our case image input), do the dot product and produce output with the non-linearity functions(such as Relu, tanh, etc). The figure below shows the simple architect of the convolutional neural network.

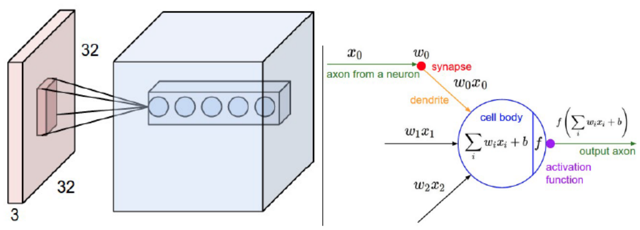

Figure 2: An example of convolutional neural network [14]. **Left**: An example of input volume (32 $\times$ 32 $\times$ 3), where neuron locally connected but to the full depth. **Right**: Shows the dot product of neurons with inputs followed by non-linearity activation function to produce output.

5.2 The Architecture

The architecture of our convolution neural network is summarized in table 1. It contains two convolutional layers. The output of the last convolutional layer is fed to the fully connected layer which produces a distribution to the 5 class labels.

The first convolutional layer filters the 300$\times$ 300$\times$ 300 input image with 16 kernels of size 5$\times$ 5 $\times$3 with a stride of 2(this is the distance between the receptive field centers of neighboring neurons in a kernel map). Which basically do the dot product between the weights(randomly define filter values from the normal distribution) and the input region where the filters are connected. The second convolutional layer takes as input, the output of the first convolutional layer and filters it with 16 kernels of size 5$\times$ 5 $\times$16. After each convolution, RELU apply an elementwise activation function, such as $max(0, x)$ thresholding at zero.

The last layer that connects the intermediate convolution with output class labels is the fully connected layer. This layer computes the class score, resulting in the volume of size 1$\times$ 1$\times$ 5, where each of the 5 numbers corresponds to a class score for our 5 categories in the data. As the name implies(fully connected), each neuron in this layer connects with all the numbers in the previous volume.

Total	Size	filter	stride	padding	No. of Weights
1. Input:	300 $\times$ 300 $\times$ 3				0
2. Conv:		5	2	2	150 $\times$ 150 $\times$ 16
3. Conv:		5	2	2	75 $\times$ 75 $\times$ 16
4. FC :					75 $\times$ 75 $\times$ 16
5. FC:					(75 $\times$ 75 $\times$ 16)$\times$ 5

Table 1: Convolutional network architectures

6 Results

The provided dataset has large size with 46000 training image of the 5 different class labels. Because of limited resources, we trained our model for only 872 image data. The training accuracy of our model is 0.93. We test our model on small validation set with 100 images and check the performance of the model. Our validation accuracy was 0.70.

In the summarized table above, we can see a large number of weights need to save in the memory in order to update through back propagation. So, if we want to use all the dataset to train the model, we need a machine with the large memory. This work will elaborate by addressing this problem to achieve the better result.

7 Conclusion Future Works

Relatively we applied a basic architecture of convolution neural network with two convolutional layers. To improve our model we can architect multilayer convolution neural network with several pooling layers and dropout layer in between those convolution layers. In addition with that, there are several process we can add to improve our model such as preprocessing, data augmentation etc.

As previously explained there are lots of ways we can improve our model. This project will elaborate in the future by seeking new resources to address the drawbacks in our project.

8 Acknowledgments

We want to thank the California Healthcare Foundation and EyePACS, along with Kaggle for the availability of data set. We also thank our professor from the machine learning class for her motivation and support.

References

[1] Kaggle competition, 2015.

[2] M. Gandhi and R. Dhanasekaran. "Diagnosis of Diabetic Retinopathy Using Morphological Process and SVM Classifier". *In Proc. IEEE International conference on Communication and Signal Processing,*, pages 873–877, 2013.

[3] A.K. Gaur A. Kumar and M.Srivastava. "A Segment based Technique for detecting Exudate from Retinal Fundus image". *Procedia Technology*, 6:1–9, 2012.

[4] C.J. Kumari and R. Maruthi. "Detection of Hard Exudates in Color Fundus Images of the Human Retina.". *Procedia Technology*, 30:297–302, 2012.

[5] R. Priya and P. Aruna. "Automated Classification System for Early Detection of Diabetic Retinopathy in Fundus Images". *International Journal of Applied Engineering Research*, 1(3), 2010.

[6] A. Sopharak and S. Barman. "Automatic exudate detection from non-dilated diabetic retinopathy retinal images using fuzzy c-means clustering". *Sensors*, 9(3):2148–2161, 2009.

[7] B. Uyyanonvara S. Barmanc T. Williamson K. Nweb A. Sopharak, M.N. Dailey and Y.A. Moeb. "Machine learning approach to automatic exudate detection in retinal images from diabetic patients". *Journal of Modern Optics*, 57(4):124–135, 2010.

[8] A. Roychoudhry A. G. Shannon J. Flack H. T. Nguyen, M. Butler and P. Mitchell. "Classification of diabetic retinopathy using neural networks". *In Proc. of the 18th Annual International Conference of the IEEE*, 4:1548–1549, 1996.

[9] http://www.chcf.org.

[10] https://www.tensorflow.org.

[11] Ashish Agarwal Martin Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.

[12] https://www.technologyreview.com/s/544356/heres-what-developers-are-doing-with-googles-ai-brain/.

[13] http://www.popsci.com/google-ai.

[14] http://cs231n.github.io/convolutional-networks/.