MONASH University

School Of Information Technology

FIT - 3162

Computer Science Project

Semester 2

by

Nabil Ahmed (ID: 25364170)

Kevin Setiawan (ID:25895710)

Final Report

Classification of Diabetic Retinopathy using Retinal Images (Convolutional Neural Network with MATLAB + Fusion Techniques)

Introduction

Diabetic Retinopathy(DR) is considered one of the top six most dominant causes of blindness in the world. By early detection, the patients chances of developing DR can be reduced significantly. Which is why classification of DR in real time is an imperative procedure that might save lot of lives. Not only that, the accuracy of this classification is very important as well because depending on the result of the classifier the patient will receive treatment. DR classification detection requires a trained clinician to classify it correctly. For doing this classification there are a several number of features that needs to be checked. For this reason, even a trained clinician may take a long time for classifying DR into a particular category. These kind of issues can be tackled with the help of a Neural Network. Nowadays, computers can be trained to learn from all kinds of images. Neural Networks can achieve a quicker, consistent and accurate results if it is trained well. Neural Network is a machine learning technique which teaches a neural network to learn from input images by using a lot of layers. The layers job is to extract features and learn from them. Another reason for using a deep neural network is that it eliminates human error significantly when classifying DR. In this project, we classify the DR into three different classes which are no DR, mild DR and severe DR. By classifying them into three different classes, it becomes easier for clinicians and doctors to treat the patient with the correct treatment.

This project was done by using Convolutional Neural networks (CNN) which is a part of deep learning that has shown remarkable result in the field of image classification. By using GPU and the CNN architecture, it is possible to detect even complicated patterns from complex images. CNN also showed promising results when training the neural network that consisted of a lot of classes.

In this project, we computed the CNN architecture by using MATLAB programming language(MathWorks). The reason we used CNN in this project is CNN is already well known in the field of image classification and many successful researches show amazing result using CNN. The dataset was provided by our supervisor which were already classified into three different classes. In total, there were 654 retinal images used in this project. The dataset is processed and augmented before it went into the network. The methodology used in this project is later explained in the methodology section.

Background

Diabetes occur when the pancreas cannot process enough insulin in the patient's body (Verma, Deep et al. 2011). If the patient does not take proper care, the blood sugar level in their body will increase and thus possibilities of developing DR becomes higher. Severe development of DR can damage the retina which can lead to loss of vision and eventually into blindness (Doshi, Shenoy et al. 2016).

World Health Organization(WHO) created a survey about DR, there are more than 347 million people in this world suffering from diabetic retinopathy. The predict that the number can reach upto 500 million people by the end of 2030. Experts and highly trained clinicians detect lesion with vascular abnormalities for detecting the level of DR. They also require lot of equipments and supplies to correctly classify DR. DR is a serious problem in overpopulated and underdeveloped countries such as India and China. In underdeveloped countries, the main problem is there is a lack of expert clinicians and equipments. Studying and becoming a proficient DR expert requires time and practice as well. So, there is a lacking. (Pratt, Coenen et al. 2016). So, real time detection for classification of DR is vital for the well being of people suffering from it.

Diabetic retinopathy stages can be divided into two classes which are early DR and advanced DR. Early DR is known as Non-Proliferative Diabetic Retinopathy (NPDR). Advanced DR is known as Proliferative Diabetic Retinopathy (PDR). NPDR is when the patient can cured if given proper treatment is given and if the patient controls their diabetes level and makes lifestyle changes. In PDR, the patient has little hope of getting cured from DR, if the condition becomes worse the patient can permanently become blind.

An Intro to Convolutional Neural Network

Convolutional Neural Network (CNN) are made up of a lot of layers. Information is passed between the layers. These layers can be divided into 3 parts. First part takes the input images. Second part extracts the features from the images and the third and last part learns from the extracted features of the second part. For our problem of detecting level of DR the output layers returns the strength of the networks prediction of each possible class.

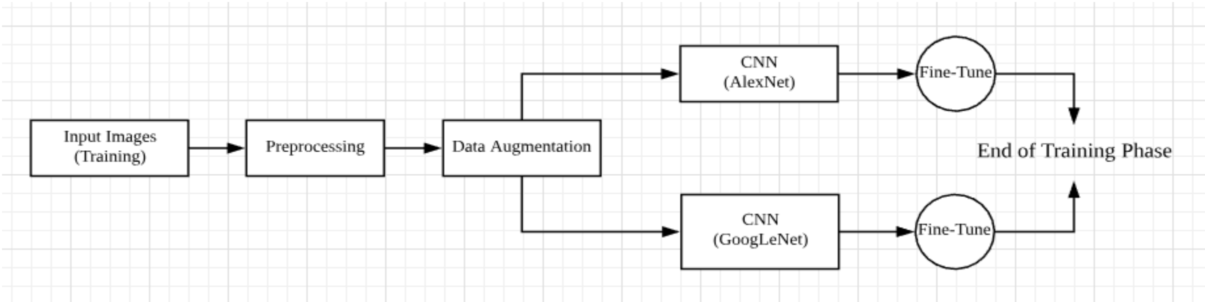
For example, the second part contains layer like convolution, pooling, relu etc that performs certain type of extractions and passes that information onto the next layer. The third part contains layers like fully connected, softmax, output etc which takes information produced by the layers in second part and learns from them. The layers themselves comes with many parameters which are called weights. The weights determine how the layers behave when data is passed through them. The values of these weights are determined by training the network on known data. In this way, the behaviour of the network is learned from the data. The networks can have the same architecture but behave differently if they were trained using different data set. However, if two networks have different architecture but are trained using the same dataset the networks will learn in a different way. In a manner, the two network are like two human beings. Both individuals learns same things in their own way.

Methodology

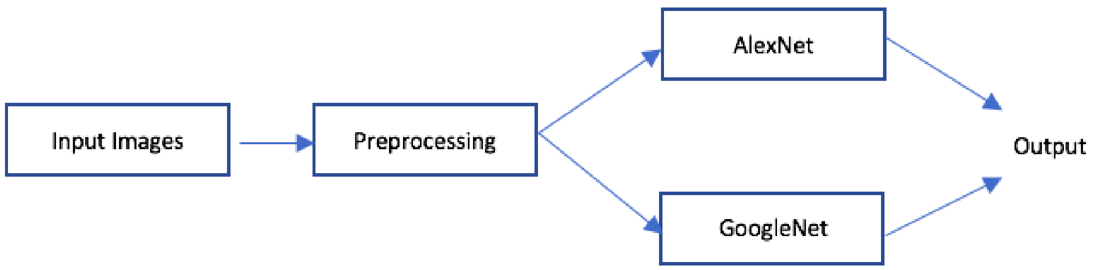
Proposed Methodology

In our proposed methodology, we divided the algorithm into two parts, Training and Testing. The two diagrams below shows our proposed approach which was given in our proposal. For the finer details of our actual proposal please refer to the Project Proposal Report. The method implemented is actually quite different then what was proposed since at the time of writing our proposal we did not have much ideas about the implementation of the CNN architecture and how the programming side of the architecture worked. We also lacked knowledge about what kind of resources we might have in order to compute the algorithms. Another thing to note is that we are quite new to Matlab and its components. But, the tutorials in the Mathworks society really helped us in developing this project. The biggest hurdle we faced during the progression of this project was understanding how the different layers works and how to pass the parameters for the training option. Frankly speaking, most of the components of how the architecture is working is still unclear as understanding those elements require deep level knowledge in mathematics, statistics, data science, computer science etc and most importantly time. With the time that was given to us we did our best in understanding and tweaking the parameters to get the validation results.

Training



Testing



<u>Figure: The figures above shows the training and testing phase of our proposed methodology</u>

Algorithm Implemented

Firstly, a note about Matlab. Out entire algorithm and code deliverables are implemented using the matlab programming language. From the experience gained we can distinctly say that Matlab is a good programming language for doing scientific work and complex numerical calculations, the libraries and tutorials provided by Mathworks are really useful without which completing this project would not have been possible.

We two CNN architecture used for implementing this project are the Alexnet and GoogleNet.

Alexnet has 25 layers in it and GoogleNet has 144.

Pre-processing

For both the architectures, the first layer is manually selected. The first layer gives the size of the input image that it accepts as input.

For Google Net the input image size is 224 * 224 * 3.

For Alex Net the input image size is 227 * 227 * 3.

We resize the images to the given dimensions of the first input layers from the respective architectures and then pass them to the network for training. We do this for both the training and testing set. For all the images we take the average RGB values. We then take the mean of the Red, Green and Blue channel [13]. For each pixel we subtract this average value from the given value of each channel. This is the pre-processing step.

The above pre-processing was selected after doing several different types of pre-processing of the retinal images. Histogram equalization, min-max normalization [Doshi, Shenoy et al. 2016] was proposed but they did not yield satisfactory results. The above method was chosen because it was used by the winner of a competition of classifying DR held by Kaggle [13].

Data augmentation

While training the images are randomly reflected in the x and y direction. This is crucial in learning since the model is learning from the image from different orientations and angles [Sun, Wan et al. 2017].

Algorithm Continued

We split the dataset into 70% for training and 30% for validation. There is a graph shown in the evaluation report that shows how the training is done and shows its progress in stages. For most of the experiments performed our model mostly overfits. This mean our model was learning things in a very specific and precise manner than validation was possible. To reduce this gap we fine-tuned our model.

Fine-Tuning

We added drop-out layers into our model. What it basically does is it tells the network to learn at a specific rate and to not learn very deeply. With this combination and increasing the epoch size it was seen that obtaining a better validation accuracy was possible. Another thing we had to do was, change the fully connected layer to 3, so that our model classifies the three stages of DR only which was given in the input files and not classify from other classes that was previously built into the architecture.

Validation

The validation is done while training. The validation dataset contains thirty percent of the data from the original training data. The validation accuracy determines how well the architecture is learning.

Testing

For testing we just take one image or a few images and pass it through the architecture. It gives us the confidence value and class is which the image belongs to.

For example, we pass an image through the architecture and it returns
No DR: 0.1 Mild DR: 0.3 Severe DR: 0.6

This means that the model is 10% confidant that the given image belongs to the No DR class. The model is 30% confidant that the given image belongs to the Mild DR class. The model is 60% confidant that the given image belongs to the Severe DR class.

Fusion Technique

The fusion technique used in this project is the late fusion. We used two architectures. The alexnet and GoogleNet. Both the architectures are trained using the same dataset. They learn differently. We save both the models and perform testing in them.

For example, testing one image using both architecture we get the following output:

Image1 = No DR: 0.2 Mild DR: 0.6 Severe DR: 0.2 (using alexnet)

Image1 = No DR: 0.3 Mild DR: 0.3 Severe DR: 0.4 (using GoogleNet)

As we can see, using alexnet on Image1 we get Mild DR as the output and Severe DR as output using GoogleNet. Meaning, both the architecture learned in a different way.

Now, this is what we did for the fusion technique:

If the output of both the classifier belongs to the same class, we return the output as that class.

If the output of the classifier belongs to different class we return the output that has the highest confidence.

So, for the example above, the output is Mild DR.

Project Management and Process

As per the curriculum, this project is divided into two semester. Last semester which was 2018, Semester One was the time we started working on this project. At first, all of the students from computer science department were given some topics from Dr. Reza, our supervisor. During the first semester, our primary focus was to understand the topic that we have chosen by reading journals and research papers. The important part was understanding the CNN architectures. There are many different types of CNN architecture that already exists and are available for use. As per our research during the semester, we agreed that Alexnet and Googlenet was suitable for our project and should give us respectable validation accuracy. In the end of semester one, we came out with our project proposal.

In addition, in semester two we were more focused on the implementation, training the architecture, preprocessing the images, testing the data, testing the images etc. There were lots of issues that we faced during this period of time like understanding Matlab language, time management, doing assignment of other units. In the project proposal, we already created a WBS and timeline for both the semesters. We tried our best to manage and to follow the schedule that we made before as a reference.

In this project, there were some requirement that needed to be complete. The most important aspect was the code implementation. Followed by the hardware used such as the need for high performance computing and memory. Next, the dataset that consisted of fundus images required for the input and the heart of the project. Next, the software used to write the code implementation (Matlab). In this project, all code is written in MATLAB programming language only. This project also required two CNN architecture to classify retinal images to three different classes. CNN architecture that are used are alexnet and googlenet.

Task Name	Duration	Start	Finish
Semester 1	35 days	Mon 09/4/18	Fri 25/5/18
Article Review	15 days	Mon 09/4/18	Fri 27/4/18
Literature Review	10 days	Mon 30/4/18	Fri 11/5/18
Project Proposal and Presentation	10 days	Mon 14/5/18	Fri 25/5/18
Final Exam and Semester Break	40 days	Mon 28/5/18	Fri 20/7/18
Research on CNN and Fusion Technique	40 days	Mon 28/5/18	Fri 20/7/18
Get Familiar with Matlab	40 days	Mon 28/5/18	Fri 20/7/18
Semester 2	60 days	Mon 23/7/18	Fri 12/10/18
Review and Discussion	5 days	Mon 23/7/18	Fri 27/7/18
Code Implementation	40 days	Mon 30/7/18	Fri 21/9/18
Pre Processing Images	10 days	Mon 30/7/18	Fri 10/8/18
CNN	20 days	Mon 13/8/18	Fri 07/9/18
Fusion Technique	10 days	Mon 10/9/18	Fri 21/9/18
Documentation	20 days	Mon 24/9/18	Fri 19/10/18

Schedule Proposed in the project proposal

Project Risk	Possibility	Effect	Strategy	Plan
Project is not finish	Average	Extreme	Have a good time management	-
Code implementation is not working well and not efficient	Average	High	Have a lot of testing and training data for get a better result	Debug the code until efficient and operate well
The result is not accurate	Low	Low	Use a reliable CNN approach	Try to use another CNN approaches

Risk table (taken from project proposal)

There are some issue and risk that we faced during the implementation, training and testing data. As per the table taken from the project proposal, we list three major risk. First, the project is partially implemented. There were a lot of room for improvement but due to time constraint it was not possible to do all the possible modifications. We can better handle this in the future by having good time management technique. Second risk was code implemented often had bugs in it. This required a lot of trial and errors, testing and fine chunk of our time. Especially during the first few week of implementation, the project did not seem to progress due to the errors in our code bundle. We managed to debug our code by a lot research, dedication and gaining for knowledge in the area of neural network and machine learning. Last risk that we listed is that the results were not accurate. Both network that we used during this project did not showing promising results like we expected during our research about CNN in semester one. We tried to modify our code and try other network but still got similar outcomes.

Next, we also realized we had a lot of shortcoming during the project. The most significant one is not having a good time management. Especially when there was another assignment due. It was really hard for us to split our time as the pile of work kept on increasing as the weeks went by. Next limitation we had was we realized that we were not really an expert in the field of DR and CNN network, so we had to gather a lot of outside knowledge. During our research in semester one, both of us assumed to have a lot of knowledge about CNNs. But while implementing, it was harder for us to even understand our mistakes and correct them promptly. In addition, we were not really familiar with MATLAB programming. Sometimes, the only reason it took so long time to write our code was because we were writing in a new language.

Outcomes

After two semesters, both of us gained a lot of knowledge about diabetic retinopathy. Just by checking the retinal images of patient, the clinician can detect the stage of DR. DR can be reduced if the patient takes care of themselves in the earlier stages of DR. From all the research from these two semesters, we can now compare the advantages and limitations of neural networks. By implementing all the code in MATLAB, we became proficient in a new programming language. Most important thing is we managed to train our dataset that consisted of retinal images that were already categorized into three different classes which is no DR, mild DR and severe DR. From more than 600 images used, using all the training and testing data, we achieved around 45% accuracy for both network.

As already stated above, by training more than 600 images into three different classes, we achieved 47.45% for alexnet with 6 epochs, 48.98% for alexnet with 20 epochs and 40.31% for google net with 6 epochs, 35.71% for google net with 20 epochs . After that, some data augmentation was applied in the network to increase the amount of data. Then after that, we use 6 and 20 epochs for training images. From research we believe that alexnet and googlenet has the best performance for doing convolutional neural network on retinal and fundus image. In the result, we only get less than 50% accuracy. Meaning that this implementation is not really accurate and cannot be used in the real life. However, the idea is there and if implemented correctly will be really helpful for the people suffering from it. Chances of getting error is high if the clinician does not classify DR correctly, it will have a major impact on the patients life.
One of the reason of why our model cannot be used in real life is because of the accuracy. If the model predicts incorrectly the patient will get different suggestion of cure depending on the level of DR. Another problem might be when the patient is suffering from severe DR but the architecture detects it as no DR or mild DR, the patient can suffer from blindness because of this miscalculation. That’s why the importance of its accuracy is impeccable.

For improving this project, there are some evaluation that can be applied. First, using bigger dataset for training the architecture. It will really help the architecture to learn better. Also, we tried using other preprocessing techniques before it gets into training phase. Our preprocessing technique may not be the best method to see the extrudes, blood-vessel, etc. clearly but it is one that works perfectly.

After implementation we achieve two results, we only get around 48% for alexnet and 40% for google net with around 600 images. But in the research that we did in semester one, both alexnet and googlenet achieve more than 95% accuracy with more than 80,000 images. The result from our experiments are still unreliable to be used in the real life. The method used in the researches had error rates less than 5%. Meaning that the chance of having a mistake when classifying images was really low. But in our implementation, the error rate is more than 50% means that it is not reliable for clinicians to use our program. To use a program that classify DR into their different classes the error rate should be less than 1%. Because once the output is wrong and classifies incorrectly, it may change someone life forever.

In our coding implementation, there were a lot of built-in functions used which are hard to understand and not readable. It was impossible to use our laptops for implementing this project because our laptops did not have enough power to run the simulations. Sometimes our laptops even crashed while training the architecture. Thanks to our supervisor, Dr. Fermi for sharing the high performing computers with us to train the architecture. We shared the computer with another group and alternatively used the VM for computing the project. Sometime the other group took long time to train their model so that also delayed our implementation a bit. Another limitation was the dataset. Our dataset was very small. It was not possible for us to train and get a high enough accuracy from the given data.

Conclusion

Automatically detecting the level of DR in a patient's eye is a crucial task, especially in countries where the rate of diabetes is high among the population and in places where the required detecting equipment or specialist are not readily available. There are many existing models out there that detects the level of DR from retinal images of the patient with outstanding accuracy. For most of the research paper given in the reference section it can be seen that a lot of proficient minds worked on them onto building neural network that can efficiently detect and classify the level of DR. Some researchers used transfer learning and compared them with CNN's that were built from scratch. By working on this project and from the experience we gained over the past ten months we can distinctly say that we have grown much more appreciation for machine learning and AI. The power that this tool has is immense and if utilized properly a lot of the problems in today's world can be simplified.

References

1. *Acharya, U., Lim, C., Ng, E., Chee, C., Tamura, T.. Computer-based detection of diabetes retinopathy stages using digital fundus images.*
   P I Mech Eng H 2009;223(5):545–553.

2. Andonová, M., et al. (2017). Diabetic retinopathy screening based on CNN. 2017 International Symposium ELMAR.
3. Tennakoon, R.B., Mahapatra, D., Roy, P., Sedai, S., & Garnavi, R. (2017). Image Quality Classification for DR Screening Using Convolutional Neural Networks.
4. García, G., et al. (2017). Detection of Diabetic Retinopathy Based on a Convolutional Neural Network Using Retinal Fundus Images. Artificial Neural Networks and Machine Learning – ICANN 2017, Cham, Springer International Publishing.
5. Doshi, D., et al. (2016). Diabetic retinopathy detection using deep convolutional neural networks. 2016 International Conference on Computing, Analytics and Security Trends (CAST).
6. Ghosh, R., et al. (2017). Automatic detection and classification of diabetic retinopathy stages using CNN. 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN).
7. Li, X., et al. (2017). Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI).
8. Pratt, H., et al. (2016). "Convolutional Neural Networks for Diabetic Retinopathy." Procedia Computer Science **90**: 200-205.
9. Sun, J., et al. (2017). Retinal Image Quality Classification Using Fine-Tuned CNN. Fetal, Infant and Ophthalmic Medical Image Analysis, Cham, Springer International Publishing.
10. Tajbakhsh, N., et al. (2016). "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?" IEEE Transactions on Medical Imaging **35**(5): 1299-1312.
11. Verma, K., et al. (2011). Detection and classification of diabetic retinopathy using retinal images. 2011 Annual IEEE India Conference.
12. Zhou, L., et al. (2018). "Deep multiple instance learning for automatic detection of diabetic retinopathy in retinal images." IET Image Processing **12**(4): 563-571.
13. http://blog.kaggle.com/2015/09/09/diabetic-retinopathy-winners-interview-1st-place-ben-graham/