Using Convolutional Neural Networks for Diabetic Retinopathy Classification

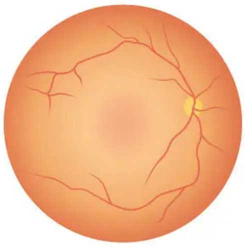
Samir Kerkar - March 12, 2020

Abstract

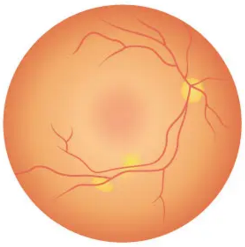
Diabetic retinopathy (DR) is a common complication of diabetes that can lead to vision loss and blindness. Early detection and diagnosis of DR are critical for the effective treatment and prevention of vision loss. Convolutional neural networks (CNNs) have been increasingly used for DR image classification, allowing for automated and accurate analysis of retinal images. This paper reviews recent advances in the use of CNNs for DR image classification, and proposes a new model trained on different retina image databases. The model includes various architectures, datasets, and performance metrics.

1. Background

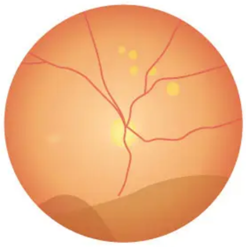
Image classification is a fundamental task in computer vision that involves assigning a label or category to an input image. Convolutional neural networks (CNNs) have emerged as a powerful technique for solving this problem. CNNs are a type of deep neural network that is specifically designed to work with image data. They use a series of convolutional and pooling layers to learn hierarchical representations of the input images, which can then be used for classification. The main advantage of using a computer is that this process can be scaled to large volumes of data to make accurate classifications quickly. This is particularly important in the field of medicine, where there is a plethora of digital images collected every day in the form of x-rays, cat scans, cellular images, etc. In almost all cases these digital images require a medical professional scientist or doctor to classify diagnose or come to any sort of conclusion from these images. However, the interpretation of retinal images can be challenging and time-consuming, requiring skilled ophthalmologists and specialized equipment. To address these challenges, convolutional neural networks (CNNs) have been increasingly used for DR image classification, allowing for automated and accurate analysis of retinal images. Furthermore, manual classification is slow and susceptible to human error and bias.



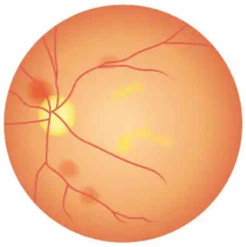
Mild NPDR



Mod. NPDR



PDR



Severe NDPR

1.1. Non-Proliferate Diabetic Retinopathy

A person diagnosed with non-proliferate diabetic retinopathy (NPDR) is said to have tiny blood vessels leak that make the retina swell. Macular edema is a condition where the macula swells are the main cause of loss of vision in diabetic patients. The other condition that can affect vision is macular ischemia—a condition that causes the blood vessels in the retina to close off. This stops blood from reaching the macula leading to the formation of tiny particles called exudates. NPDR can further be classified into three types based on the severity of symptoms.
1. Mild NPDR—a few microaneurysms.
2. Moderate NPDR—Presence of cotton-wool spots and hemorrhages.
3. Severe NPDR—Presence of intra-retinal hemorrhaging in four quadrants of the eye–two with venous beading or one with intra-retinal microvascular abnormality.

1.2. Proliferative Diabetic Retinopathy

When left untreated, diabetic retinopathy progresses to a more serious stage, called proliferative diabetic retinopathy (PDR). In this type, new blood vessels start growing in the retina at an abnormal pace. Pressure is built in the eyeball when the new blood vessels grow to interfere with the fluid flow causing the retina to uncouple itself from behind the eye. Blood also leaks into the jelly-like substance present in the center of the eye—vitreous. As a result of the above factors, the optic nerve is damaged, the nerve that passes through the blind spot carrying inverted images from the eye to the brain resulting in vision loss. For the purpose of speeding up the process and precise predictions, the CNN approach is developed. CNN has already been applied for effective predictions in various fields like healthcare [1,2] and intelligent automation [3]. By understanding its strength, CNN is applied in this work to diagnose diabetic retinopathy from eye images and classify them accurately based on the severity. This system shall diagnose diabetic retinopathy automatically without user intervention. The proposed models are analyzed on the publicly available Kaggle dataset [4] to demonstrate their impressive results.

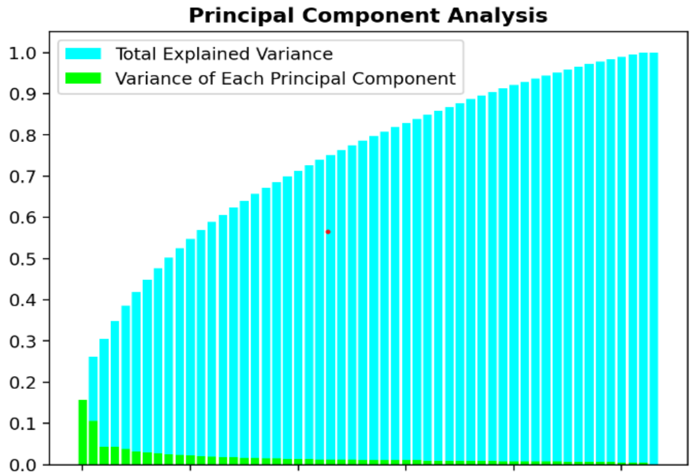
1.3 Classification Algorithms

Recently, researchers have turned to machine learning algorithms to assist in classification tasks. There are several conventional algorithms that can be used for image classification: K-Nearest-Neighbor (KNN), Support Vector Machines (SVM), and Convolutional Neural Networks (CNN) [2]. The KNN algorithm predicts the label of an object by looking at the closest “neighboring” examples. It chooses a number of K examples to look at, then chooses the example that occurs the most among those K neighbors. KNN is a simple algorithm, so it is memory efficient and easy to execute but will perform very slowly for more complex problems [2]. SVMs work by creating a boundary line that divides the data into separate classes. Based on this principle, SVMs work very well in discriminating between one class and another, also known as binary (2- class) classification. However, SVMs are limited in that they don’t support multiclass classification. Additionally, SVMs are not suitable for large datasets and will perform poorly if there are more features required for analysis [2]. A CNN is an artificial neural network, which works like a funnel that takes in input data, searches for different feature patterns, repeats this process until it learns the patterns, then outputs a prediction label. CNNs in particular are built specifically for image processing and analysis and can work with large datasets without compromising speed and accuracy [3]. In this paper, we will use a CNN. Various CNN architectures have been used for DR image classification, including VGG, ResNet, Inception, DenseNet, and EfficientNet. These architectures differ in the number of layers, depth, and complexity of the network.

2. Data Collection and Processing

The data required for this project were from various databases including EyePACS, Messidor, and IDRiD. EyePACS is a large-scale dataset that contains 88,702 retinal images from 4,614 patients, with labels indicating the presence or absence of DR. Messidor is a dataset that contains 1,200 retinal images from 1,200 patients, with labels indicating the presence or absence of DR. IDRiD is a dataset that contains 81 color fundus images and corresponding labels for DR and diabetic macular edema. In order to successfully analyze each image we need to resize the images to 424 x 424 pixels in size. Then we use a standard normalization technique for each image. Having all of the images normalized and the same pixel size speeds up data processing significantly and produces more accurate results. Before fitting our data to a model, we can perform dimensionality reduction techniques. In this case, we will use Principal Component Analysis (PCA) which is a widely used technique in data analysis and pattern recognition that is used to identify and extract the most important features from a high-dimensional dataset. In many real-world applications, such as image processing, financial analysis, and genetics, datasets may contain a large number of variables or features that are highly correlated, making it difficult to identify patterns or trends in the data.

PCA allows the dimensionality of a dataset to be reduced while preserving the essential information contained in the data. The goal of PCA is to transform the original high-dimensional data into a lower-dimensional space, where the new variables, called principal components, are uncorrelated and capture the maximum amount of variance in the data. Below we can see we vastly reduced the dimensions of our data while preserving over 90% of the variance.



After gathering our data, we split the dataset into two parts. In every machine learning model, there must be at least two datasets: a training set and a testing set, which we can call df_train for training and df_test for testing (df stands for data frame and is a commonly used variable name in programming). df_train is used to train the model with the correct labels, or answers. After the model is trained, df_test is used to test the model. Typically in a machine learning model, the dataset is split between 70% training and 30% testing. Because our dataset contains around 90,000 images, we split the dataset into 63,000 images for df_train and 27,000 images for df_test.

2.1 Building the Model

Next, we move on to building our CNN model using Keras and TensorFlow. A typical CNN architecture consists of an input layer, Convolutional layers, Pooling layers, Fully Connected (FC) layers, and an output layer [3]. The input layer takes in the original image. The Convolutional layer extracts features from the input image. The pooling layer gives the model variance and controls overfitting, which means it prevents the model from learning only one specific type of image. The Fully Connected layer is used to form the final output of the data. Lastly, the output layer outputs the final prediction that the model has made. We built the architecture of our model based on VGG's model. We will improve on VGG's relatively simple architecture with 16 or 19 layers.

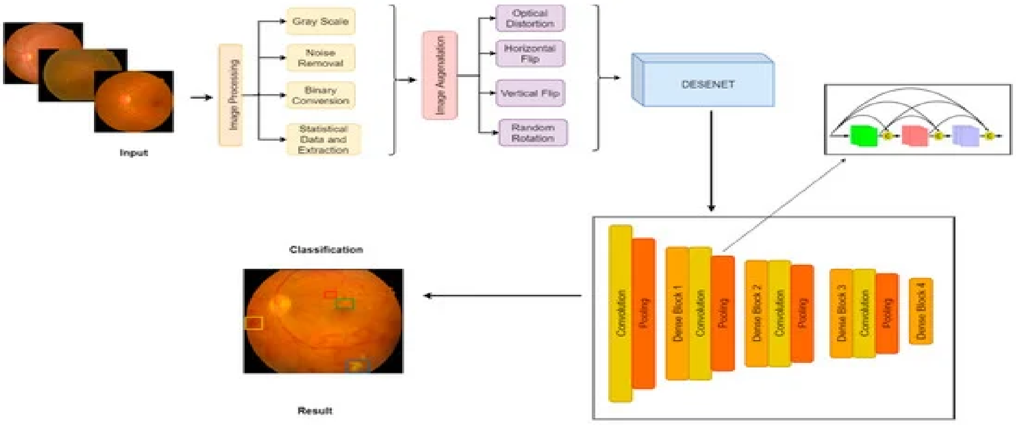


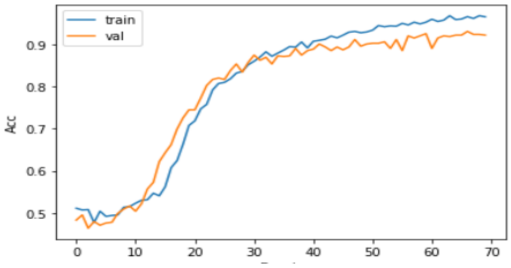
FIG 1. Proposed Diabetic Retinopathy Classification Framework

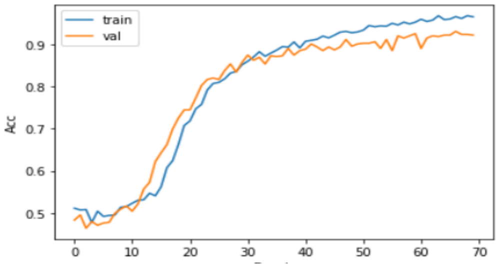
2.2 Training and Testing

After building the model, we ran the model to train and test it. We used epochs as a measurement of time. We trained it on 40 epochs, which took around 170 minutes to complete. An epoch is one full learning cycle of the machine learning algorithm. Therefore, the more epochs we set the model to train on, the more time it has to learn the correct results. However, if we used more than 40 epochs, we would risk overtraining our model. Overtraining the model leads to overfitting, which reduces the accuracy of the model.

3. Model Results

Once we finished training and testing, we want to visualize the model accuracy and loss curves by plotting them with respect to the epoch. The accuracy curve plots the accuracy percent rate as a function of time (in epochs). We aimed to produce a log curve, which has a low accuracy rate in the beginning, then exponentially increases to a high accuracy rate at the end. Around 40 epochs are when the training curve starts to diverge, in which we will perform early stopping to prevent overfitting. Adapted from [5]. The loss curve indicates how bad the model's prediction is on a single example. It plots the loss number as a function of time (in epochs). A loss number of zero signifies that the model's prediction was perfect, so we wanted to achieve a low loss number close to zero. We aimed to produce a negative log curve, which has a high loss number in the beginning, then exponentially decreases to a plateau towards zero. Around 40 epochs are when the training curve starts to diverge, in which we will perform early stopping to prevent overfitting.





4. Conclusion

In summary, convolutional neural networks have emerged as a powerful tool for DR image classification. In this paper, we successfully created a CNN that classified the stages of DR with 95% accuracy. While our CNN has shown promising results in automating DR image classification, there are still several challenges that need to be addressed, including the limited availability of high-quality and diverse datasets, the interpretability of the models, and the potential biases in the training data. Addressing these challenges will be critical for realizing the full potential of CNNs in DR image classification and improving the accuracy and efficiency of medical diagnosis and treatment.

5. References

1. Chazhoor, A.; Sarobin, V.R. Intelligent automation of invoice parsing using computer vision techniques. *Multimedia. Tools Appl.* **2020**, *81*, 29383–29403. [**Google Scholar**] [**CrossRef**]
2. Sanket, S.; Vergin Raja Sarobin, M.; Jani Anbarasi, L.; Thakor, J.; Singh, U. Narayanan, S. Detection of a novel coronavirus from chest X-rays using deep convolutional neural networks. *Multimedia. Tools Appl.* **2020**, *81*, 22263–22288. [**Google Scholar**] [**CrossRef**] [**PubMed**]
3. Kumar, S.L. Predictive Analytics of COVID-19 Pandemic: Statistical Modelling Perspective. *Walailak J. Sci. Technol. (WJST)* **2019**, *18*, 15583. [**Google Scholar**] [**CrossRef**]
4. Available online: **https://www.kaggle.com/competitions/aptos2019-blindness-detection/data?select=test_images** (accessed on 13 June 2019).
5. Jiang, H.; Yang, K.; Gao, M.; Zhang, D.; Ma, H.; Qian, W. An Interpretable Ensemble Deep Learning Model for Diabetic Retinopathy Disease Classification. In Proceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019; pp. 2045–2048. [**Google Scholar**]
6. Roy, A.; Dutta, D.; Bhattacharya, P.; Choudhury, S. Filter and fuzzy c mean based feature extraction and classification of diabetic retinopathy using support vector machines. In Proceedings of the International Conference on Communication and Signal Processing (ICCSP), Tamil Nadu, India, 6–8 April 2017; pp. 1844–1848. [**Google Scholar**]
7. Qian, Z.; Wu, C.; Chen, H.; Chen, M. Diabetic Retinopathy Grading Using Attention-based Convolution Neural Network. In Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2019; pp. 2652–2655. [**Google Scholar**]
8. AbdelMaksoud, E.; Barakat, S.; Elmogy, M. Diabetic Retinopathy Grading Based on a Hybrid Deep Learning Model. In Proceedings of the International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 26–27 October 2018; pp. 1–6. [**Google Scholar**]
9. Pires, R.; Jelinek, H.F.; Wainer, J.; Goldenstein, S.; Valle, E.; Rocha, A. Assessing the Need for Referral in Automatic Diabetic Retinopathy Detection. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 3391–3398. [**Google Scholar**] [**CrossRef**] [**PubMed**]