International Journal on Emerging Technologies 11(3): 1082-1087(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Diabetic Retinopathy Detection and Classification using Pre-trained Convolutional Neural Networks

Sanskruti Patel
Assistant Professor, Faculty of Computer Science and Applications,
Charotar University of Science and Technology (CHARUSAT), Changa-388421 (Gujarat), India.

(Corresponding author: Sanskruti Patel)
(Received 24 April 2020, Revised 27 May 2020, Accepted 28 May 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Diabetic Retinopathy (DR) is a situation that affects the eye and it is a complication that occurred due to diabetes. It can be detected using retinal fundus photographs. Traditional image processing methods have some limitations to analyse data and hidden patterns from the images. Deep neural networks have been successfully applied in many computer vision tasks in the recent years. In this paper, we have applied pre-trained Convolutional Neural Network (CNN) models VGG16 and MobileNetV1 on the publically available retinal fundus image dataset on kaggle. We have applied pre-processing and augmentation techniques that helps to improve the accuracy in feature extraction. The experiment is carried out using 35,126 retinal funds image dataset available from kaggle with five class labels. Transfer learning with fine tune on pre-trained VGG16 and MobileNetV1 obtained the training accuracy of 89.42% and 89.84% respectively. The test accuracy achieved by VGG16 is 89.51% and MobileNetV1 is 89.77%.

Keywords: pre-trained convolutional neural networks, transfer learning, diabetic retinopathy.

I. INTRODUCTION

Diabetes, also known as Diabetes mellitus is a metabolic disease is a condition of a body when there is an availability of high blood sugar in the blood. It is a chronic disease and can damage many body organs especially eyes. Diabetic retinopathy is a situation that affects the eye and it is caused by uncontrolled blood sugar in the body. It is a complication that occurred due to diabetes and it damaged the eye that leads to irreversible loss in vision. Early detection of diabetic retinopathy is essentially helpful for prognosis. Diabetic retinopathy is commonly detected through retinal fundus image [1]. There is a lack of clinical facilities available in many countries that generate barriers in timely diagnosis.

The recent advancements in Information and Communication Technology (ICT) greatly impact in improving health care facilities. Computer vision integrated with artificial intelligence is able to recognize and understand images and represent the data that helps to execute actions. It has been shown promising results in the field of healthcare. Deep learning, a subfield of machine learning, is made up with layered architecture that helps to analyse the data. With the increased complexity in the medical imaging data, deep learning architecture make computers to process, analyze, recognize and classify medical images.

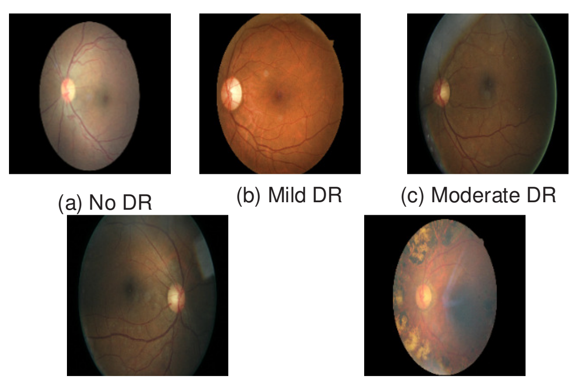
Convolutional neural network (CNN) is a kind of deep learning architecture made up with one or more convolutional layers and mainly used for digital image processing, segmentation and classification work. In last few years, it has successfully applied to process and analyse medical image data including retinal fundus image [2]. This paper represents the application of CNN models to detect and classify the diabetic retinopathy from retinal funds images. A retinal fundas image is taken as an input to the CNN model that extracts the features automatically and generates an output.

The experiment is conducted on the dataset used from Kaggle data repository that contains 35,126 retinal fundas images that falls into five classes. Two CNN models, VGG16 and MobileNet V1 applied with transfer learning and fine tuning and performance analysis is carried out using various parameters.

The paper organises as follows. Section 2 contains the background and related work. Section 3 explains the experimental methods. Section 4 explains the dataset and results obtained during experiment. Section 5 contains discussion and conclusion.

II. BACKGROUND AND RELATED WORK

Diabetic retinopathy (DR) normally has no early warning signs and in some cases no symptoms. There are many types of medical examinations can take place to diagnose the disease. One of the examination is called fundus photography, where the experts can examine blood vessels, nerve tissues condition and other necessary things [3].



(a) No DR (b) Mild DR (c) Moderate DR

(d) Severe DR (e) Proliferative DR

Fig. 1. Retinal Images for Diabetic Retinopathy with Disease Grades

Majority of the research has been considered five classes of diabetic retinopathy as mentioned in following figure 1. The Fig. 1 shows the stages of diabetic retinopathy that encompasses through No DR, Mild DR, Moderate DR, Severe DR and Proliferative DR [4]. The experts examine the above images for diagnosis for DR with its grades. However, Computer enabled diagnosis of diabetic retinopathy has been discovered as a potential approach by many researchers since last few years. It reduces the barriers of having accessibility of experts and even may help to remove inconsistent diagnosis.

Many researchers worked in the area of implementing computer enabled diagnosis of diabetic retinopathy. Traditional digital image processing techniques involves steps like image pre-processing, segmentation, feature extraction and classification. Tanthuwapathom, R. *et al.* [5] used image processing including image segmentation techniques for detecting diabetic retinopathy. They have experimented on 60 retina images and achieved accuracy up to 85%. Ratanapakorn, Tanapat *et al.,* [6] developed software for diagnosing diabetic retinopathy from fundus photograph using techniques of digital image processing. They have experimented on 400 funds images and achieved results as 98%, 67% and 96.25% for sensitivity, specificity and accuracy respectively. R.J. Winder [7] *et al*., carried out an extensive literature survey on availability of different algorithms at various stages of image processing. K. Bhatia *et al*., [8] represented their work on applying machine learning classifying algorithms for detecting diabetic retinopathy. They have extracted different features first and then applied various classification algorithms including SVM, decision tress etc. Al-hazaimeh, Obaida *et al.,* [9] represented an effective method based on image processing that detecting diabetic retinopathy that includes various stages from pre-processing to classification. Z. Gao *et al.,* [10] presented their work for automatic diagnosis of DR using deep neural networks. They were able to active the accuracy of 88.72%. Harry Pratt *et al.,* [11] presented an approach for detecting DR using CNN. They have achieved sensitivity of 95% and an accuracy of 75%. Lam, C. *et al*., [12] used CNN on color fundus image for recognizing the staging in DR. they have used transfer learning on pre-trained models. Gulshan V *et al.,* [13] created an algorithm using deep learning techniques for recognition of diabetic retinopathy and macular edema. There is a wavelet-based fusion method is proposed for MRI and PET images by Kumar, A. *et al.,* [14]. Also, a comprehensive review of detecting edge in an image is presented by Patel, A. *et al.,* [15].

In recent years, many researchers have been applied deep learning architectures for detecting DR. Many of them considered CNN as an effective architecture as it provides feature extraction without manual intervention.

III. METHODS

Convolutional neural network (CNN) is a deep learning architecture that made up with convolutional layers, pooling layers, fully connected layers and activation functions. The architecture of CNN is basically inspired by working of the visual cortex of a human brain and it designed to mimic the It forms on the basis of regular neural network architecture that consists of input layers, one or more hidden layers and output layers. CNN image classification takes an image as input, process it using hidden layers and classify it as an output [16]. CNN uses convolution layers that extract features from an image automatically [17]. For that, it uses small squares that preserve relationship between pixels. It also uses set of filters or commonly known as kernels that are used to produce the output image. It generates feature map also known as an activation map. To reduce the amount of dimensionality, pooling is implemented by pooling layers. It performs on each feature map generated by convolutional layer and always added after convolutional layer. The commonly used pooling operations are max, sum, average etc. Activation function is also known as transfer function and it can be linear and non-linear. Sigmoid, Tanh, ReLU and leaky ReLU are the most commonly used activation functions. There are many networks are evolved in past few years that are based on convolutional neural network. A very first architecture introduces was LeNet and subsequently AlexNet, ZF Net, GoogLeNet, VGGNet, ResNets, DenseNet and MobileNet are introduced.

VGG16: Among of all other CNN based models, VGG16 is a CNN architecture that is proposed by K. Simonyan and A. Zisserman in 2014 [18]. It was developed from Visual Geometry Group Lab of Oxford University andVGG16 was a winner of 2014 ILSVRC competition. It achieved 92.7% top-5 test accuracy and it has been implemented on ImageNet dataset that contains 14 million images belonging to 1000 classes. The following Fig. 2 shows the architecture of VGG16 with input, set of convolution and pooling layers and an output [19].

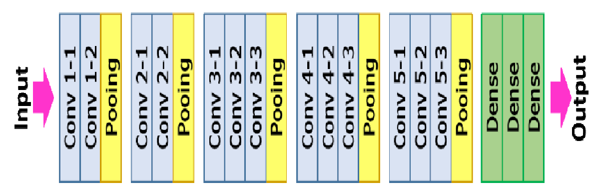


Fig. 2. Architecture of VGG16.

The input to the VGG16 is a fixed size 224 × 224 image in RGB, having dimensions set as (224, 224, 3). It then passes through two convolutional layers where 3X3 filters are used. It followed by pooling layer that uses max pooling with stride of 2 × 2. It again passed through a set of two convolutional layers followed by one pooling layer. It then passes through the sequence of convolutional layers and pooling layers. At the end, there are three fully connected (FC) layers available that follow a stack of convolutional layers.

These layers are having different architecture than its predecessor layers. The first two dense layers consists 4096 channels each. The last dense layer is a softmax layer having 1000 channels as it predicts from 1000 classes available in the ImageNet dataset.

MobileNet V1: A special kind of deep learning network called MobileNet V1 is introduced for applications related to mobile and embedded visions[20]. The base of this network is depth wise separable neural networks. A set of two hyper parameters are used for building MobileNet which is very small and having low latency.

There are many real time applications are available that used MobileNet as depicted in Fig. 3 [21].

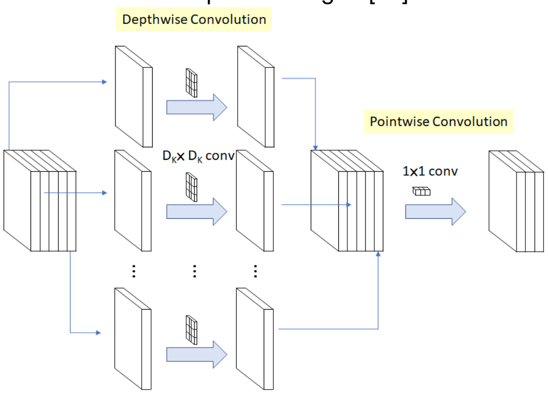


Fig 3. An Architecture of Depth-wise Separable Convolution.

A single filter is applied to each input channel in MobileNet. The outputs are combined by applying a 1×1 convolution using the pointwise convolution. In a single step, a standard convolution performs filters and combines inputs into a new set of output. For that, there are two layers used that are split from depthwise separable convolution. Therefore, there is separate layer available to filter and separate layer available to combine. This has made a drastic impact on reduction of size and computation. MobileNet has 28 layers including depthwise and pointwise convolutions as separate layers. It does essentially helpful to develop smaller and faster networks. MobileNet has emerged an effective base network for object detection model.

Transfer Learning and Fine Tuning: CNN is successfully applied in many computer vision tasks and other applications. When a CNN model is applied to a dataset of a real time application, it is required to train the CNN model. It is very time consuming and computationally expensive to train the whole model from scratch. Rather, a common approach is to use a pre-trained CNN model that has been already trained on a very large dataset. Here, the weights are reused in one or more layers from a pre-trained model by either keeping all weights fixed or fine tune the model [22]. It means transfer the knowledge gained from one model to another model. It makes the learning process of a new model fast and more accurate.

In first approach, a pre-trained CNN model is used as feature extractor for a new dataset. Therefore, only last fully connected layer is replaced according to the number of classes available in new dataset. Here, except the last layer, all other layers are freeze during training. Freezing layers are not updated during training process on a new dataset [23].

Another approach is to fine-tune the CNN model where along with replacement of the last fully connected layer, the weights of the pre-rained network are fine-tune by continuing the backpropagation. It is possible to either fine tune all the layers by adjusting their weights or keeping lower layers static and only fine-tune some higher level layers. The lower layers are normally extract general features but it become more specific when it goes to higher level layers [24].

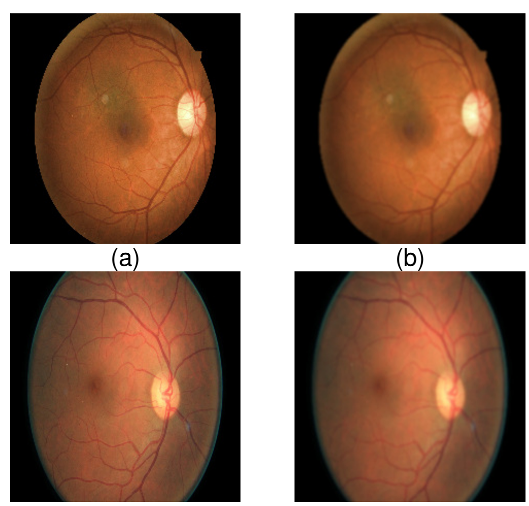
IV. DATASET AND EXPERIMENTAL RESULTS

This section explains the dataset and the results obtained during an experiment.

Dataset Description: For diabetic retinopathy detection, we have taken the dataset with resized images from kaggle [25]. The dataset contains 35,126 retina scan images. This dataset is based on the original dataset available at kaggle for Diabetic Retinopathy Detection in the 2015 competition [26, 4]. Images in this dataset are resized from the original images into size of 224 × 224 pixels. This is the standard size that can be used with many pre-trained models. The images are organized in five folders respective to their class. The folders are ranges from 0 to 4 where 0 represents No DR, 1 represents Mild, 2 represents Moderate, 3 represents Severe and 4 represents Proliferate DR. There are 35,126 images labeled as No DR, 2443 images labeled as Mild, 5292 images labeled as Moderate, 873 labeled as severe and 708 labeled as Proliferate DR.

Data pre-processing: In data pre-processing, often normalization is carried out first. Images are defined as matrices of pixel values. Normalization defines as a process that changes the range of pixel intensity values[25][26]. Colored images are defined by RGB where pixel range is between 0 and 255. If it passed in its original format, it may make the training and evaluation of neural network model difficult. Therefore, it is necessary to scale the pixel values before passing it into CNN model. For that, pixel values are scaled having a value in range between 0 to 1. For that, it is necessary to divide all pixels values by the largest pixel value 255 [27, 28].

After normalization, a Gaussian blur is applied to an image by a Gaussian function. It helps to reduce the noise from an image. Blurring or smoothing images using Gaussian blur removes rapid changes in pixel intensity by removing outlier pixels. These outlier pixels may be noise existed in an image [29]. The following Fig. 4 represented the effect of applying Gaussian blur to the left and right eye fundus images.



(a) (b) (c) (d)

Fig. 4. (a) original fundus image of right eye (b) after applying Gaussian Blur (c) original fundus image of left eye (d) after applying Gaussian Blur

Data Augmentation: CNN model requires enough training data to train a model which is not satisfied in the domain of medical data analysis due to unavailability of labeled data. For that data augmentation is necessary that reduces overfitting of the model and thus improves the localization capability. We have applied real time augmentation, where during each epoch; a radom augmentation of images was performed. It helps to solve the problem of less data during training of the network[30].

Hardware and Software: We have carried out experiment using Keras and Tensorflow deep learning packages. We have used Google Colab notebook and executed code on Google's GPU using Google's cloud servers. Colab supports a development environment in Python that makes easy to use other packages like NumPy, Matplotlib etc. We have used these packages for data visualization.

Experiments: VGG16 and MobileNet V1 models were trained on the acquired dataset using transfer learning and fine tuning. For training, we used a batch size of 32, epoch as 25 and learning rate as 0.001 with SGD optimizer. We have split the dataset into training, validation and testing by keeping ratio of 70-20-10 respectively. We have also measured losses occurred during training and validation. Losses are the errors occurred in the process of prediction while training of the model. The optimum training process always reduces the errors and increases the accuracy. When consistent accuracy and loss obtained, training could be stopped.

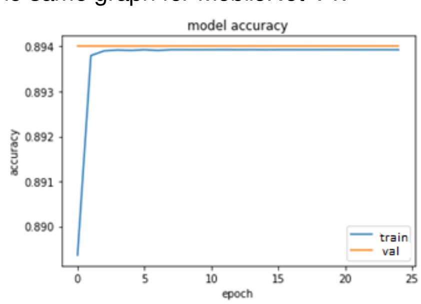
Using Pre-trained CNN for feature extraction by replacing the last fully connected layer: We have experimented two approaches where first is using pre-trained CNN as a feature extractor and replaced only the last fully connected layer with dense layer with 256 nodes with Relu activation function, added dropout to avoid overfitting and classification layer with 5 classes that are respective to No DR, Mild, Moderate, Severe and Proliferative DR and Softmax activation function. All other layers were freeze during the training. The following Table 1 represented the results of experiment carried out.

Table 1: Results of using pre-trained CNN as a feature extractor

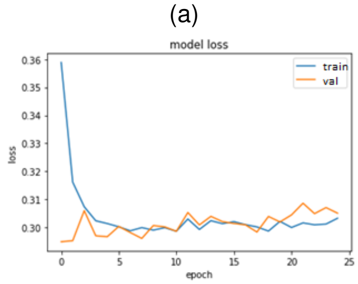
CNN Model	Number of Training Parameters	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Testing Accuracy	Testing Loss
VGG16	21M	89.38%	0.3033	89.39%	0.3051	89.40%	0.3051
MobileNet V1	16M	89.39%	0.3299	89.40%	0.4554	89.40%	0.4554

Here, both of the models provide almost same accuracy where VGG16 achieved 89.38% training accuracy and MobileNet V1 achieved 89.39% training accuracy. VGG16 achieved 89.39% validation accuracy and MobileNet V1 achieved 89.40% validation accuracy which is similar to each other. Both of the model obtained accuracy of 89.40% while testing.

The parameters learned are 21M and 16M respectively for VGG16 and MobileNet V1. The following figure 5 shows the graph of accuracy and loss for both training and validation obtained by VGG16. While figure 6 shows the same graph for MobileNet V1.

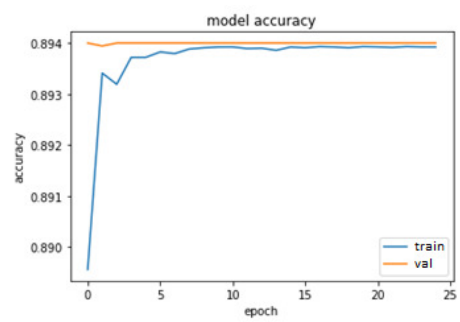


(a)

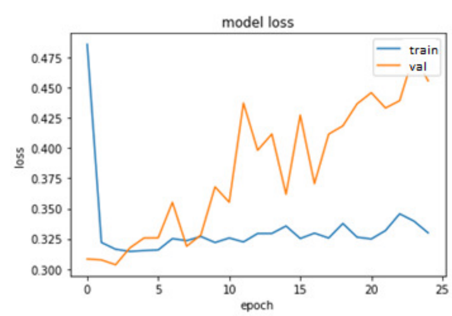


(b)

Fig 5. (a) Model accuracy and (b) loss obtained using pre-trained CNN as a feature extractor for VGG16.



(a)



(b)

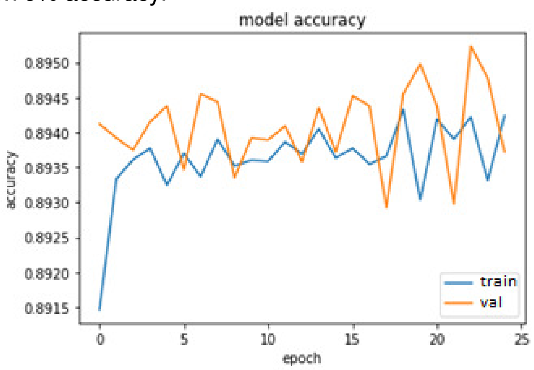
Fig 6. (a) Model accuracy and b) loss obtained using pre-trained CNN as a feature extractor for MobileNetV1

Fine-tune the pre-trained CNN model by replacing set of convolutional layers: The second approach is to fine-tune the pre-trained model by keeping lower layers freeze and fine tune the higher layers along with fully connected last layer. We have freeze all the layers of VGG16 and MobileNetV1 except last fully connected layer and three more convolutional layers. We have retrained the last three convolutional layers and replaced the last fully connected layer with dense layer that has 256 nodes with Relu activation function, added dropout to avoid overfitting and classification layer with 5 classes that are respective to No DR, Mild, Moderate, Severe and Proliferative DR and Softmax activation function. Therefore, training is required to these layers to fine tune the pre-trained CNN model on our dataset. The following Table 2 represented the results of experiment carried out using the above mentioned approach.

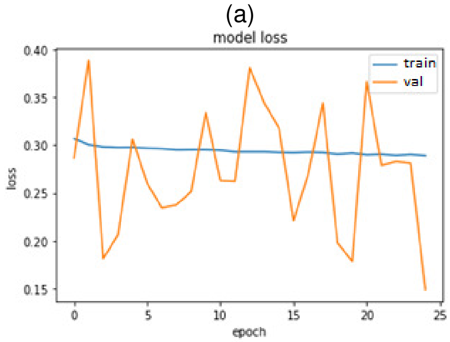
Table 2: Results of fine tune the pre-trained CNN models.

CNN Model	Number of Training Parameters	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Testing Accuracy	Testing Loss
VGG16	21M	89.42%	0.2890	89.52%	0.2829	89.51%	0.2830
MobileNet V1	16M	89.84%	0.2689	88.79%	0.3215	89.77%	0.3210

Here, VGG16 achieved 89.42% training accuracy and MobileNet V1 achieved 89.84% training accuracy, which is slightly better than VGG16. During validation, VGG16 achieved 89.52% accuracy and MobileNet V1 achieved 88.79% accuracy.

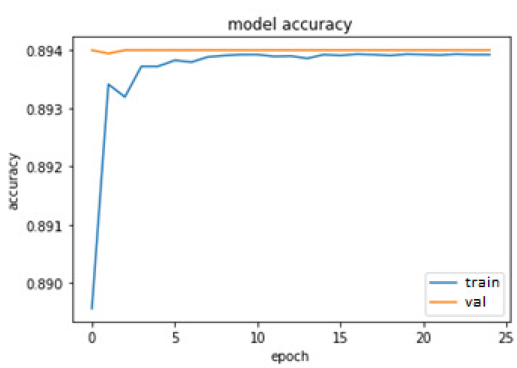


(a)

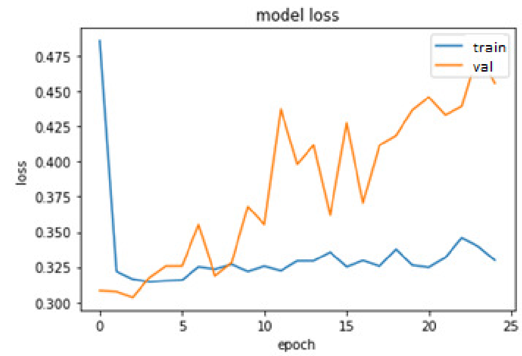


(b)

Fig. 7. (a) Model accuracy and b) loss obtained during fine tune the pre-trained CNN for VGG16



(a)



(b)

Fig 8. (a) Model accuracy and b) loss obtained during fine tune the pre-trained CNN for MobileNetV1

Finally, MobileNetV1 obtained 89.77% testing accuracy which is higher than VGG16. The following figure 7 shows the graph of accuracy and loss for both training and validation obtained by VGG16. While figure 8 shows the same graph for MobileNet V1.

V. DISCUSSION AND CONCLUSION

There is a great demand in availability of automated diagnostic system for diabetic retinopathy. It is always acceptable to have devices that directly diagnosis the disease from the fundus image without much clinical intervention. We have experimented the pre-trained VGG16 and MobileNetV1 using transfer learning and fine tuning on relatively small dataset that contains 5 classes and 35,126 ratinal fundus images. From the results obtained, it is concluded that fine tuning the pre-trained CNN model by replacing higher level layers is giving more accuracy compare to only replace the last fully connected layer. Moreover, MobileNetV1 provided slightly better accuracy compare to VGG16. MobileNetV1 is computationally less expensive CNN model and specially designed to use with embedded systems. In future, it is possible to devise an automated embedded device for diabetic retinopathy detection using MobileNetV1 as a pre-trained CNN model.

VI. FUTURE SCOPE

Experiment can be performed on different dataset to evaluate the results. Also, CNN network can be trained from scratch to see the impact on accuracy.

ACKNOWLEDGEMENTS

We are grateful to kaggle repository to provide the dataset for experiment.

Conflict of Interest. There is no conflict of interest.

REFERENCES

[1]. Xu, K., Feng, D., Mi, H. (2017). Deep Convolutional Neural Network-Based Early Automated Detection of Diabetic Retinopathy Using Fundus Image. Molecules.22(12), 2054. Published 2017 Nov 23. doi:10.3390/molecules22122054

[2]. Riaz, H., Park, J., Choi, H., Kim, H., & Kim, J. (2020). Deep and Densely Connected Networks for Classification of Diabetic Retinopathy. Diagnostics (Basel, Switzerland), *10*(1), 24. https://doi.org/10.3390/diagnostics10010024

[3]. Colomer, A., Igual, J., & Naranjo, V. (2020). Detection of Early Signs of Diabetic Retinopathy Based on Textural and Morphological Information in Fundus Images. Sensors (Basel, Switzerland*), 20*(4), 1005. https://doi.org/10.3390/s20041005

[4]. Cuadros, J., Bresnick, G., EyePACS. (2009). An Adaptable Telemedicine System for Diabetic Retinopathy Screening. Journal of diabetes science and technology (Online). *3*(3), 509-516.

[5]. Tanthuwapathom, R., Hnoohom, N. (2018). Detection of Diabetic Retinopathy Using Image Processing. In: Theeramunkong T., Kongkachandra R., Supnithi T. (eds) Advances in Natural Language Processing, Intelligent Informatics and Smart Technology. SNLP 2016. Advances in Intelligent Systems and Computing, *684*. Springer, Cham.

[6]. Ratanapakorn, T. Daengphoonphol, A., Eua-Anant, N., Yospaiboon, Y. (2019). Digital image processing software for diagnosing diabetic retinopathy from fundus photograph. Clin Ophthalmol. *13*. 641-648. Published 2019 Apr 17. doi:10.2147/OPTH.S195617

[7]. Winder, R. J., Morrow, P. J., McRitchie, I. N., Bailie, J. R., & Hart, P. M. (2009). Algorithms for digital image processing in diabetic retinopathy. Computerized medical imaging and graphics : *the official journal of the Computerized Medical Imaging Society, 33*(8), 608–622. https://doi.org/10.1016/j.compmedimag.2009.06.003

[8]. Bhatia, K., Arora, S., & Tomar, R. (2016). Diagnosis of diabetic retinopathy using machine learning classification algorithm. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT),* 347-351.

[9]. Al-hazaimeh, Obaida & Nahar, Khalid & Al-Naami, Bassam & Gharaibeh, Nasr. (2018). An effective image processing method for detection of diabetic retinopathy diseases from retinal fundus images. *International Journal of Signal and Imaging Systems Engineering, 11*. 206. 10.1504/IJSISE.2018.10015063.

[10]. Z. Gao, J. Li, J. Guo, Y. Chen, Z. Yi and J. Zhong. (2019). Diagnosis of Diabetic Retinopathy Using Deep Neural Networks. *IEEE Access. 7,* 3360-3370, doi: 10.1109/ACCESS.2018.2888639.

[11]. Harry Pratt, Frans Coenen, Deborah M. Broadbent, Simon P. Harding, Yalin Zheng. (2016). Convolutional Neural Networks for Diabetic Retinopathy, *Procedia Computer Science, 90*, 200-205.

[12]. Lam, C., Yi, D., Guo, M., & Lindsey, T. (2018). Automated Detection of Diabetic Retinopathy using Deep Learning. AMIA Joint Summits on Translational Science proceedings. *AMIA Joint Summits on Translational Science,* 147–155.

[13]. Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., & Webster, D. R. (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA, 316*(22), 2402–2410. https://doi.org/10.1001/jama.2016.17216

[14]. Kumar, A., Khandelwal, A. (2019). Analysis of Multimodal Medical Image Fusion using Discrete Wavelet Transform. *International Journal of Electrical, Electronics and Computer Engineering, 8*(2), 01-05.

[15]. Patel, A., Khandelwal, A. (2019). A Review of Techniques used for Edge Detection in Image. *International Journal of Electrical, Electronics and Computer Engineering, 8*(2), 06-09.

[16]. Patel, S. (2020). A Comprehensive Analysis of Convolutional Neural Network Models. *International Journal of Advanced Science and Technology, 29*(04), 771 - 777. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/4654

[17]. Patel R, Patel S. (2020). A Comprehensive Study of Applying Convolutional Neural Network for Computer Vision. *International Journal of Advanced Science and Technology, 29*(6s), 2161-2174. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/10929

[18]. Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

[19]. VGG16 – Convolutional Network for Classification and Detection, https://neurohive.io/en/popular-networks/vgg16/

[20]. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv:1704.04861.

[21]. ik-Ho Tsang , Review: MobileNetV1 — Depthwise Separable Convolution (Light Weight Model) https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69

[22]. https://cs231n.github.io/transfer-learning/

[23]. Kaur, T., Gandhi, T.K.(2020). Deep convolutional neural networks with transfer learning for automated brain image classification. Machine Vision and Applications 31, 20. https://doi.org/10.1007/s00138-020-01069-2.

[24]. Dipanjan (DJ) Sarkar, A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learninghttps://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

[25]. https://www.kaggle.com/sovitrath/diabetic-retinopathy-2015-data-colored-resized

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

[27]. Jason Brownlee, How to Normalize, Center, and Standardize Image Pixels in Keras, https://machinelearningmastery.com/how-to-normalize-center-and-standardize-images-with-the-imagedatagenerator-in-keras/

[28]. Jason Brownlee, How to Manually Scale Image Pixel Data for Deep Learning, https://machinelearningmastery.com/how-to-manually-scale-image-pixel-data-for-deep-learning/

[29]. https://datacarpentry.org/image-processing/06-blurring/

[30]. Harry Pratt, Frans Coenen, Deborah M. Broadbent, Simon P. Harding, Yalin Zheng. (2016). Convolutional Neural Networks for Diabetic Retinopathy, *Procedia Computer Science, 90*, 200-205.

How to cite this article: Patel, S. (2020). Diabetic Retinopathy Detection and Classification using Pre-trained Convolutional Neural Networks. *International Journal on Emerging Technologies*, *11*(3): 1082–1087.