DiRetNet - A Deep Convolutional Neural Network for determining Diabetic Retinopathy Category from Retinal Images

1st Nachiket Makwana
Information Technology Department
St. Francis Institute of Technology
Mumbai, India
nachimak28@gmail.com

2nd Akash Dabhi
Information Technology Department
St. Francis Institute of Technology
Mumbai, India
akashdabhi03@yahoo.com

3rd Archit Masurkar
Information Technology Department
St. Francis Institute of Technology
Mumbai, India
architmasurkar21@gmail.com

4th Sarvesh Narkar
Information Technology Department
St. Francis Institute of Technology
Mumbai, India
sarveshnarkar2d104@gmail.com

5th Dr. Vaishali Jadhav
Information Technology Department
St. Francis Institute of Technology
Mumbai, India
vaishalijadhav@sfitengg.org

***Abstract*—This paper proposes a model that will classify retinal images having Diabetic Retinopathy using Convolutional Neural Networks. The model will take retinal images as input and will classify them into one of the following five categories of Diabetic Retinopathy: None, Mild, Moderate, Severe and Proliferative. Our algorithm is trained on a dataset containing 25,000 retinal images, labelled by trained ophthalmologist. This model achieves a training accuracy of 94% and test accuracy of 92%.**

***Keywords*—Diabetic Retinopathy, Deep Learning, Convolutional Neural Networks, medical diagnosis, healthcare**

I. Introduction

In 2017, diabetes affected approximately 425 million adults (20-79 years) which will rise to 629 million by 2045 [1]. One of the leading causes of blindness is Diabetic Retinopathy (DR) which occurs due to diabetes. More than 65 million people in India are affected due to Diabetes. Approximately, one out of every ten persons with diabetes is affected by retinopathy, diabetes related eye disease. The prevalence of diabetic retinopathy, however increases with the duration of diabetes with the rate being 18% in urban Indian population.

On a global basis, DR is the leading cause of blindness. From the estimated 285 million people with diabetes, every third person has signs of DR and Diabetic Macular Edema.

Due to DR, the retina of the patient is damaged which may cause blindness or other retinal complications. Fig. 1 shows the effect of vision problems when a person is affected with DR. The basic causes of DR include high sugar level in blood, high blood pressure and high cholesterol, etc. After a long period of time, the blood vessels in the retina get affected due to diabetes. This causes the blood vessels to leak blood and other fluids thereby leading to blurred vision. Also, there are cases when the patient sees black spots in vision. These are called as floaters which lead to difficulty in having a clear vision. Hence the earliest signs of DR include blurred visions, floaters, dark areas of vision, etc. More than half of the cases of DR can be prevented by regular monitoring and checking of eyes of the affected patients. The chances of a person suffering from DR is directly proportional to the amount of time the particular person has diabetes. Currently DR is detected by trained ophthalmologists using various techniques such as Ophthalmoscopy and slit lamp exam, Gonioscopy, Tonometry, etc. These tests detect DR by measuring the pressure inside the eye or examine the eye area from which the fluid drains out, etc. However, the decisions of ophthalmologists for the same patient vary greatly which causes ambiguous diagnosis. The proposed system can be used for instant and correct diagnosis of DR and prove beneficial for those affected. In the sections to follow, the inadequacies of the current DR diagnosing system are highlighted and how the proposed model reduces those drawbacks by using Inception v3 algorithm is explained.

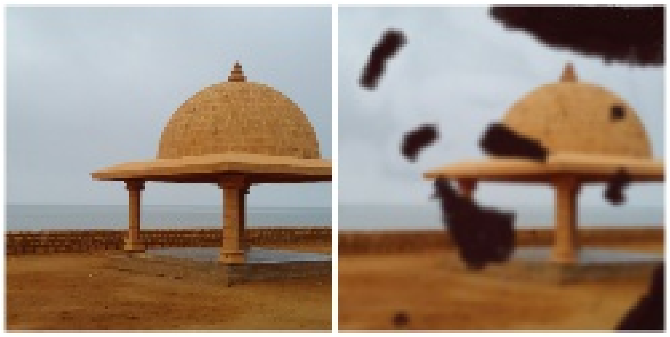

Fig. 1. Normal vision v/s vision of a person with diabetic retinopathy

978-1-5386-5995-3/18/$31.00 ©2018 IEEE

II. Related Work

Pranav Rajpurkar et al [2] proposed a model ChexNet that is used for detecting pneumonia from chest x-rays. ChexNet is a Dense Convolutional Neural network, having 121 layers is trained on Chest X-Ray 14 dataset. It is a binary classification problem indicating whether the patient has pneumonia or not. Their network consists of weights which are pre-initialized as per the ImageNet weights. Also they have retrained only the last layer of the network in order to give a single output.

Varun Gulshan et al [3] developed a deep learning model for detecting diabetic retinopathy using retinal fundus images. The neural network used was Inception v3 architecture. They trained a single network which made multiple binary predictions viz. moderate or worse DR (moderate, severe and proliferative) and severe or worse DR. A group of 10 networks was trained on the same data and linear average of all the predictions was computed and selected as the final output.

III. Input Dataset

The dataset used is provided by Kaggle.com [4] which hosts machine learning competitions. The dataset consisted of 35000 annotated images for category 0,1,2,3,4. Kaggle has collected this data from EyePacs which is a free platform for diabetic retinopathy screening. Fig. 2 shows the glimpse of the dataset. Each image is of size 4000 pixels in width and 2500 pixels in height. This dataset was very unclean and the distribution of images among the 5 categories was unequal. The image distribution spanning all 5 categories is as shown in Table 1. Due to this unequal distribution, data augmentation had to be performed. If this unequalized data would be fed as it is to the network, then the network outputs would get biased towards to the category which has maximum number of samples which in our case, is category 0.

TABLE I
Distribution of Images per category in the dataset

Category	Number of Images	Percentage
0	25810	73.47%
1	2443	6.95%
2	5292	15.06%
3	873	2.48%
4	708	2.04%
Total	35126	100%

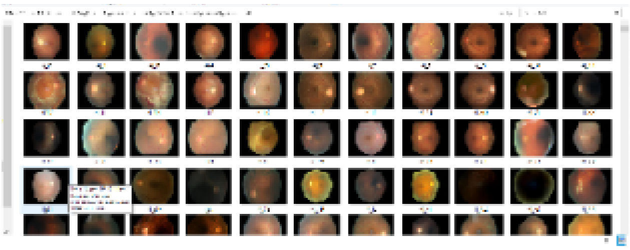

Fig. 2. Input Data-set

IV. Methodology

A. Data Augmentation

One of the major challenges faced during the training process was inadequate image dataset for Mild and Moderate categories. To overcome this challenge, data augmentation was used which drastically increased the number of images in these categories. Data Augmentation is a useful technique to overcome the shortage of training examples by applying various transformations on the image. Some of the commonly used data augmentation techniques are flipping, mirroring, random crop [5], colour shifting, etc. However, transformations like zooming, rotation, sheer, translation, etc. are not used widely due to the complexities involved in feeding such images to the training model. Performing data augmentation also helps the training model to identify various perspectives of images which can be fed as input once the model is ready to use.

B. Data Pre-processing

Data preprocessing is done to enhance the features of the image which leads to higher accuracy of the system. The retinal images acquired are of dimensions 4000x2000 pixels approximately which are then rescaled to 299x299 pixels. The circular retina in each image is not of uniform size, hence this resizing is done in order to compensate for this diametric variation of retinas in each image. All these resized images were then rescaled, dividing each pixel intensity by 255. The sole reason to do this is because RGB coefficients in the range 0-255 are too high for CNNs to process and they must be rescaled to values between 0 and 1 [6]. Furthermore, each sample was normalized by dividing all pixels by the standard deviation. This concept is same as that of batch normalization, the one which takes place in the CNN network.

V. Convolutional Neural Network model

Initially some of the common supervised learning algorithms were used such as K-Nearest neighbours (K-NNs) and Support Vector Machines (SVMs). For the proposed classification task these algorithms failed in classifying the images in our dataset. An accuracy of 25% was achieved for KNN and 24% for SVMs. This is because the feature extraction capabilities of KNNs and SVMs is lower as compared to CNNs [7]. This can be justified because an accuracy of around 66% was achieved in the initial stages of our research when only the final layer of the Inception V3 architecture [8] was trained. The diagram of the inception block is given in the Fig. 3.

Many such inception blocks are stacked over each other to make a deep network which is called as GoogLeNet [9] but is generally referred as Inception V3 model. As previously mentioned, the Inception V3 model has been used and it has been retrained for the proposed classification task. The understanding of doing this was derived from the concept of Transfer learning. It is a widely used concept by many deep learning researchers. Transfer learning must be used when there is a lot of data for the problem we are transferring learning from and usually relatively less data for the problem

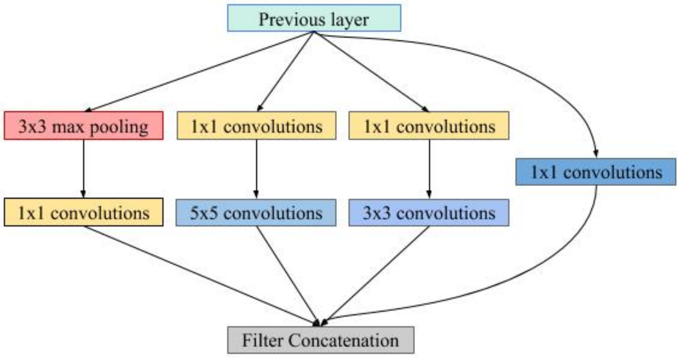

Fig. 3. Inception module

we are transferring learning to. In our case, we are transferring the learning of Inception V3, which is trained on 1.2 million images of the ImageNet challenge, to our classification task which consists of 25000 retinal images. The underlying logic behind this is that the low level features such as random shapes, edges, lines etc. contained in the weights of the lower layers of the Inception V3 are very helpful for any custom image classification task. In the proposed implementation, the top layer of Inception V3, which is a softmax layer for the default 1000 category classification of ImageNet [10], [11], has been removed. Instead, a Global Average Pooling [12] layer, a fully connected layer of 1024 neurons, a DropOut [13] layer with a ratio of 25% connections to be cutoff between the fully connected layers and a final fully connected layer with 5 neurons and a softmax activation function, which maps the outputs for the proposed classification categories, have been added. The softmax activation function is the function which reports the probabilities of the input being of a particular category. It is the standard function used in case of multi-class classification problems. The equation of the softmax activation is given as follows:

$$\begin{aligned} If\ \ z^{[L]} &= w^{[L]} * a^{[L-1]} + b^{[L]} \\ Then\ \ t &= e^{(z^{[L]})} \\ a^{[L]} &= \frac{t_j}{\sum_{j=1}^{n}(t_j)} \end{aligned} \tag{1}$$

Where L = layer number,
$z^{[L]}$ = Intermediate value at layer L,
$w^{[L]}$=Weights of layer L,
$a^{[L-1]}$= Activations of previous layer(outputs of layer L-1, inputs to layer L),
$b^{[L]}$= Bias value of layer L

These layers are added over the Inception V3 model in a sequence. The weights of the lower layer layers of the Inception V3 architecture have been frozen and only the top two blocks (layer 249 to layer 313) have been retrained. The final architecture can be seen in Fig 4. Initially only the top 3 layers were trained which were added for a few epochs and then training was done for the top two blocks along with the 3 final layers for more epochs. The optimizer used in training of the model was Adam optimizer [14] with a learning rate of 0.001. The batch size set is of 64. Rest of the parameters of Adam were kept default. The important considerations in the input to the Inception network is that an image of size less than 139 pixels height and width cannot be supplied. The image size goes on decreasing as the convolution operations are applied which is governed by the following formulae:

$$\begin{aligned} n_h^{[l]} &= \left(\frac{n_h^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right), \\ n_w^{[l]} &= \left(\frac{n_w^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right) \end{aligned} \tag{2}$$

Output Image dimensions:(n_h, n_w, n_c)
Where
n_h = Height of image (number of pixels),
n_w = Width of image (number of pixels),
[l] = Layer number
p = Padding applied to image,
f = Convolutional Kenrel size
s = Strides applied in a convolution layer
n_c = Number of channels, also referred to as depth

If the image size in a layer decreases up to a limit where the image height and width become less than the filter size of a convolution layer, then mathematically, this operation is not possible. This causes the training to stop as an error occurs. The optimum sizes that can be supplied as input are 299/256/224/192/139 pixels. Increasing the image size to a maximum 299 can give a good accuracy but trains slowly as the input image is large enough. The optimizer used in training is the Adam optimizer. The training has been carried out in 2 steps: training the final 5 layers while keeping the inception V3 layers frozen for a few epochs and then retraining the top 2 blocks of inception V3 architecture along with the pretrained 5 layers. In the first step as well as second step, the Adam optimizer has been used with default parameters. The default parameter values are: learning rate = 0.001, beta_1 = 0.9 and beta_2 = 0.999 with no learning rate decay. The training has been done on two GPUs - Nvidia GeForce 940MX with VRAM 4GB and Nvidia GTX 1060 with VRAM (Virtual RAM) 6GB. The training time varied drastically on both the GPUs. It took 10 hours to train on Nvidia 940MX as opposed to just 1.5 hours for the same task on the GTX 1060 owing to the large number of CUDA (Compute Unied Device Architecture) cores in the later one and increased capacity of the VRAM to hold the model parameters. Using a faster GPU with a larger VRAM is beneficial in terms of performance as well as accuracy. The memory limitations can hamper the models performance and may cause a decrease in the accuracy because it may dump some values out of the memory. This is done in order to keep the model training going.

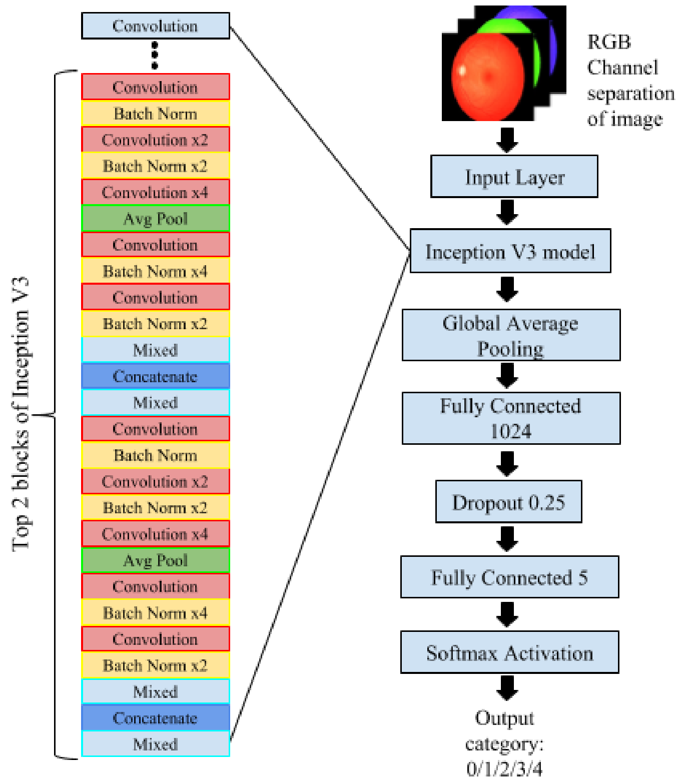

Fig. 4. Architectural Diagram of CNN Model

TABLE II
SUMMARY OF RESULTS

Approach Number	**Number of images per category**					**Channel Mode**	**Number of Epochs**	**Training Accuracy**	**Validation Accuracy**	**Training Loss**	**Validation Loss**
	0	**1**	**2**	**3**	**4**						
1	5855	4964	5844	4311	3163	Grayscale	20	90.05%	28.02%	0.3115	3.168
2	5855	4964	5844	4311	3163	Color	30	90.02%	82.96%	0.2916	0.4975
3	5855	4964	-	4311	3163	Color	4+20	94.70%	92.85%	0.1705	0.2154
4	3163	3163	3163	3163	3163	Color	4+20	87.67%	70.36%	0.3552	0.9003
5	5855	4964	5844	4311	3163	Color	4+20	90.85%	84.55%	0.2934	0.4468

VI. RESULTS

The summary of the training can be found in the Table 2. For every epoch there are certain steps per epoch whose exact number is given by the total samples in the dataset divided by the batch size. The batch size was kept 64 owing to the large dataset to maintain a trade off between the accuracy and training time. The number of epochs in the table with pattern 4+20 indicate that the training was done in 2 steps: training the final 5 layers for 4 epochs until the accuracy saturated to around 45% and in the second step, running 20 epochs training the top 2 blocks of inception V3 along with the top 5 layers until desirable accuracy was achieved. Each row in the above table corresponds to various approaches that were tried out during the training. In the first approach, all the coloured (3 channel) images in the dataset were converted to grayscale images thereby reducing the number of channels to 1. The sole reason for using grayscale images was to reduce the computations required for training and hence reduce the time required for training the model. After training the model for 20 epochs, the training accuracy achieved was 90.05%, however the validation accuracy was just 28.02%. It was observed that the validation accuracy saturated to 28.02% after 15 epochs. This model failed because while converting the images to grayscale the important features of the images were lost. Thus, it was inferred that using single channel images did not help in achieving the desired accuracy and hence colored images need to be used for further training. In the second model training, color images were used. This time, the upper 3 layers were not trained for a few epochs, but started training them along with the top two blocks, hence the low accuracy and it can ne seen that the network is overfitting upto some extent. Also, in this training run, an issue faced was that the bias towards category 2 was larger than other categories and the prediction defaulted to category 2 for any category image provided. To mend this, another training run was conducted without the images for category 2, i.e. a classification for only 4 categories. This time a higher accuracy was achieved and the predictions given by the model were almost correct in each case. It ca be said that the category 2 images are pretty much same as any other category and due to the dearth of significant features present in them, the previous model got confused and gave some wrong predictions. In the 4th attempt of training, an important property of neural networks that they perform great when the training dataset is balanced was exploited. It was observed that the training and validation accuracies saturated and didnt show significant changes from epochs 18 to 20, so the training was stopped. Also, it can be seen that there i a huge difference of 17.31% which means that the networl has overfitted and this model would work better if each clas had more than 5000 images per class. In the final attemp of training, the original dataset was supplied as it is bu this time in color mode. The results show that the networl achieved a good training accuracy but ended up saturating a this level with a difference of 6% in the training and validatioi accuracies. It can be concluded, that the network trained ii the third attempt without the category 2 images in the datase This network has the best in class accuracy and looks ver promising as the difference between the training and validatioi accuracies is only 1.85%. Thus, it was decided to proceed witl this same model which classifies the images into 4 categories viz. Category 0, 1, 3 and 4. The graphs for the third approach are as as illustrated in Fig 5(a) and 5(b).

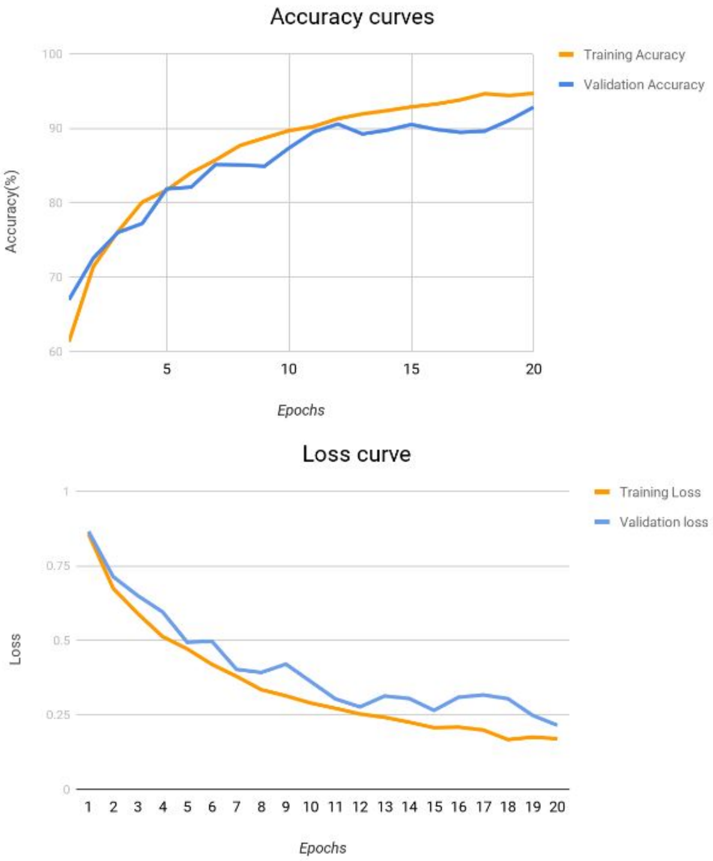

Fig. 5. (a) Training v/s Validation accuracy (b) Training v/s Validation loss

VII. Discussion

This project has been done using Inception V3 as the base model. Multiple approaches were tried using the same base model. The further work that can be done is to use another such base model. There are multiple pretrained models available such as ResNet [15], DenseNet [16], HighwayNet [17], MobileNet [18] etc. Once a conclusion is found that which of the above mentioned approaches work and gives perfect results, one must try retraining the ResNet model. It is again, a state of the art network which gives a tough competition to Inception in the case of ImageNet large scale visual recognition challenge. Although the ResNet takes more time to train since it is deeper than Inception V3 and also has a larger memory requirement, the results must be compared in order to check which model fits the best for this particular retinal image classification task.

VIII. Future Work

As already stated, the aim of this project is to make DR screening services available in remote villages and to people who do not have access to trained ophthalmologists as well as retinoscopes. So, an idea of a handheld retinal camera has been proposed. This will contain the following components: Smartphone, an optical tube and a condensing lens. The lens used here is a 20D (Diopters) aspheric condensing lens. This optical tube and lens can be mounted on the phone case as shown in Fig. 6. The images can be captured using a standard smartphone camera and it's torch for illuminating the retinal surface. These captured can be fed directly to the deep learning model for prediction over a simple web interface.

IX. Conclusion

Diabetic Retinopathy is a rapidly growing cause of blindness all over the world. The estimate of affected population is predicted to rise to from 31 million to 79 million in the next decade. This project is an aid to the ophthalmologists in the early detection and prevention of Diabetic Retinopathy. The screening and diagnosis facilities in India are limited to urban areas and there is a dearth of such resources in remote rural areas. This deep learning model has been developed which has been successful in achieving state of the art accuracy in image classification which are comparable to a clinical setup. The aim of this project is to provide convenient access to rural areas for screening purposes where access to trained ophthalmologists is limited. These accurate results were obtained due to transfer learning and data augmentation. Data augmentation helped in tackling the problem of class imbalance which is a crucial factor while training CNNs. Also transfer learning done using the Inception V3 architecture has proved fruitful because of its virtue of generalization of low level features.

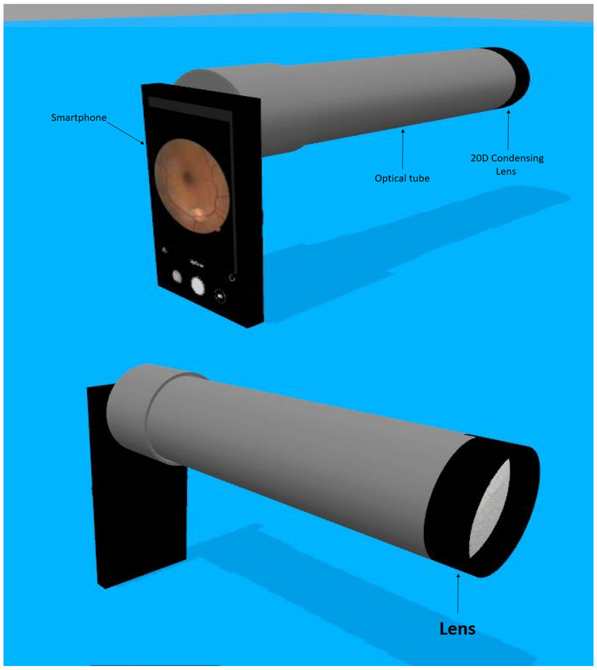

Fig. 6. (a) Front look of prototype (b) A view from the lens

References

[1] En.wikipedia.org. (2018). Diabetic retinopathy. [online] Available at: https://en.wikipedia.org/wiki/Diabetic_retinopathy [Accessed 11 Jun. 2017].

[2] Pranav Rajpurkar et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X- Rays with Deep Learning. In: CoRR abs/1711.05225 (2017). arXiv: 1711.05225. URL: http://arxiv.org/abs/1711.05225.

[3] Gulshan V, Peng L, Coram M, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA. 2016;316(22):24022410. doi:10.1001/jama.2016.17216

[4] Kaggle.com. (2018). Diabetic Retinopathy Detection — Kaggle. [online] Available at: https://www.kaggle.com/c/diabetic-retinopathy-detection [Accessed 5 Jun. 2017].

[5] Luis Perez and Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. In: CoRR abs/1712.04621 (2017). arXiv: 1712.04621. URL: http://arxiv.org/abs/1712.04621.

[6] Keras.io. (2018). Image Preprocessing - Keras Documentation. [online] Available at: https://keras.io/preprocessing/image/ [Accessed 23 Jul. 2017].

[7] Y.LeCun,B.Boser,J.S.Denker,D.Henderson,R.E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4):541551, Dec. 1989.

[8] Christian Szegedy et al. Rethinking the Inception Architecture for Computer Vision. In: CoRR abs/1512.00567 (2015). arXiv: 1512.00567. URL: http://arxiv.org/abs/1512. 00567

[9] Christian Szegedy et al. Going Deeper with Convolutions. In: CoRR abs/1409.4842 (2014).arXiv: 1409.4842. URL: http://arxiv.org/abs/1409.4842

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classication with deep convolutional neural networks.In:Advances in neural information processing systems.2012, pp. 10971105.

[11] Jia Deng et al. Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE. 2009, pp. 248255.

[12] Min Lin, Qiang Chen, and Shuicheng Yan. Network In Network. In: CoRR abs/1312.4400 (2013). arXiv: 1312.4400. URL: http://arxiv.org/abs/1312.4400

[13] Nitish Srivastava et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: Journal of Machine Learning Research 15 (2014), pp. 19291958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In: CoRR abs/1412.6980 (2014). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412. 6980.

[15] Kaiming He et al. Deep Residual Learning for Image Recognition. In: CoRR abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385

[16] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In: CoRR abs/1608.06993 (2016). arXiv: 1608.06993. URL: http://arxiv.org/abs/1608.06993.

[17] Rupesh Kumar Srivastava, Klaus Greff, and Jurgen Schmidhuber. Highway Networks. In: CoRR abs/1505.00387 (2015). arXiv: 1505.00387. URL: http://arxiv.org/abs/1505. 00387.

[18] Andrew G. Howard et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In: CoRR abs/1704.04861 (2017). arXiv: 1704.04861. URL: http://arxiv.org/abs/1704.04861.