2017 24th national and 2nd International Iranian Conference on Biomedical Engineering (ICBME), Amirkabir University of Technology, Tehran, Iran, 30 November - 1 December 2017

Comparative Study of Fine-Tuning of Pre-Trained Convolutional Neural Networks for Diabetic Retinopathy Screening

Saboora Mohammadian*, Ali Karsaz**
Electrical Engineering Department
Khorasan Institute of Higher Education
Mashhad, Iran
Emails: sa.m.roshan@khorasan.ac.ir*, karsaz@khorasan.ac.ir**

Yaser M. Roshan
Electrical Engineering Department
Point Park University
Pittsburgh, PA, US, 15222
Email: yroshan@pointpark.edu

***Abstract*—Diabetic retinopathy is the leading cause of blindness, engaging people in different ages. Early detection of the disease, although significantly important to control and cure it, is usually being overlooked due to the need for experienced examination. To this end, automatic diabetic retinopathy diagnostic methods are proposed to facilitate the examination process and act as the physician's helper. In this paper, automatic diagnosis of diabetic retinopathy using pre-trained convolutional neural networks is studied. Pre-trained networks are chosen to avoid the time- and resource-consuming training algorithms for designing a convolutional neural network from scratch. Each neural network is fine-tuned with the pre-processed dataset, and the fine-tuning parameters as well as the pre-trained neural networks are compared together. The result of this paper, introduces a fast approach to fine-tune pre-trained networks, by studying different tuning parameters and their effect on the overall system performance due to the specific application of diabetic retinopathy screening.**

Keywords- Diabetic retinopathy; convolutional neural network; deep learning; Inception model.

I. INTRODUCTION

Diabetic Retinopathy (DR) is the leading cause of blindness in adult ages from 20 to 74, and exhibits a serious risk for general population health. The disease occurs when diabetes damages blood vessels in the retina. It is estimated that the number of people diagnosed with DR will increase from 126.6 million in 2010 to 191 million by 2030, and the number of people with vision-threatening DR will grow from 37.3 million to 56.3 million by the same time [1]. Despite of this worrying statistics, evidence shows that early treatment can slow down the progression of DR [2]. However, the clinical challenge for early-diagnosis and treatment is that the patients with DR may not experience any symptoms until it becomes a serious treat for their vision.

The diagnosis of DR requires an experienced ophthalmologist to carefully investigate images of the retina (or fundus images). Fundus images provide valuable information regarding the presence of microaneurysms, hemorrhages, neovascularization, and exudates, and the presence of any of these can be related to DR. Fig. 1 illustrates a fundus image from a patient with DR and demonstrates the injuries to the retina [3]. The high cost of physical examination and lack of professional experts are two important obstacles for early DR diagnosis. Therefore, the common procedure for DR screening is not efficient enough and thus, health care providers miss large numbers of early stage DR cases [4].

Studies have shown that automatic screening systems can be utilized to diagnose DR accurately and consistently in early stages. There has been an increase in applying various machine learning techniques to classify images into DR and NO DR classes in the recent years [5]. The majority of these efforts use hand-crafted features of retinal images for training their systems. Some of these machine learning algorithms are Neural Network (NN), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) [6]-[9]. A comparative study of conventional feature-based machine learning algorithms is done which demonstrates the Linear SVM approach to be a reliable classification method for DR screening [10]. Extracting features from images is a time-consuming and complicated task that needs professional experts to study the images and infer the most relevant set of features and apply feature extractors to the images. These extracted features can be used for image classification via different machine learning approaches.

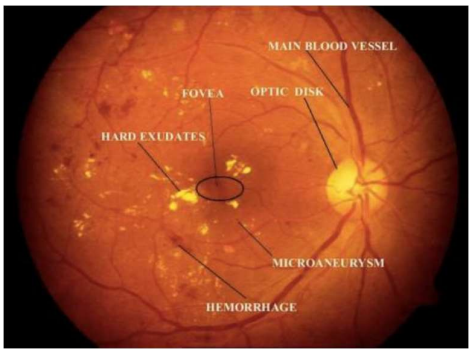


Figure 1. Example of a retinal photo with diabetic retinopathy.

To overcome the feature selection and extraction issues, there has been an increase in the studies using Convolutional Neural Networks (CNN) for medical image analysis [11]. The main advantage of CNN is its ability to extract features automatically via deep multiple layers [12]. Based on this advantage, CNN has been recently used in multiple medical applications mostly related to disease diagnosis. For instance, in the technique proposed by Pan et al. CNN was applied to MRI images to grade brain tumors [13]. Another CNN-based method was conducted by Roth et al., in which multi-level deep convolutional network is used for pancreas segmentation in computed tomography scans [14]. Related works on DR diagnosis utilizing CNN include, using CNN for feature extraction and ensemble classification for retinal blood vessel segmentation [15], classification of DR severity using CNN [16]. And a comparative study of two CNN structures with multiple filter sizes for DR recognition [17].

The main issue with the studies being done with CNN is proposing an architecture for the network that on one hand can demonstrate acceptable performance in the diagnostic application, and on the other hand can be trained considering the hardware limitations. The hardware limitations and familiarity with the database and images characteristics makes it more challenging to develop proper CNN structure and train it using the available databases [16]. One approach that has been used in the literature to address this issue is fine-tuning the pre-trained CNNs and customizing them based on specific applications. To this end, in this paper, two well-known pre-trained CNN architectures named Inception-V3 (also known as GoogLeNet) and Xception are employed and evaluated to classify fundus images into cases of DR patient or healthy. These CNN models were previously trained for classifying the ImageNet dataset and their weights have been published for future studies [18]. Generally speaking, the first layers of these CNNs are mostly related to extracting general information from the images such as the edges, while the last layers are specifically trained to extract more detailed features related to the images dataset specifically. Therefore, in the DR screening application the weights of the last layers of these networks are fine-tuned to adapt the networks for this application and increase the performance accuracy.

The comparative study which is done in this paper, introduces an approach for further research, to select a pre-trained CNN and fine-tune it such that the application requirement is reached with minimum hardware requirements. The effects of various parameters are compared in this study. The parameters include number of retrained layers, activation function, optimization function, and use of data augmentation. The results reported in this paper can be utilized as a starting point for further research and enhance the accuracy of the DR screening approaches using CNN, while being acceptable from the practical standpoint.

The rest of the paper is organized as follows. Section II briefly explains preparing the data set that is used for classification. Section III discusses the methodology of comparison CNN architectures. The results of the simulations and the comparison of the classifiers are presented in Section IV, while Section V concludes the paper.

II. Data Preparation

A. Diabetic Retinopathy Dataset

The dataset from a recent Kaggle competition is used in this paper [19]. Kaggle diabetic retinopathy dataset includes 35126 retina images, which are taken with different types of cameras. Various qualities of the images in the dataset make the feature extraction approaches more difficult to implement. For each individual, the images of his/her right and left eye are included in the dataset. Each image is rated by a clinician for the presence of diabetic retinopathy on a scale of 0 to 4. The scales of 0, 1, 2, 3, and 4 correspond to No DR, Mild, Moderate, Severe, and Proliferative DR, respectively. The aim of this study is to detect NO DR images (with the label 0) from DR ones (with labels 1-4). Therefore, there are two classes for classification in this study. However, the platform can be extended for classification of DR stages, as well.

To employ an image classification CNN, the images should be preprocessed to decrease the effects of camera variations as well as images quality differences due to different brightness and exposure settings. Also, a data augmentation approach is needed to enhance the robustness of the network to noises in the data (such as rotations in the images). Both preprocessing and data augmentation approaches are described in this section.

B. Image Preprocessing

To decrease the variation among images due to different camera resolutions and settings, an image preprocessing algorithm is applied to the images using OpenCV package. The first step of the algorithm is rescaling the images such that all the input images have the same size. In the next step, the color of each pixel is subtracted by the local average, mapping the average to 50% gray. Using this approach, the sharpness of the images will be more unified. Finally, the edges of the images are clipped to remove the "boundary effects" [20].

C. data augmentation

The most common method to reduce overfitting of the deep networks is to enlarge the datasets [21]. For this purpose, the dataset is augmented by shifting, rotating, and flipping the images in the middle of each training. Flipping the images includes horizontal flipping, vertical flipping, and horizontal plus vertical flipping. Using this approach, the dataset becomes larger, while the augmented images would not be stored into the system's memory and more space would not be needed.

Another benefit of this approach is increasing the robustness of the trained network to variations in the input image. In other words, the CNN is trained for variations of the original images to compensate for variations in the images taken due to user experience. Furthermore, the number of healthy eyes images in the Kaggle dataset are much more than the ones with DR. Therefore, the DR dataset is augmented to almost the same size as the No DR dataset.

III. CNN Architectures and Fine-Tuning

CNN is a kind of multilayer neural networks which typically consists of convolutional, subsampling, and fully connected (FC) layers [15]. Convolutional layer is the core of the network and works as a feature extraction layer. It performs the convolution operation over the input value. Therefore, all

neurons in a particular feature map shares the same set of weights and the same biases which helps them to detect features at the different positions on the input. Moreover, this weight sharing reduces the number of parameters that needs to be trained. Subsampling layer reduces the dimensionality of each feature map but keeps the most important information. This layer helps to reduce the amounts of learning parameters and is usually placed after the convolutional layer. Two common types of pooling layer are max pooling and average pooling. The output layer of a CNN is a FC layer of neurons at the end of the network. Neurons in a FC layer have full connections to all activations in the previous layer, as seen in a traditional multilayer neural network [22]-[24].

A. *Fine tuning*

In medical imaging and diagnosis field, it is relatively rare to have an image dataset of sufficient size to completely train a CNN from scratch [12]. In addition, the state of art convolutional neural networks included in the Keras core library demonstrates a strong ability to be generalized to images outside the ImageNet dataset via transfer learning, such as feature extraction and fine-tuning [21]. Therefore, it is very common to fine-tune a CNN that has been trained using a large labeled dataset from a different application to avoid training networks for many general features [25]. Training a CNN from scratch require a large amount of data as well as extensive computational and memory resources [12]. Besides, training a deep network with small dataset often leads to overfitting and convergence issues. In this paper, two pre-trained CNN architectures are fine-tuned and are used for DR classification. These models are Inception-V3 and Xception which are the CNN models that had been applied to the well-known ImageNet dataset.

1) *Inception-V3 Architecture*

The Inception architecture was first introduced by Szegedy et al. [26]. Inception module computes 1×1, 3×3, and 5×5 convolutions within the same module of the network and concatenates the output of the whole process to pass it to the next layer of the network. So, the Inception module is called "multi-layer feature extractor", and is shown in Figure 2. In a more recent publication Szegedy et al. introduced Inception-V3. That is an updated inception module to improve classification accuracy. Figure 3 demonstrates schematic diagram of Inception-V3 [27]. For brevity, a more detailed description of the network structure is avoided here, and the interested reader may refer to the cited literature.

2) *Xception Architecture*

Xception architecture designed by the extension of the Inception module [28]. In this architecture, inception modules are replaced with depth wise separable convolutions. In Xception architecture data goes through the entry flow, then it goes through the middle flow and finally through the exit flow. The entry flow includes the convolutional layers to extract features of the images, the middle flow is responsible to summarize the features extracted and find meaningful relationships, while the exit flow will perform the classification utilizing the extracted finalized features. The overview of the Xception architecture is demonstrated in Fig. 4 [28]. .

B. *Software and Hardware*

The software packages that are used in this paper are, Tensorflow, Numpy, h5py, Scikit-learn, OpenCv, and Keras. The latter one is an open source neural network library in Python, which includes several pre-trained CNN architectures. These networks represent some of the highest performing and well-known CNNs on the ImageNet dataset over the past few years. ImageNet is a large database (over 1.2 million images) that is used for visual object recognition (1000 separate object categories). The Tensorflow and Keras frameworks are installed on an Ubuntu operating system for the ease of implementation of CNN architectures, while the other mentioned packages are being used to perform common related calculations and image processing in Python.

Utilizing pre-trained CNNs for classification makes GPUs and external memories unnecessary. Utilizing this approach although the results may be slightly less accurate than designing an individual CNN from scratch but save training time and resources for the application, while providing a useful insight on the usage of CNN for each application. All the steps of the proposed method are done by an Intel i7 core CPU, with 8GB memory, which is considerably advantageous comparing to common CNN training hardware requirements [16], [29].

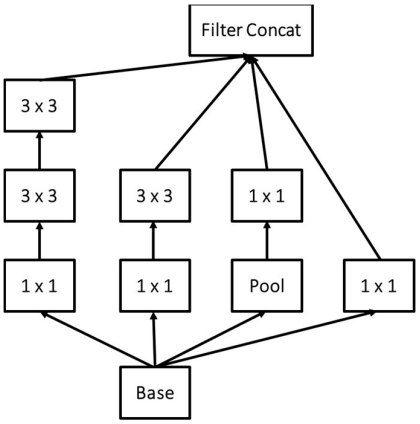


Figure 2. The structure of an Inception module [26]

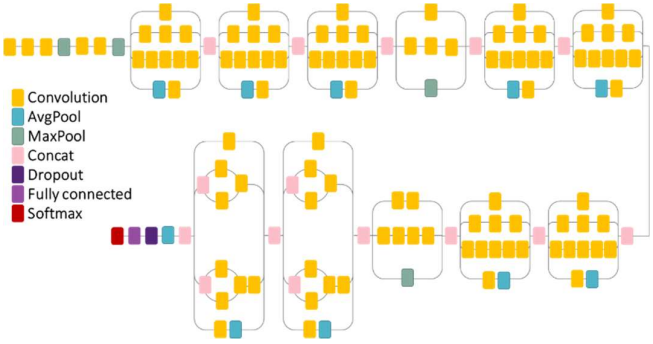


Figure 3. Schematic diagram of Inception-V3 [27]

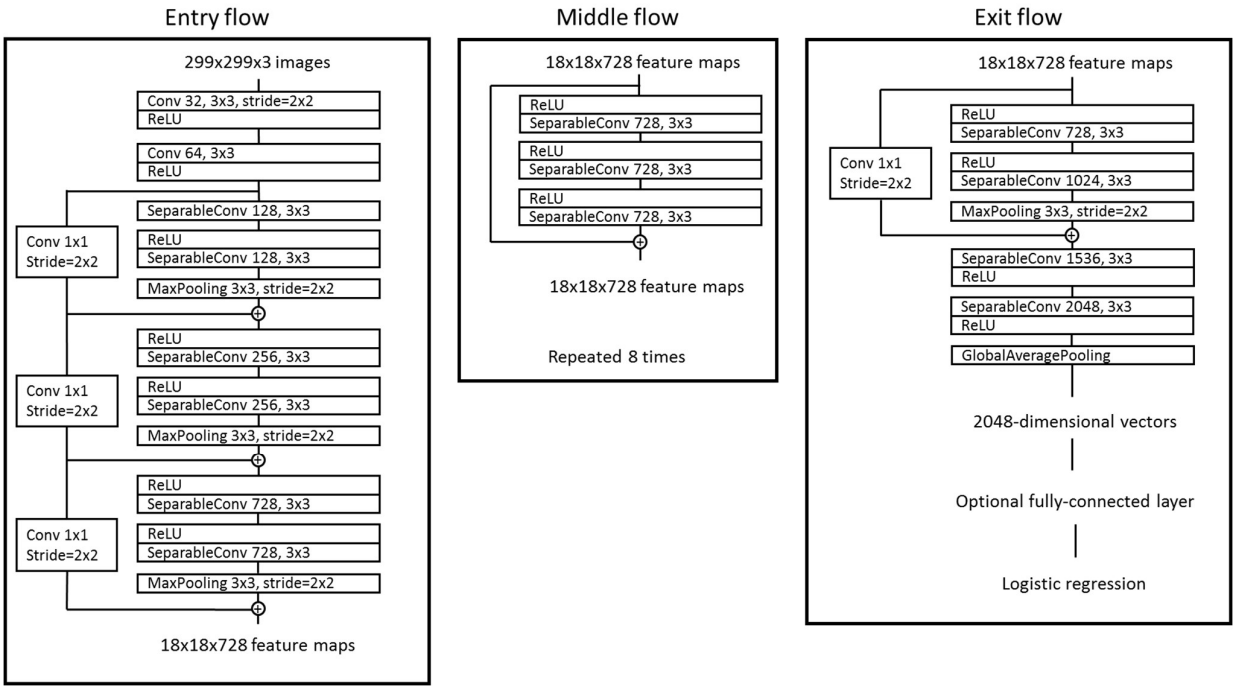


Figure 4. Schematic diagram of Xception [28]

IV. Results and Discussion

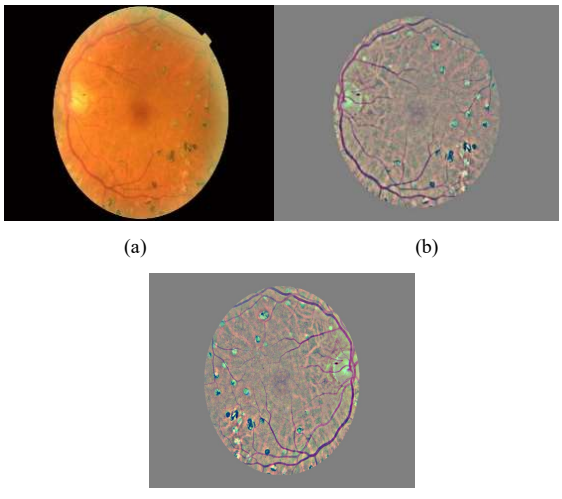
The dataset contains images with various levels of resolution. The first step of this study is to remove the variations using image preprocessing algorithms. The result of the algorithm described in Section II-B on an image with proliferative DR is demonstrated in Fig. 5a and 5b.

The data from Kaggle is used as the baseline for the classification problem, which includes 35126 images in which 25810 images are assigned as NO DR and the rest of the dataset are the images with signs of DR. In order to balance the size of images in two classes, various image augmentation techniques are applied to DR images randomly as mentioned in Section II-C. Therefore, the number of DR images increased to 25619 which is almost as same as NO DR images. A sample image which is horizontally flipped is demonstrated in Fig. 5c.

For training and testing phases of the classifications, 20 percent of the available data has been selected randomly as the testing set, while the others are being used as the training set. The testing and training sets are kept the same for all simulations, keeping the results comparable.

To fine-tune the CNNs, it is very common to retrain the last two blocks of the pre trained networks. Therefore, the first 172 layers of Inception-V3 and the first 115 layers of Xception networks have been frozen and the weights and biases of these layers are not changing during the training. The training process is applied to the rest of layers including Fully Connected layer at the end of each network. However, to complete the comparative study, two situations for which only the fully connected layer is trained while the CNN remained fixed, and 4 last blocks are unfrozen and retrained, are also tested. Needless to say, the training time varies for each experiment, mostly depending on the number of unfrozen blocks of the CNNs and is variable from 1 to 6 hours, using the mentioned hardware in Section III-B.

To employ Convolutional Neural Networks for image classification tasks, RELU (Rectified Linear Unit) and ELU (Exponential Linear Unit) are the two well-known activation functions, which are being utilized and compared in this study.



(a) (b)

(c)

Figure 5. (a) Proliferative DR image, (b) Preprocessed image, (c) horizontally flipped image

Also, SGD (Stochastic Gradient Descent) and ADAM (Adaptive Moment estimation) optimizers are being widely used to train the network. In the comparison stage, different activation functions are used while being trained utilizing the two common optimization approaches.

The first step in comparing the mentioned networks is to consider the effect of data augmentation. Before that the initial CNNs were setup with a RELU activation function for dense layer of the fully connected layer, and a SGD (learning rate=0.0001 and momentum=0.9) as the optimizer of the network. For achieving better performance and to avoid overfitting of the network, train dataset is augmented (rotation-range=40, width-shift-range=0.2, shear-range=0.2, height-shift-range=0.2, and zoom-range=0.2). The effect of the data augmentation on both networks is shown in the first two rows of Table I. All the accuracy results reported in Table I are based on the accuracy of the trained network for test dataset. Because of the promising performance of using data augmentation as it is shown in Table I, from now on, similar data augmentation is used for all of the proposed models.

To determine the effect of the activation function of the dense layer, both RELU and ELU are applied to the networks, while other parameters are kept constant. The accuracy for both networks, reported in rows 2 and 3 of Table I, demonstrates that RELU acts better than ELU in the DR screening application.

To avoid local minima, a state of art optimizer, ADAM, is proposed in literature and is used here to enhance the results of SGD. As shown in rows 2 and 4 of Table I, the ADAM optimizer shows better results than SGD in both CNN models. Therefore, ADAM can be used to improve the accuracy of the networks.

At last, to study the effect of fine tuning the earlier layers or only the fully connected layer (while the CNN layers are intact), two new cases are studied. In the first case, reported in row 5 of Table 1, the first 136 and 95 layers of the Inception-V3 and Xception have been frozen and the rest of the layers are fine-tuned (i.e., 4 sets of blocks are unfrozen for each network). For this case, even though more layers of the networks have been retrained, but the accuracy results demonstrate a decrease in accuracy. The main reasoning behind this issue is that the original CNNs are trained for a relatively larger a more comprehensive dataset, and the first sets of layers are generally tuned to extract less-detailed features of the data set. In other word, the pre-trained networks are capable of extracting edges, shapes, etc., and retraining these layers with smaller dataset, and lower iteration number, will only disturb the pre-trained weights and hence it will affect the accuracy results. In the second case, reported in row 6 of Table 1, all of the CNN layers are frozen and the retraining is done on the fully connected layer only. As the results demonstrate, in this case, and due to lack of training for detailed features extraction layers (last layers of CNN) the results are showing less accurate networks.

Overall, fine tuning the last two blocks of Inception-V3 model utilizing RELU as the activation function and ADAM optimizer demonstrates the best result of classifying diabetic retinopathy cases, with about 87% accuracy on the test dataset. The result of this work is significantly better, in terms of accuracy of the overall network as well as complexity of the hardware needed to re-train the network, than similar works reported in [16] and [29]. In these works, similar dataset is being used, however, a new CNN architecture is proposed which needed extensive GPU-based hardware to train the networks. It is worth noting that the purpose of this paper was not to design the most accurate DR screening network, but to demonstrate the effect of varying parameters in fine-tuning the available pre-trained CNN networks. Using the results of this study, one may enhance the results by increasing the iteration number for training (maximum 200 iterations in this study) or by designing a pre- or post-processing algorithm to manipulate the images for easier feature extraction or integrating the results of different networks to reach a cumulative DR screening result.

TABLE I. CLASSIFICATION PERFORMANCE FOR DIFFERENT EXPERIMENTS (ACCURACY RESULT ON TEST DATASET)

CNN Fine-Tuning Parameters	Inception-V3	Xception
Unfrozen blocks: ***2*** Activation function: ***RELU*** Optimizer: ***SGD*** With ***NO*** data augmentation	0.6048	0.6979
Unfrozen blocks: ***2*** Activation function: ***RELU*** Optimizer: ***SGD*** With data augmentation	0.8074	0.7860
Unfrozen blocks: ***2*** Activation function: ***ELU*** Optimizer: ***SGD*** With data augmentation	0.5341	0.5031
Unfrozen blocks: ***2*** Activation function: ***RELU*** Optimizer: ***ADAM*** With data augmentation	**0.8712**	0.7449
Unfrozen blocks: ***4*** Activation function: ***RELU*** Optimizer: ***ADAM*** With data augmentation	0.8570	0.5742
Unfrozen blocks: ***0*** (only fully-connected layer) Activation function: ***RELU*** Optimizer: ***ADAM*** With data augmentation	0.7314	0.6025

V. CONCLUSION

In this paper, a comparative study on fine-tuning of two well-known pre-trained convolutional neural networks is presented. The reasoning behind fine-tuning of a pre-trained network is to avoid a time- and resource-consuming training approach for the convolutional systems, while being able to leverage the pre-trained systems for their trained architecture on feature extraction. The comparative study is performed to demonstrate the effect of variations of different parameters (such as retrained layers, activation function, and optimization approach) of the networks on their accuracy in screening diabetic retinopathy cases. The results of this study can be utilized in further research to choose the best network parameters, and to propose novel pre- or post-processing algorithms to increase the diagnosis accuracy.

978-1-5386-3609-1/17/$31.00 ©2017 IEEE

REFERENCES

[1] N. Congdon, Y. Zheng, and M. He, "The worldwide epidemic of diabetic retinopathy," Indian Journal of Ophthalmology, vol. 60, no. 5, p. 428-431, 2012.

[2] B. Antal and A. Hajdu, "An ensemble-based system for Microaneurysm detection and diabetic retinopathy grading," IEEE Transactions on Biomeical Engineering, vol. 59, no. 6, pp. 1720–1726, Jun. 2012.

[3] G. Patry, G. Gauthier, B. Lay, J. Roger, and D. Elie, "ADCIS Download Third party: Messidor database," ADCIS S.A., 2016. [Online]. Available: http://messidor.crihan.fr. Accessed: Nov. 16, 2016.

[4] A. F. M. Hani and H. A. Nugroho, "Gaussian Bayes classifier for medical diagnosis and grading: Application to diabetic retinopathy," 2010 IEEE EMBS Conference on Biomedical Engineering and Sciences, Nov. 2010.

[5] M. Niemeijer et al., "Retinopathy online challenge: Automatic detection of Microaneurysms in digital color Fundus photographs," IEEE Transactons on Medical Imaging, vol. 29, no. 1, pp. 185–195, Jan. 2010.

[6] A. Osareh, B. Shadgar, and R. Markham, "A computational-intelligence-based approach for detection of Exudates in diabetic Retinopathy images," IEEE Transactions on Information Technology in Biomedicine, vol. 13, no. 4, pp. 535–545, Jul. 2009.

[7] A. P. Bhatkar and G. U. Kharat, "Detection of diabetic Retinopathy in retinal images using MLP Classifier," 2015 International Symposium on Nanoelectronic and Information Systems, Dec. 2015.

[8] R Priya and P. Aruna, "Diagnosis of diabetic retinopathy using machine learning techniques," ICTACT Journal on Soft Computing, vol. 03, no. 04, pp. 563–575, Jul. 2013.

[9] K. Saranya, B. Ramasubramanian, and S. Kaja Mohideen, "A novel approach for the detection of new vessels in the retinal images for screening diabetic Retinopathy," 2012 International Conference on Communication and Signal Processing, Apr. 2012.

[10] S. Mohammadian, A. Karsaz, and Y. M. Roshan, "A Comparative Analysis of Classification Algorithms in Diabetic Retinopathy," 29th International Conference on Software Engineering and Knowledge Engineering, Pittsrburgh, PA, US, Jul. 2017.

[11] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," NeuroImage, vol. 108, pp. 214–224, 2015.

[12] N. Tajbakhsh et al., " Convolutional Neural Networks for Medical Image Analysis: Fine Tuning or Full Training?," IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1299–1312, May. 2016.

[13] Y. Pan et al., " Brain Tumor Grading Based on Neural Networks and Convolutional Neural Networks," 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Aug. 2015.

[14] H.R. Roth et al. , " DeepOrgan: Multi-level Deep Convolutional Networks for Automated Pancreas Segmentation," Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015. Lecture Notes in Computer Science, vol 9349. Springer, Cham.

[15] S. Wang et al. , "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning," Journal of Neurocomputing, Vol. 149, pp. 708–717, Feb. 2015.

[16] H. Pratt et al. , "Convolutional Neural Networks for Diabetic Retinopathy," International Conference On Medical Imaging Understanding and Analysis, Jul. 2016.

[17] Holly H. Vo and Abhishek Verma, "New Deep Neural Nets for Fine-Grained Diabetic Retinopathy Recognition on Hybrid Color Space," 2016 IEEE International Symposium on Multimedia, Dec. 2016.

[18] [Online]. Available: https://github.com/fchollet/deep-learning-models. Accessed: Dec. 15, 2016.

[19] [Online]. Available: https://www.kaggle.com/diabetic retinopathy-detection. Accessed: Jul. 2015.

[20] [Online]. Available: https://kaggle2.blob.core.windows.net/forum-message-attachments/88655/2795/competitionreport.pdf. Accessed: Jul. 2015.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[23] [Online]. Available: http://cs231n.github.io/transfer-learning/

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Neural Information Processing Systems Conference (NIPS), 2012.

[25] D. Erhan et al., "The difficulty of training deep architectures and the effect of unsupervisedpre-training," in International Conference on artificial intelligence and statistics, 2009, pp. 153–160.

[26] C. Szegedy et al., "Going deeper with convolutions," IEEE Conference onComputer Vision and Pattern Recognition (CVPR), Jun. 2015.

[27] C. Szegedy et al., "Rethinking the Inception Architecture for Computer Vision," IEEE Conference on Computer Vision, Dec. 2015.

[28] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," IEEE Conference on Computer Vision, Oct. 2016.

[29] Z. Wang and J. Yang, "Diabetic Retinopathy Detection via Deep Convolutional Networks for Discriminative Localization and Visual Explanation", arXiv:1703.10757v2 [cs.CV], Apr. 2017.