ISSN: 2586-7652 (Print) Vol. 04, No. 02, September 2021

ISSN: 2635-7607 (Online)

International Journal of Advanced Engineering

Source: http://ictaes.org

Manuscript received: August 21, 2021 ; Revised: September 12, 2021 ; Accepted: September 24, 2021

Fine-tuning of ResNet Pre-trained Models for Diabetic Retinopathy Stages Classification

Bal Krishna Nyaupane[1], Rupesh Kumar Sah [2]

[1,2]Department of Electronics and Computer Engineering, Institute of Engineering, Tribhuvan University
bkn@wrc.edu.np[1], rupesh@wrc.edu.np[2]

Abstract

For the early treatment, Diabetic Retinopathy (DR)stages classification is a crucial step. DR is the complication of diabetes that damages the retina blood vessels caused by the high blood glucose level and different extent of microstructures. The major symptoms of DR are seeing an increasing number of floaters, eye pain or redness, having poor night vision, blurriness, sudden vision loss, etc. Diabetic Retinopathy has five stages of complication from Non-Proliferative Diabetic Retinopathy (NPDR) to proliferative Diabetic Retinopathy. In this paper, a state-of-art deep learning-based approach has been applied. ResNet architecture has recently been proven as the most advanced approach in image analysis. To detect the different stages of DR, the three pre-trained models ResNet50, ResNet101, and ResNet152 on Kaggle "Diabetic Retinopathy Detection" dataset have been applied. In this research, Gaussian filter and image rescale have been used as image preprocessing steps. Fine-tuning of ResNet pre-trained models with different hyperparameters is retrained to achieve the best result to classify the DR stages. Fine-tuning is conducted with the full frozen layer, partially frozen layer, and the full unfrozen layer of pre-trained models of ResNet. Training loss, training accuracy, test loss, and validation accuracy graphs have been used as major evaluation matrices to compare the results of three pre-trained models. Among three models, ResNet152 was found low test loss and maximum test accuracy.

Keywords: *Diabetic Retinopathy, ResNet, training, loss, test, accuracy*

1. Introduction

Diabetic retinopathy (DR) is a diabetes complexity that influences eyes and harms in the veins of the light-sensitive tissue at the retina. Diabetic Retinopathy is one of the reasons for the visual impairment among working-age grown-ups. Roughly four hundred million individuals worldwide have been analyzed with diabetes mellitus [1]. The comprehensiveness of this infection has multiplied in the previous 20 years particularly in Asia. Around thirty-three percent are required to be tested, a persistent eye sickness that can cause vision misfortune. Because of the investigation of the early treatment of Diabetic Retinopathy, DR can be characterized into five phases [2]. According to The National Eye Institute, the severity of DR patients can be categorized as follows:

Corresponding Author: *Bal Krishna Nyaupane*
Author's affiliation: *Institute of Engineering, Tribhuvan University*
Email: *bkn@wrc.edu.np*

Table -1: DR severity stages

Stage	Observable Findings	Severity
I	No abnormalities	No DR
II	Lesions of micro-aneurysms, small areas of balloon-like swelling in the retinas blood vessels.	Mild non-proliferative DR
III	Hard exudates or cotton wool spots swelling and distortion of blood vessels.	Moderate non-proliferative DR
IV	Many blood vessels are blocked, which causes abnormal growth factor secretion.	Severe non-proliferative DR
V	Growth factors induce proliferation of new blood vessels inside the surface of the retina, the new vessels are fragile and may leak or bleed, scar tissue from these can cause retinal detachment.	Proliferative DR

The DR severity stages, as mentioned in Table 1, stage I has no abnormalities; stage II has small areas of balloon-like swelling in the retinas blood vessels; stage III has some cotton wool spots swelling and distortion of blood vessels; stage IV has many blood vessels are blocked with abnormal growth factors; and the last stage has the proliferation of new blood vessels inside the surface of the retina, the new vessels are fragile and may leak or bleed, scar tissue from these can cause retinal detachment.

The number of people with diabetes rose from 108 million in 1980 to 422 million in 2014. More than 33% of the people that were living with diabetes in 2010 have given indications of Diabetic Retinopathy (DR), and 33% of those with indications of DR were influenced by vision compromising DR. In 2050, it is normal that the quantity of Americans 40 years or more seasoned with DR will significantly increase from 6.7 million in 2005 to 19.4 million; and among those 65 years old or more Americans, it will increment from 3 million to 11.8 million [2]. All people with diabetes mellitus will be in danger of creating DR. DR is one of the main sources of vision hindrance and visual impairment.

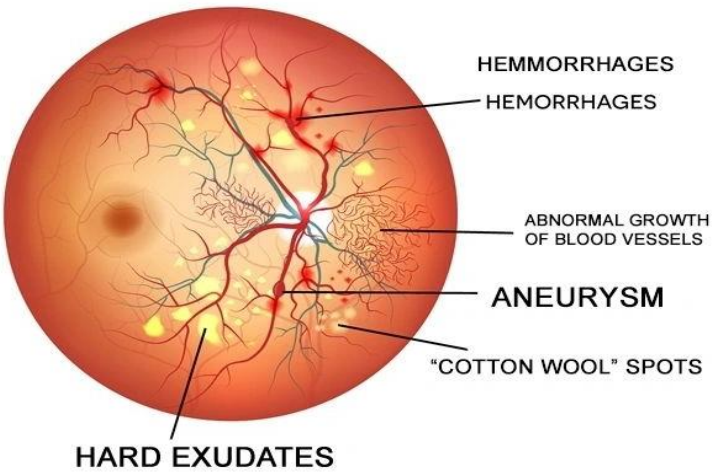


Figure 1. Diabetic Retinopathy Severity

Transfer Learning is the way toward taking a pre-trained neural network and adjusting the neural network to another diverse dataset by moving or repurposing the learned features. It is a short-cut process to re-use the model weights from pre-trained models that were developed for standard computer vision benchmark datasets, such as the ImageNet image recognition tasks. There are different approaches to use transfer learning. Each approach can be effective and save significant time in developing and training a deep convolutional neural network model. It may not be clear as to which the pre-trained model may yield the best results on new computer vision tasks, therefore some experimentation may be required [3].

Transfer learning approaches are classifier, train the entire model, train some Convolution layers and leave some frozen, and freeze all the Convolution Layer and Leave Full connected layer.

In the classifier approach, the pre-trained model is used directly to classify new images. There is no training required. In train the entire model approach, use the architecture of the entire pre-trained model and train it to achieve the desired accuracy. To train the entire pre-trained model, a large dataset is needed along with a huge amount of computational power. This approach might be expressive if a large dataset and GPU are not available.

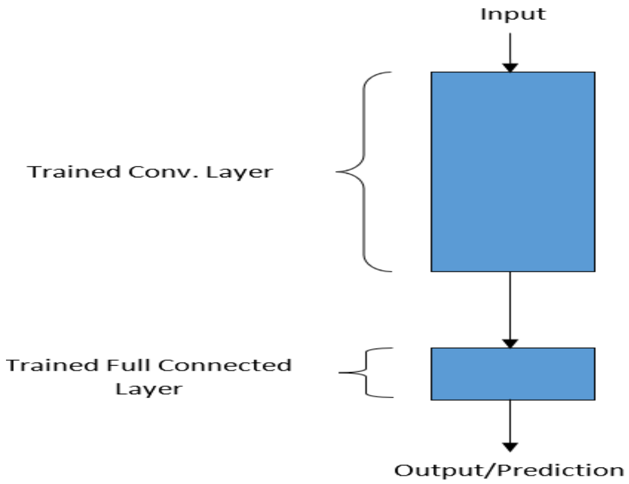


Figure 2. Approach -Train the entire model

In train some Convolution layers and leave some frozen approach, some higher convolution layers of the pre-trained model make trainable and the lower layer is left frozen. The fully connected layer can be added as per the requirement.

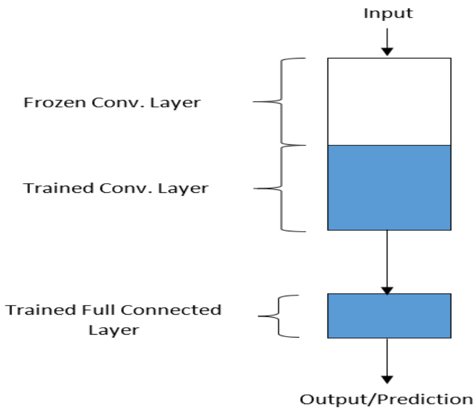


Figure 3. Approach - Train some Convolution layers and leave some frozen

In freeze all the Convolution Layer and Leave Full connected layer. approach, all the convolution layer of the pre-trained model makes trainable false and add a fully connected layer to train the model.

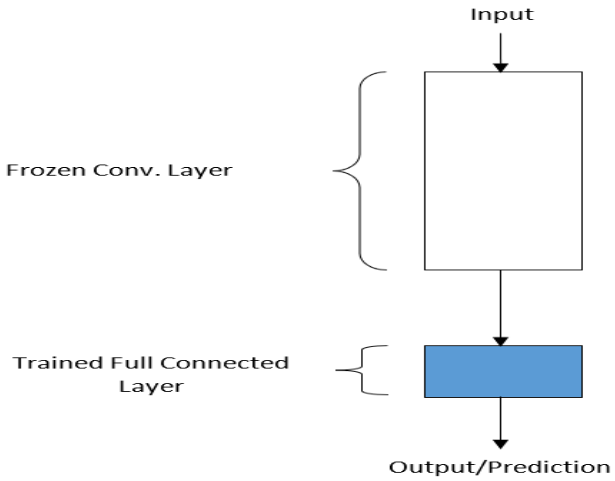


Figure 4. Approach - Freeze all the Convolution Layer and Leave Full connected layer

2. Methodology

1.1. Image Acquisition and Preprocessing

The dataset is taken from a Kaggle competition in this paper. Kaggle diabetic retinopathy dataset includes 1539 retina images. Each image is labeled by 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 main aim of this research is to classify the different stages of DR. The training and test images are rescaled in the range [0-1] from [0 – 255].

Table -2. Dataset Descriptions

Stages	Training Dataset	Test Dataset
No DR	400	100
Mild DR	300	100
Moderate DR	400	100
Severe DR	175	70
Proliferate DR	224	70
Total	**1499**	**440**

As mentioned in Table 2, the training dataset includes 400 images with no DR; 300 images with mild DR; 400 images with Moderate DR;175 images with Severe DR; and 224 images with Proliferate DR and the testing dataset includes 100 images with no DR; 100 images with mild DR; 100 images with Moderate DR;70 images with Severe DR; and 70 images with Proliferate DR.

1.2. ResNet Architecture

ResNet, also called Residual Network, was proposed in 2015 by researchers at Microsoft Research to solve the problem of vanishing/exploding gradient. Residual Network uses the skip connections to solve the vanishing/exploding gradient in deep neural networks. The idea of skip connection is to connect the input of a layer directly to the output of a layer after skipping a few [10].

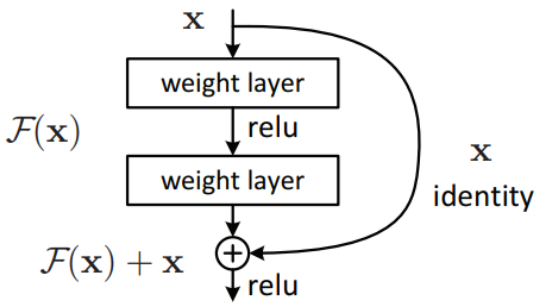


Figure 5. Skip Connections

In the above figure, we can see x is the input to $l+1^{th}$ layer which we are directly using to connect to a layer after skipping the identity connections and assume the output from $l+1^{th}$ layer is F(x). Then we can assume the output of the $l+2^{th}$ layer will be $F(x) + x$. This means that the value coming out from the activation function of the identity blocks is the same as the input from which we skipped the connections. It helps the connections by allowing the model to learn the identity functions which ensures that the higher layer will perform at least as good as the lower layer, and not worse [10]. The ResNet Network is composed of several blocks - consisting of a convolution + batch normalization + max-pooling operation. Every layer of a ResNet is composed of several blocks. When ResNet goes deeper, they normally do it by increasing the number of operations within a block. An operation refers to a convolution, a batch normalization, and a ReLU activation to an input, except the last operation of a block, that does not have the ReLU activation. A summary of the output size at every layer and the dimension of the convolutional kernels at every point in the structure is shown below.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 6 Sizes of outputs and convolutional kernels for ResNet Architecture

1.3. Fine-tuning

Fine-tuning is a process of unfreezing the entire model or unfreezing some part of a model or freezing all the layers of the model based on requirements. In addition to this, remove the full connected layer of architectures and add some full connected layers to match the number of classes and re-train model on the new dataset with a very low learning rate [4,5]. The low learning rate might achieve meaningful improvements, by incrementally adapting the pre-trained model weight to the new dataset. If randomly initialized trainable layers mix with trainable layers of the pre-trained model, the randomly initialized layers will cause very large gradient updates during training, which will destroy pre-trained model weights. Therefore, it is critical to use a very low learning rate at this stage, because training is typically a very small dataset [6]. As a result, it is the risk of overfitting very quickly if the model updates a large weight. Here, the model needs to readapt the pre-trained weights in an incremental way to prevent overfitting.

3. Experiment and Result

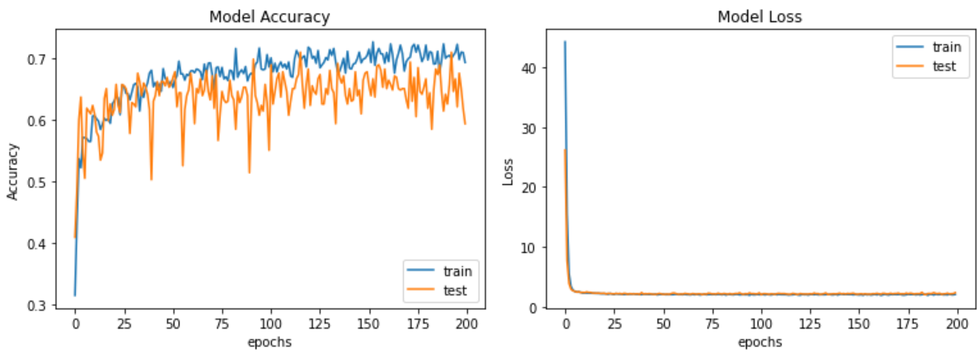
To conduct this research, Google Collaboratory has used to run code directly through a browser utilizing cloud computing. The three pre-trained models of ResNet architecture are trained one after another to complete this research. Google Collaboratory gives a decent GPU for free to run continuously for 12 hours. This research has used the GPU runtime equipped up to Tesla K80 with 12 GB of GDDR5 VRAM, intel Xeon Processor with two cores@ 2.20 GHz and 13 GB RAM, and available disk space of 69GB for 12 hours to perform research work.

Table 3 Hyperparameter and Fine-tunning

Architecture	Unfrozen Layer	Learning Rate	Batch Size [Training, Test]	Epoch	Fully Connected Layer	Regularization
ResNet152	Fully Frozen	0.00015	[64, 32]	200	[16, 5]	L1 and L2
	Fully Frozen	0.00075	[64, 32]	150	[32,16, 5]	L1 and L2
	Fully Frozen	0.000025	[32, 16]	250	[32,16, 5]	L1 and L2
	Partial frozen	0.00075	[32,16]	100	[16, 5]	L1 and L2
	All Unfrozen	0.00075	[32,16]	100	[16, 5]	L1 and L2
ResNet101	Fully Frozen	0.00015	[64, 32]	200	[16, 5]	L1 and L2
	Fully Frozen	0.00075	[64, 32]	150	[32,16, 5]	L1 and L2
	Fully Frozen	0.000025	[64, 32]	250	[32,16, 5]	L1 and L2
	Partial frozen	0.00075	[32,16]	100	[16, 5]	L1 and L2
	All Unfrozen	0.00075	[32,16]	100	[16, 5]	L1 and L2
ResNet50	Fully Frozen	0.00015	[64, 32]	200	[16, 5]	L1 and L2
	Fully Frozen	0.00075	[64, 32]	150	[32,16, 5]	L1 and L2
	Fully Frozen	0.000025	[64, 32]	250	[32,16, 5]	L1 and L2
	Partial frozen	0.00075	[32,16]	100	[16, 5]	L1 and L2
	All Unfrozen	0.00075	[32,16]	100	[16, 5]	L1 and L2

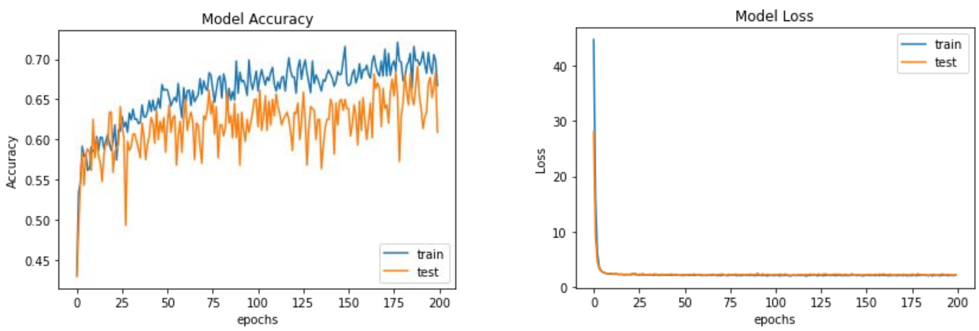
In this research, fine-tuning of ResNet architectures has been done in three phases- first frozen all the layers; second frozen partial (fifty present layers of the architecture) layer of the architecture; and unfrozen all the layers of the architectures. The hyperparameters used during the fine-tuning process are presented in above table 3. To evaluate the performance of the different combinations of hypermeters for fine-tuning process, this research has taken training loss, test loss, training accuracy, and validation accuracy plots. We have carried out different values of hyperparameter with the fine-tuning process. In total, fifteen combinations of fine-tuning and hyperparameters are conducted. The fully unfrozen layer of all three ResNet models completely failed to learn, so model accuracy and loss plot has not been presented in this report. Among them, only nine combinations of results are presented below from Figure 7 to Figure 15.

Figures 7, 8, and 9 show the model accuracy and model loss plot of the fully frozen layer of ResNet50, ResNet101, and ResNet152 architectures with learning rate=0.00015, batch-size=64, and L1 & L2 regularization at fully connected layer to prevent overfitting. Even though the ResNet152 model slightly suffers from overfitting than others, it has low variance. All three-model model loss plots remain same up to 200 epochs.



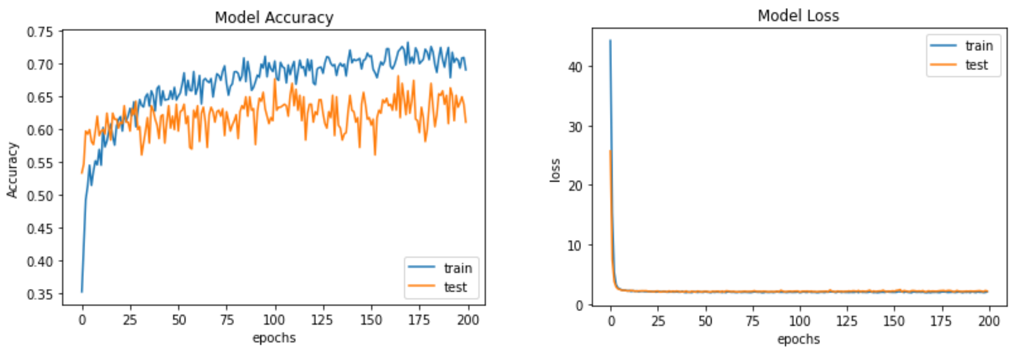
(a) (b)

Figure 7. Full frozen layer of ResNet50 architecture with different hypermeters.



(a) (b)

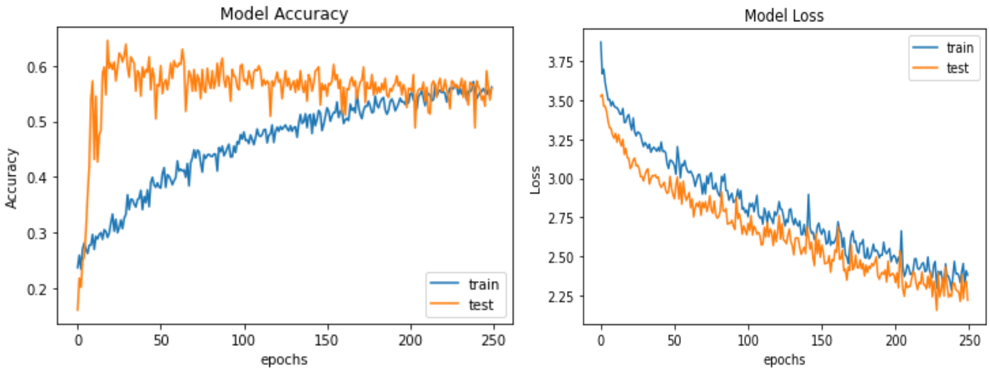
Figure 8. Full frozen layer of ResNet101 architecture with different hypermeters.



(a) (b)

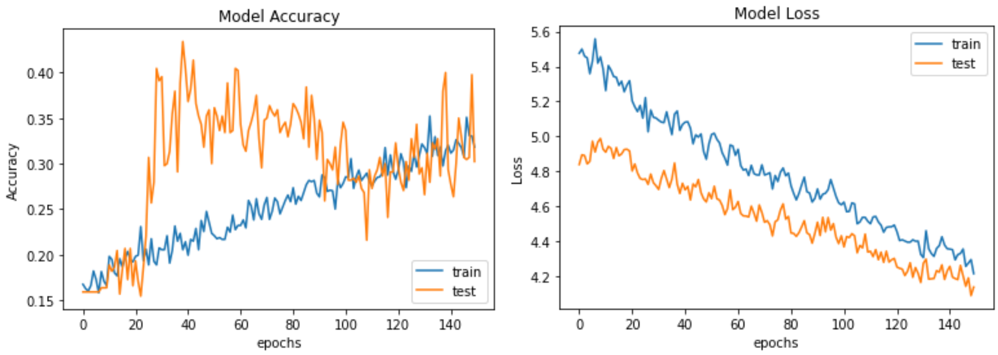
Figure 9. Full frozen layer of ResNet152 architecture with different hypermeters.

Figures 10, 11, and 12 show the model accuracy and model loss plot of fully frozen layer of ResNet50, ResNet101, and ResNet152 architectures with learning rate=0.000025, batch-size=32, dropout=0.5 and L1 & L2 regularization at fully connected layer to prevent overfitting. Even though the ResNet101 model has a good model loss plot, it failed to learn. The ResNet50 model training accuracy started to decrease from 100 epochs, even if the model loss plot is decreasing. The Resnet152 model test accuracy gradually improved up to 250 epochs and reached at sixty-five percent of test accuracy and it has a good model loss plot.



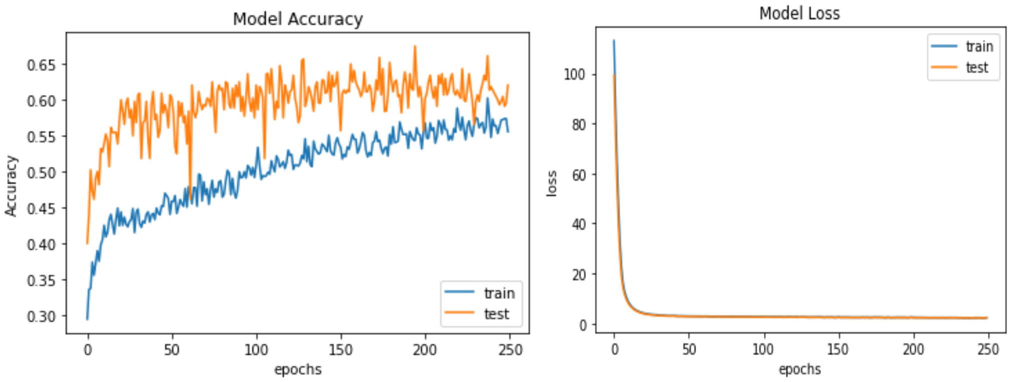
(a) (b)

Figure 10. Full frozen layer of ResNet152 Architecture with different hypermeters



(a) (b)

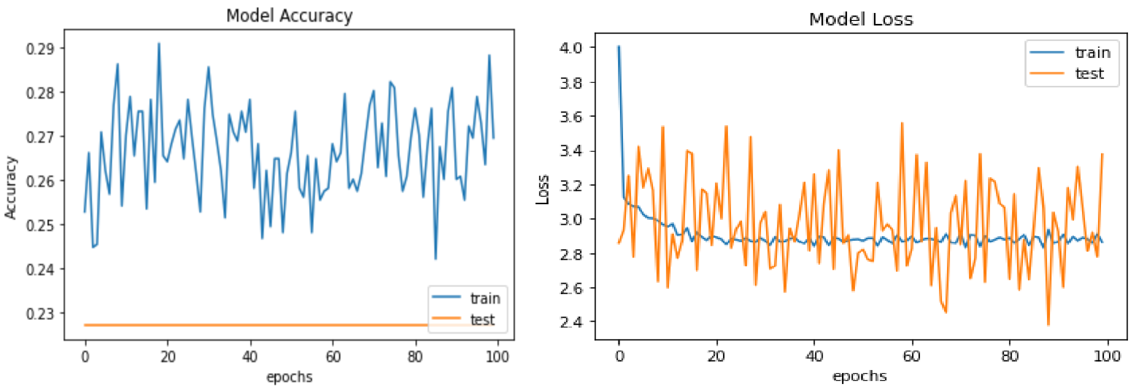
Figure 11. Full frozen layer of ResNet110 Architecture with different hypermeters



(a) (b)

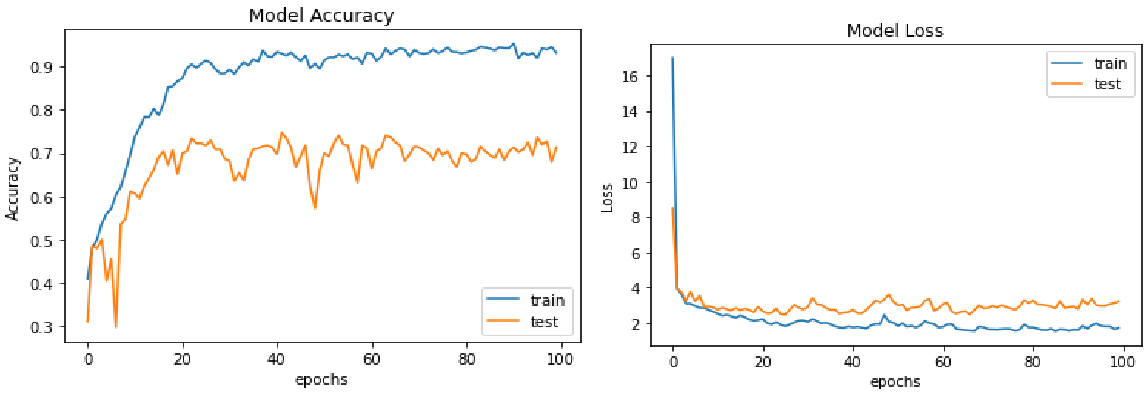
Figure 10: Full frozen layer of ResNet152 Architecture with different hypermeters

Figures 13, 14, and 15 show the model accuracy and model loss plot of partial frozen layer of ResNet50, ResNet101, and ResNet152 architectures with learning rate=0.00075, batch-size=32, dropout=0.5 and L1 & L2 regularization at fully connected layer to prevent overfitting. The ResNet50 model completely failed to learn when layers 1 to 155 are frozen and 156 to 234 layers remain unfrozen. The ResNet101 model has reached the maximum of seventy percent test accuracy when layers 1 to 155 are frozen and 156 to 344 layers remain unfrozen; however, it suffers from overfitting with the increasing gap between the training loss and test loss as well as training accuracy and test accuracy. The ResNet152 model started to overfit from 20 epochs when layers 1 to 455 are frozen and 456 to 520 layers remain unfrozen.



(a) (b)

Figure 13. Partial Frozen layer of ResNet50 Architecture with different hypermeters



(a) (b)

Figure 14. Partial Frozen layer of ResNet101 Architecture with different hypermeters

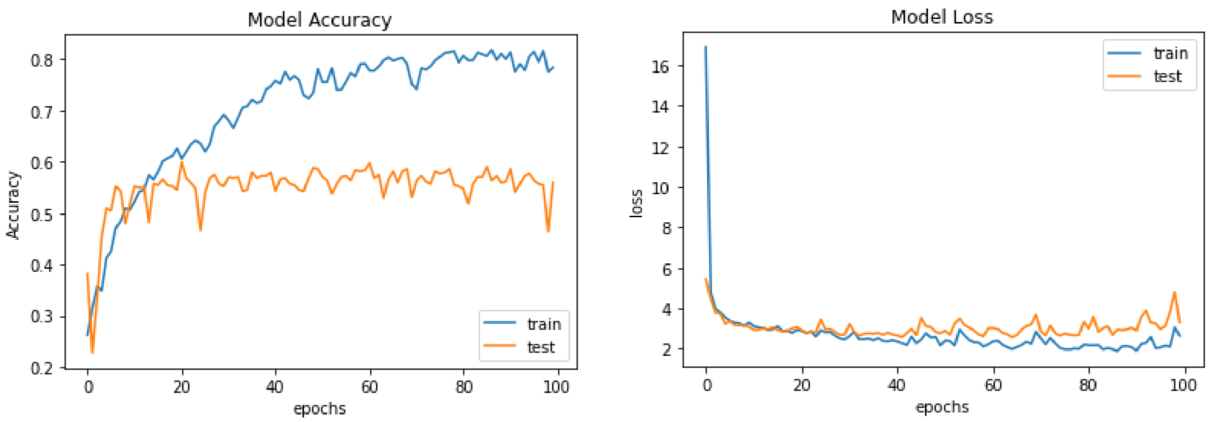


Figure 15. Partial Frozen layer of ResNet152 Architecture with different hypermeters

4. Conclusion

In this research, pre-trained models of the ResNet architectures (ResNet50, ResNet101, and ResNet152) have been implemented to classify Diabetic retinopathy (DR) of five stages (no DR, Mild DR, Moderate DR, Severe DR, and Proliferative DR). This research was conducted with fine-tuning of pre-trained ResNet models for different hyperparameters. In this research three value of learning rate was used to train the models that are 0.00015,0.00075, and 0.000025.

When all the layers of pre-trained models were frozen, all three models worked well when the learning rate is 0.00015 and batch size is 64. Among them, the ResNet152 model worked better than rest two models. It is found that if all layers make trainable, then all three pre-trainable models do not work perfectly to classify the DR stages. Even though there was a good loss plot for three models for all layers make trainable, they did not work well for training and validation. When some layers make frozen the ResNet50 model complete failure for training and testing, whereas the ResNet101 model suffers from overfitting from epochs 50 with training accuracy in between 80 and 90 percent and testing accuracy in between60 and 70 percent. And the ResNet152 model also suffers from overfitting from epochs 20 with training accuracy in between 70 and 80 percent and testing accuracy in between 50 and 60 percent. Hence, this research was conducted to build ResNet architectures models that perform well on unseen DR datasets by efficiently learning from the small dataset because there is a limited dataset in Diabetic retinopathy. Our model has reached an accuracy that is higher than other techniques that have been used for ResNet architectures on the Kaggle Diabetic retinopathy challenge dataset.

References

[1] Xiaoliang Wang; Yongjin Lu; Yujuan Wang; Wei-Bang Chen,"Diabetic Retinopathy Stage Classification Using Convolutional Neural Networks", 2018 IEEE International Conference on Information Reuse and Integration (IRI),Year: 2018

[2] National Diabetes Statistics Report, Centers for Disease Control and Prevention (CDC), US Dept. of HHS, "Estimates of Diabetes and its Burden in the United States," pp. 1-20, 2017

[3] A. Rakhlin, "Diabetic Retinopathy detection through integration of deep learning classification framework," bioRXiv, pp. 1-11, Feb. 2017.

[4] Narayana Bhagirath Thota and Doshna Umma Reddy "Improving the Accuracy of Diabetic Retinopathy Severity Classification with Transfer Learning" 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)

[5] S. Mohammadian, A. Karsaz and Y. M. Roshan, "Comparative Study of Fine-Tuning of Pre-Trained Convolutional Neural Networks for Diabetic Retinopathy Screening," 2017 24th National and 2nd International Iranian Conference on Biomedical Engineering (ICBME), Tehran, 2017, pp. 1-6.

[6] Shankar, K.; Zhang, Y.; Liu, Y.; Wu, L.; Chen, C.H. Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification.IEEE Access 2020, 8, 118164–118173

[7] Shu, D.; Ting, W.; Cheung, C.Y.L.; Lim, G.; Siew, G.; Tan, W.; Quang, N.D.; Gan, A.; Hamzah, H.; Garcia-franco, R.; et al. Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images from Multiethnic Populations with Diabetes. JAMA 2017, 318, 2211–2223

[8] S. Burewar, A. B. Gonde and S. K. Vipparthi, “Diabetic Retinopathy Detection by Retinal segmentation with Region merging using CNN,” 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS), Rupnagar, India, 2018, pp. 136-142.

[9] Xiaoliang Wang; Yongjin Lu; Yujuan Wang; Wei-Bang Chen,” Diabetic Retinopathy Stage Classification Using Convolutional Neural Networks”, 2018 IEEE International Conference on Information Reuse and Integration (IRI), Year: 2018M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[10] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. arXiv 2015, arXiv:1512.03385.