International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Comparative Analysis of Hyperparameter Tuned Convolutional Neural Networks for classification of Diabetic Retinopathy

[1]Rupa Patel [2]Dr. Anita Chaware

Submitted: 25/04/2023 **Revised:** 26/06/2023 **Accepted:** 05/07/2023

Abstract Diabetic Retinopathy (DR) is one of the leading cause for loss of vision among diabetic patients. Early detection and treatment of disease can prevent serious complications. Deep learning in medical image analysis is very effective. One of the important aspects in building efficient machine learning algorithm is to choose the right combination of hyper parameters. To understand the sensitivity of hyper parameter, hyperparameter tuning is applied to ten deep convolutional neural network (DCNN). The fundus images from Kaggle diabetic retinopathy dataset is pre-processed and resized to 224 x 224x3. Depending upon the severity of disease dataset images belongs to one of the class (0, 1, 2, 3, 4) so performance of all the ten networks are evaluated class wise. Experimental results reveal that Vgg16 outperformed InceptionV3, Xception, ResNet50, DenseNet121, Densenet169 and DenseNet201. MobileNetv2 outperformed other two light weight models MobileNetv1 and NASNetMobile. Training accuracy for vgg16 is 92%, validation accuracy is 85%, sensitivity is 84%, specificity is 96% and f1 score is 0.84. For MobileNetv2 training accuracy is 98%, validation accuracy is 81%, sensitivity is 81%, specificity is 94% and f1 score is 0.81. Training of Vgg16 and MobileNetv2 is carried out using Adam optimizer with learning rate of 0.001 and 0.00002 respectively, dropout 0.5, batch size is set to 32 and no. of Epoch to 40 and 20 respectively. While comparing the proposed work with previously related similar work it is found that our approach yields better results. This work contributes towards the improvement of the image classification techniques.

Keywords: *Hyperparameters tuning, Machine learning, Diabetic Retinopathy, Deep Convolutional Neural network*

1 Introduction

Diabetics is one of the fasted growing disease worldwide and by the year 2030 it will reach 454 million [1]. Diabetic retinopathy is caused by increase in blood sugar level due to diabetes. Diabetic retinopathy can be classified into Non Proliferative (NPDR) and Proliferative (PDR) [2]. Early stage of diabetic retinopathy is Non Proliferative DR, in which blood vessels within retina starts to discharge fluid or blood causing retina to swell. In due course of time vision gets blurred. Advanced stage of DR is proliferative DR where in there is growth of new blood vessels that can bleed into vitreous cavity which ultimately cause vision loss [2].Early diagnosis and treatment of DR is important to avoid loss of vision. This requires regular visit to ophthalmologist for screening of diabetic eye and grading it according to severity of disease. To fight the disease, in the form of treating it or diagnosing it at an early stage, researchers are using artificial intelligence in assisting people with diabetes. Emphasis is on finding automatic computational mechanism that can check the severity of diabetic retinopathy.

Deep Convolution Neural network (DCNN) has become an effective approach in medical image analysis. DCNN is a multilayer neural network that is trained to extract the features of dataset. Deeper the efficient DCNN is, the more detailed features are extracted [3]. This helps DCNN to classify images more correctly. VGG16 [4], Resnet50 [5], InceptionV3 [6], Xception [7], variants of DenseNet (DenseNet 121 ,DenseNet169, DenseNet201) [8 ,9] are the architectures that focuses on deep learning. Lightweight deep learning models MobileNetV1 [10], MobilenetV2[11] and NASNetMobile[12] are suitable for resource constrained devices. Researchers proposed these architectures for deep feature extraction and for classification of images.

The performance of the neural network depends on the underlying architecture used for specific task domain. Hyperparameter tuning plays an important role in improving the performance of DCNN [13] on a given learning task. It is the process of obtaining the right combination of hyperparameter by running several trials in a single training process. Once the process gets complete an optimal set of hyperparameters are obtained, ultimately this improves the performance of DCNN.

The focus of this study is to identify the important hyperparameters that influence the training of the network and the accuracy of the networks. The remainder of this paper is organized as follows. In section 2 Neural Network hyper parameters are discussed. Section 3 provides an overview of

1Research Scholar, P.G. Department of Computer Science. S.N.D.T.W.U, Asst. Professor, Department of Information Technology, Anna Leela College of Commerce and Economics, Mumbai, India
Orcid id: 0009-0004-2187-7554
rupalearn1@gmail.com
2P.G. Department of Computer Science , S.N.D.T. Women's University, Juhu Campus, Mumbai, India
Anita.Chaware@computersc.sndt.ac.in

related work .In section 4 implementation of proposed work and performance matrices used to evaluate the network are discussed. In section 5 results and discussion of the hyperparameters tuning experiments are presented. Paper is concluded in section 6.

2 Neural Network Hyperparameters

Artificial Neural network is a collection of neurons arranged in layers where each neuron performs some computation [14]. Mathematically it represent relationship between data. The formula is

$$y = w^T x + b \tag{1}$$

Where y represent numeric value that network should learn to predict, x is a vector representing data features. Each connection to the neuron has a weight w. This value determines how much influence the input will have on the output. Constant b is a special kind of input data which influence the output of neuron. Weight and bias are model parameter. They are learnable parameters since their values gets updated during data learning process of the network.

Another set of network parameters are hyperparameters. They are often referred to as configurable parameters. Their values cannot be directly estimated from data learning process but must be set before training a machine model. Proper selection of hyperparameters plays a crucial role in building an efficient model. It controls the structure and complexity of the network [15]. Hyperparameters can be categorized on the basis of role they play. The parameters of the neural networks are depicted in Fig. 1.

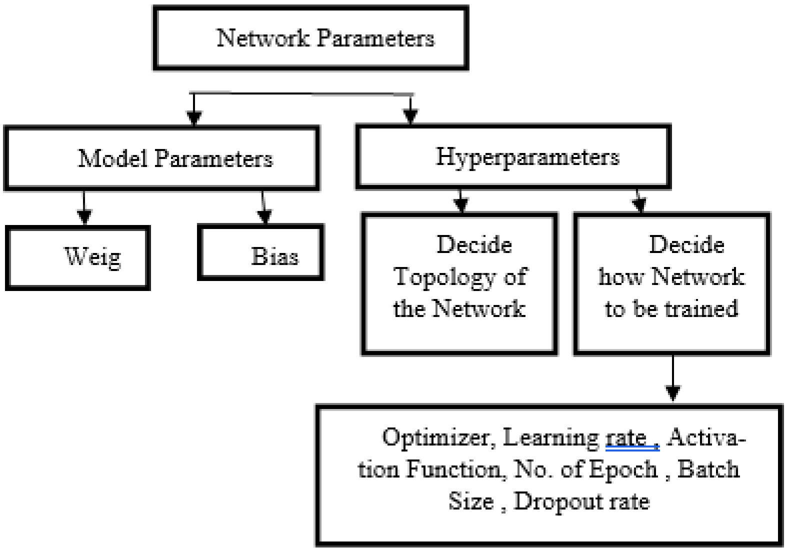

Fig. 1. Neural Network Parameters

First category determine the topology of the network and another set comprises of the hyper parameters that determines the way the network can be trained, Variables that decides the structure of the network [16] are number of layers in a network, number of filters used to extract the features from the image data, size of filter, padding and stride. Another category is variables that decides how networks are trained [16-18]. This paper focus on second category. The second category hyperparameters are shown in Fig. 2.

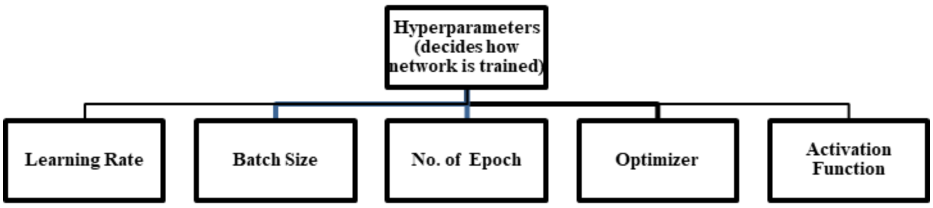

Fig. 2. Hyperparameters that decides how networks are to be trained

2.1 Learning Rate

Learning rate is often referred to as step size at each iteration. It is used to update the value of learnable parameter weight. Weights are modified to reduce error of estimated function. A configurable parameter learning rate is specified below 1.0. If learning rate is 0.2 then weights are updated with 0.2 * estimated weights. If it is too small then it will take longer time for model to train. If it is too large it will fail to converge and so selecting the good learning rate is important. To find the appropriated value of learning rate, learning rate scheduler is often used. Two techniques commonly used are constant learning rate and learning rate decay. In constant learning rate, learning rate doesn't change during the training while in learning rate decay, a certain value is selected as initial learning rate and gradually decreases [16 ,18].

2.2 Batch Size

One of the important step while training the network is to prepare the input dataset properly so that required features can be extracted. The dataset is divided into several small group often referred as batch, as entire dataset cannot be passed at once. Batch size determines number of training samples used during one iteration [19]. Iteration refers to the no. of batches needed to complete one epoch. If a dataset contains 992 images, it can be divided into batches of 32 then it will take 31 iterations to complete one epoch. Training and validation accuracy of the network depends on batch size. So choosing the batch size depends upon the dataset also. Generally batch size could be 32, 64, 128, or 256 [18,19].

2.3 Number of Epoch

One epoch means complete dataset is passed through the network. Since passing the entire dataset once is not enough, dataset is to be passed multiple time to same network. Number of epoch indicates how many times a model /algorithm will passes through entire dataset [19]. Identifying the correct value for number of epoch is related with the diversity of the dataset.

2.4 Optimizer

Optimizers play a significant role in improving the accuracy of the network. Hyperparameters related to optimizers is choice of optimizer, min-batch size, beta and momentum [18]. Optimizers can either belong to gradient descent optimizer or adaptive optimizer family. Adam and SGD optimizer is considered for the study. Stochastic gradient descent (SGD) update the parameters for each sample in the dataset. Adam, an adaptive optimizer. It is combination of SGD and RmSprop [18].

2.5 Activation Function

They generally decide whether neurons should be activated or not. Activation function can be linear or nonlinear. Linear function can produce any output while nonlinear function produce output in the range (0, 1) or (-1, 1). In this study Relu and softmax activation functions are used for classification task. The Relu (rectified linear unit) is a nonlinear function shown by equation 2.

$$f(x) = \max(0, x), f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2)$$

Softmax function is generally used in the output layer for multi class classification task. Generally used in the output layer of network. It calculates multinomial probability distribution function of each target class over all possible target classes. [19]. Softmax function is given by equation 3.

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad (3)$$

Where z_i is input vector, e^{z_i} is standard exponential function for the vector, k is the number of classes in case of multiclass classification and e^{z_j} standard exponential function for output vector.

There is a direct impact of hyperparameters selection on machine learning performance. Optimal setting of hyperparameters is data dependent. There are many search approaches for tuning the hyperparameters to improve the performance of network. In manual approach the hyperparameters are chosen manually using trial and error approach. These hyperparameters are then tuned manually. The automatic methods involves grid search [20], random search [21], Bayesian optimization [22], evolutionary optimization [23].In grid search, a grid is created which contains all the possible values of the selected hyper-parameter. Grid search first collects all possible parameter combinations, and then tries to find the best combination. Random search selects random parameter combinations and finds the best combination to train network [19]. Bayesian optimization objective is to find the global maximum or minimum of an unknown function [24].To make an assumption about unobserved parameters, Bayesian uses a set of previously evaluated parameters and resulting accuracies [19].

3 Related Work

Several researchers have attempted to classify the retinal fundus image. Some of them performed binary classification. They have classified DR dataset into two classes DR or NO DR (normal images).Others performed multi-class classification. DR images were classified into five classes. This section review the studies conducted to classify the diabetic retinopathy dataset into No DR, Mild, moderate, severe & proliferative classes.

H.Prattet al [25] detected the severity of diabetic retinopathy of Kaggle dataset using their CNN architecture. They used ten Convolutional layer, 8 max-pooling layer and three fully connected layers. They have used dropout methods, L2 regularization and softmax function. This network achieves the accuracy of 75%.

The performance of the three pretrained network vGG16 ,Inception V3 and ResNet50 was investigate in Ref.[26] to analyzed the levels of Diabetic Retinopathy in the Kaggle dataset. The networks were fine-tuned with no. of epoch (100), batch size (32), learning rate and momentum coefficient hyperparameters.

Vimukthi et al. [27] suggested a light weight CNN model with six convolution layer to classify images from Kaggle and Messidor dataset into five severity level. Augmentation, hyperparameter tuning and regularization technique were applied to the model. They have compared their model with

VGG16, ResNet50, Inception V3 and Xception. Best accuracy for the proposed architecture was achieved with number of units in second layer was 256, dropout rate 0.1 and SGD optimizer.

S. Mohammadian et al. [28] studied the performance of two pretrained architecture InceptionV3 and Xception. They have used activation function (RELU & ELU) and optimizer (Adam & SGD) hyperparameter for comparative study.

4 Implementation Details

In this section we give an overview of the approach, datasets and algorithms used for study, hyperparameter settings and performance metrics.

4.1 Approach

Fig. 3 depicts the approach of the study. The images of the dataset are pre-processed and then split into training and validation sets with the ratio of 80:20 respectively. Hyperparameter tuning of CNN architecture is performed. Once the networks are trained the classification results of the networks are evaluated. The details of each step are described in subsections.

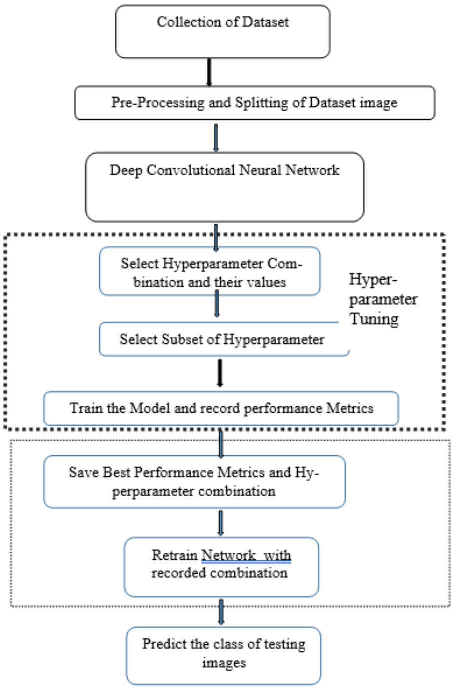

Fig. 3. Flowchart of the approach

4.2 Dataset

In medical domain privacy and security is important. For experimental purpose dataset has been taken from Kaggle - APTOS 2019 blindness detection [29]. Based on the severity level of diabetic retinopathy, each image is rated on a scale of 0 to 4: The number of fundus images belonging to a particular class and the meaning of their severity level is summarized in Table 1.

Table 1. Number of images of each severity level

Class	Types of DR	Meaning [30]	Number of images in Dataset [31]
0	No DR	No abnormality	1805
1	Mild nonproliferative	Only Microneurysms (small swelling in the retina's tiny blood vessels)	370
2	Moderate nonproliferative	Macular edema (fluid buildup in the center area of the retina)	999
3	Severe nonprolifera-	Walls of blood vessels in retina	193

	tive	gets weaken,new blood vessesl not growing	
4	Proliferative DR	New blood vessels start to grow on the surface of the retina. These are fragile and can leak blood or fluid.	295

Out of 3662 images, 80 percent (2929 images) of the dataset are allocated to the training set while 20 percent (733 images) are allocated to validation set. For all the classifiers same training and validation datasets were selected. The fundus images from dataset is pre-processed and resized to 224 x 224x3.

4.3 Deep learning network

We developed ten fine-tuned convolution neural network architectures to classify diabetic retinopathy images. These architecture are VGG16, Inceptionv3, Xception, ResNet50, DenseNet121, DenseNet 169, DenseNet201, MobileNetV1, MobilenetV2, and NASNetMobile. MobileNetV1, MobilenetV2 and NASNetMobile are light weight models suitable for resource constrained devices such as mobile.

The input size of the image is 224 x 224 x3, accordingly the input layer of all the pretrained CNN were considered. Since the number of classes are five the dimension of last fully connected dense layer was set to five as depicted in Table 2[31 , 32].

Table 2. Deep Neural Networks

Classifier	**No. of Layers**	**No. of Parameters**	**Input Layer Size**	**Output Layer Size**
VGG16	16	138.4 M	224 x 224 x 3	5 by 1
Inception V3	189	23.9 M	224 x 224 x 3	5 by 1
Xception	81	22.9 M	224 x 224 x 3	5 by 1
ResNet50	107	25.6 M	224 x 224 x 3	5 by 1
DEnseNet121	242	8.1 M	224 x 224 x 3	5 by 1
DenseNet169	338	14.3 M	224 x 224 x 3	5 by 1
DenseNet201	402	20.2 M	224 x 224 x 3	5 by 1
MobileNetv1	55	4.3 M	224 x 224 x 3	5 by 1
Mobilenetv2	105	3.5 M	224 x 224 x 3	5 by 1
NASNetMobile	389	5.3 M	224 x 224 x 3	5 by 1

4.4 Hyper parameter tuning

Fine-tuned VGG16, Inception V3**,** Xception**,** ResNet50, MobileNetv1, Mobilenetv2, DenseNet121, DenseNet169, DenseNet201 and NASNetMobile networks are used to classify the DR images.

Hyper parameters that influence the training of the classifiers are manually selected as shown in table 3 to improve the performance of the network. Nnumerous experiments were performed before finalizing the set of hyperparameters for every pre-trained model.

Hyperparameters considered in this study are Optimizers (Adam, SGD) , learning rate (0.001, 0.00001, and 0.00002) , dropout rate (0.2 ,0.5) , batch Size (8,16,32) and epoch (10,15,20,30,40) .

Table 3. Hyperparameters and their values

	Hyper Parameters

Classifier	Optimizer	Learning Rate	Dropout rate	No. of Epoch	Batch Size
VGG16	Adam	0.001	0.5	40	32
Inception V3	Adam, SGD	0.001	0.5	10	32
Xception	Adam	0.001	0.5	10	32
ResNet50	Adam	0.00002	0.5	15	32
DenseNet121	Adam	0.001	0.2	10	32
DenseNet169	Adam	0.001	0.2	30	32
DenseNet201	Adam	0.001	0.2	20	32
MobileNetv1	Adam	0.00002	0.5	30	32
Mobilenetv2	Adam	0.00002	0.5	20	32
NASNetMobile	Adam	0.00001	0.5	30	32

4.5 Performance Matrices

To evaluate the performance of the classifier the metrics used are accuracy, recall, precision, and f1 score parameters [26 , 33].

5 Experimental Results and Discussion

Large networks have higher speedups on GPU. It also shows better programmability for computations involving small batches [34]. During program execution, convolution operations are computationally costly so to speed up DCNNs, implementation of a DCNN using GPUs is done [35].

To evaluate the neural network performance different metrics mentioned in section 4.5 is considered. For each network and for each class, performance metrics are calculated.

Table 4. Label /Class Validation accuracy

Models	Class 0	Class 1	Class 2	Class 3	Class 4
VGG16	0.99	0.65	0.89	0.36	0.43
Inception V3	0.96	0.45	0.56	0.16	0.28
Xception	0.98	0.46	0.92	0.03	0.32
ResNet50	0.99	0.44	0.82	0.06	0.13
DenseNet121	0.98	0.44	0.87	0.17	0.43
DenseNet169	0.96	0.52	0.83	0.12	0.15
DenseNet201	0.98	0.46	0.65	0.10	0.57
MobileNetv1	0.99	0.44	0.74	0.10	0.25
Mobilenetv2	0.99	0.46	0.76	0.19	0.45
NASNet Mobile	0.98	0.45	0.74	0.03	0.20

Accuracy (equation 4) is an important evaluation metric. Class wise accuracy for each of the class in dataset is computed and depicted in Table 4.Vgg16 outperforms in all the 5 classes, Xception, Densenet121 and Mobilenetv2 in four out of five cases.

Precision (equation 7) and recall (equation 5) are other two evaluation metrics that are widely used. Precision gives us a clear picture about what percentage of predicted positives is actually true positive whereas as recall tells us about what fraction of actual positives is correctly classified. The computed precision and recall of tuned model are depicted in Table 5 and Table 6.

Table 5. Performance Metric: CLASS SENSITIVITY /RECALL

Models	Class 0	Class 1	Class 2	Class 3	Class 4
VGG16	0.99	0.65	0.89	0.36	0.43
Inception V3	0.96	0.45	0.56	0.16	0.28
Xception	0.98	0.46	0.92	0.03	0.32
ResNet50	0.99	0.44	0.82	0.06	0.13
DenseNet121	0.98	0.44	0.87	0.17	0.43
DenseNet169	0.96	0.52	0.83	0.12	0.15
DenseNet201	0.98	0.46	0.65	0.10	0.57
MobileNetv1	0.99	0.44	0.74	0.10	0.25
Mobilenetv2	0.99	0.56	0.86	0.17	0.45
NASNet Mobile	0.98	0.45	0.74	0.03	0.20

Table 6. Performance Metric: PRECISION

Models	Class 0	Class 1	Class 2	Class 3	Class 4
VGG16	0.85	0.81	0.84	0.81	0.90
Inception V3	0.85	0.52	0.65	0.18	0.26
Xception	0.94	0.89	0.66	0.99	0.79
ResNet50	0.82	0.79	0.7	0.62	0.62
DenseNet121	0.92	0.63	0.70	0.50	0.76
DenseNet169	0.88	0.61	0.65	0.71	0.64
DenseNet201	0.82	0.73	0.75	0.10	0.42
MobileNetv1	0.79	0.72	0.70	0.67	0.71
Mobilenetv2	0.84	0.72	0.74	0.44	0.68
NASNet Mobile	0.79	0.67	0.71	0.25	0.57

Sensitivity /recall are the test that correctly identify patients with DR and No DR. Vgg16 turned out to be good performer in doing this while Mobilenetv2 is the next good performing model.

The specificity (equation 6) correctly identify negative cases. In highly sensitive domain it is more concerned for minimal wrong positive diagnosis. Hence high specificity is one of the major concerned. As observed from Table 7 for class 3 (Severe nonproliferative) and for class 4(Proliferative DR) all the networks have high specificity,

Table 7. Performance Metric: CLASS SPECIFICITY

Models	Class 0	Class 1	Class 2	Class 3	Class 4
VGG16	0.83	0.98	0.93	0.99	0.99
Inception V3	0.86	0.95	0.87	0.96	0.93

Xception	0.94	0.98	0.82	0.99	0.99
ResNet50	0.79	0.99	0.86	0.99	0.99
DenseNet121	0.90	0.98	0.89	0.99	0.90
DenseNet169	0.87	0.96	0.83	0.99	0.99
DenseNet201	0.79	0.98	0.92	0.99	0.93
MobileNetv1	0.75	0.98	0.88	0.99	0.99
Mobilenetv2	0.82	0.98	0.90	0.99	0.98
NASNet Mobile	0.98	0.45	0.74	0.03	0.20

Another evaluation metric F1-score (equation 8) is used. For class 0 all the networks has highest values in the range 0.88 to 0.96 as depicted in table 8. Class 2 has second highest whereas class 3 has the lowest F1-score for all the models.

Table 8. Performance Metric: F1 SCORE

Models	Class 0	Class 1	Class 2	Class 3	Class 4
VGG16	0.92	0.72	0.86	0.50	0.58
Inception V3	0.91	0.48	0.60	0.17	0.27
Xception	0.96	0.55	0.77	0.05	0.45
ResNet50	0.89	0.57	0.75	0.1	0.21
DenseNet121	0.95	0.52	0.77	0.26	0.55
DenseNet169	0.92	0.56	0.73	0.21	0.24
DenseNet201	0.89	0.57	0.70	0.10	0.49
MobileNetv1	0.88	0.54	0.72	0.17	0.37
Mobilenetv2	0.91	0.56	0.80	0.25	0.54
NASNet Mobile	0.87	0.54	0.73	0.05	0.30

Vgg16 is and MobileNetv2 are very good at classifying images under class label 0 and is good for class label 2.

Classification results of ten models were compared and is summarized in Table 9. And depicted in figure 4.

Table 9. Models Validation statistics (Accuracy, Sensitivity, Specificity, F1-score)

Models	Training Accuracy	Validation Accuracy	Sensitivity/ recall	Specificity	F1-Score
VGG16	0.92	0.85	0.84	0.96	0.841
Inception V3	0.84	0.74	0.69	0.92	0.72
Xception	0.98	0.81	0.81	0.95	0.81
ResNet50	0.93	0.811	0.77	0.94	0.77
DenseNet121	0.85	0.81	0.84	0.96	0.84
DenseNet169	0.84	0.79	0.77	0.94	0.77
DenseNet201	0.92	0.75	0.75	0.93	0.75
MobileNetv1	0.99	0.81	0.75	0.93	0.75

MobileNetv2	0.98	0.83	0.84	0.96	0.84
NASNetMobile	0.90	0.80	0.75	0.93	0.75

It is clear from Table 9 that in terms of overall accuracy measure among heavy weight model Vgg16 and among light weight model MobileNetv2 model generates the most accurate predicted class label in the validation set.

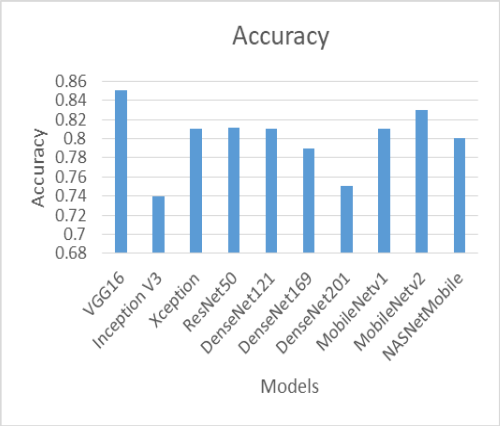

Fig. a. Accuracy

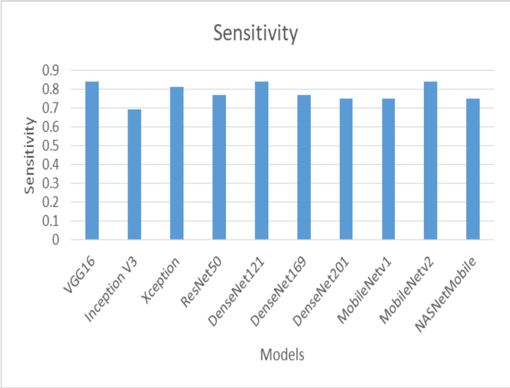

Fig. b. Sensitivity

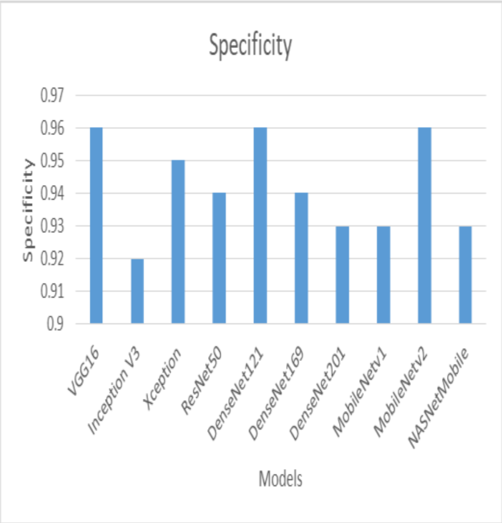

Fig. c. Specificity

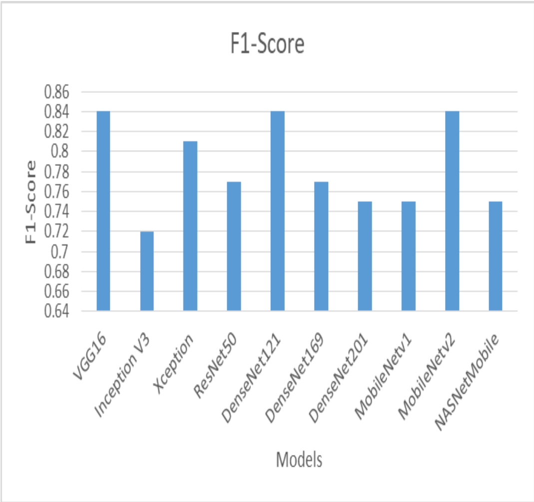

Fig. d. F1 Score

Fig. 4. Classification Metrics

6 Comparative Analysis

Literature studies reveal that by tuning network hyper parameters the performance of network can be improved. Table 10 demonstrates a comparative study with previously proposed classification approach and the accuracy of the classification. On comparing with other similar related work, it is observed that our proposed tuned models has been able to achieve promising results by choosing the appropriate combination of hyper parameter.

Table 10. Comparison between previously related work and proposed approach

Models	Reference	Hyperparameters	Accuracy
	[26]	Learning Rate=0.001, Learning rate decay=10^{-6} , Batch Size=50, No. of Epoch=100 , momentum=0.9	78%
	[27]	Optimizer=Adam,SGD ,Learning rate	80.34%

VGG16		range 0.1 and 0.25	
	[36] (Binary classification)	Adam optimizer ,learning rate of 0.001 ,momentum 0.09,batch size of 32, drop-out rate 0.05	55%
	Our VGG16 Model	Adam Optimizer ,Learning Rate:0.001, Dropout=0.5,No. of Epoch=40 , Batch Size=32	85%
Inceptionv3	[26]	Learning Rate=0.001, Learning rate de-cay=10^{-6} , Batch Size=50, No. of Epoch=100 , momentum=0.9	73%
	[27]	Optimizer: Adam, SGD ,Learning rate range 0.1 and 0.25	49.12%
	[36] (Binary classification)	Adam optimizer ,learning rate of 0.001 ,momentum 0.09,batch size of 32, drop-out rate 0.05	55%
	Our Inception Model	Adam Optimizer ,Learning Rate:0.001, Dropout=0.5,No. of Epoch=10 , Batch Size=32	74%
ResNet50	[26]	Learning Rate=0.001, Learning rate de-cay=10^{-6} , Batch Size=100, No. of Epoch=150 , momentum=0.9	73%
	[27]	Optimizer=Adam,SGD ,Learning rate range 0.1 and 0.25	78.12
	[36] (Binary classification)	Adam optimizer ,learning rate of 0.001 ,momentum 0.09,batch size of 32, drop-out rate 0.05	55%
	Our Resnet50 Model	Adam Optimizer ,Learning Rate:0.00002, Dropout=0.5, No. of Epoch=15 , Batch Size=32	77%
Xception	[27]	Optimizer=Adam,SGD ,Learning rate range 0.1 and 0.25	43.97%
	Our Xception Model	Adam Optimizer ,Learning Rate:0.001, Dropout=0.5,No. of Epoch=10 , Batch Size=32	81%
MobileNetV1	[36]	Adam optimizer ,Learning rate of 0.001 ,momentum 0.09,batch size of 32, drop-out rate 0.05	51.25%
	Our Mo-bileNetv1 Model	Adam Optimizer , Learning Rate:0.00002, Dropout=0.5,No. of Epoch=30 , Batch Size=32	81%
MobileNetV2	[31]	Adam optimizer ,Learning rate of 0.000002 ,No. of epoch 60	81%
	Our mo-bileNetv2 Model	Adam optimizer ,Learning rate of 0.00002 ,batch size of 32, dropout rate 0.05 , no. of epoch 20	84%

NASNetMobile	[36]	Adam optimizer ,Learning rate of 0.001 ,momentum 0.09,batch size of 32, drop-out rate 0.05	50.62%
	Our NASNetMobile Model	Adam Optimizer , Learning Rate:0.00001, Dropout=0.5,No. of Epoch=30 , Batch Size=32	80%

7 Conclusion

The hyper-parameters are the configurable parameters of the network .Tuning them is beneficial to improve the accuracy of the machine learning model. For tuning the ten DCNN networks hyperparameter such as optimizer, learning rate, batch size, dropout and number of epoch that decides how networks are to be trained are considered. Validation accuracy, recall, specificity, f1score, precision metrics are considered for performance evaluation. Table 4 to Table 8 depicts class-wise accuracies, recall, specificity, precision, and f1 score respectively of ten networks for each of the five classes from dataset. According to the validation accuracy top four shortlisted models are VGG16 with accuracy 85%, MobileNetv2 with accuracy 83%, Xception and DenseNet121 with accuracy 81%. While comparing the performance of networks F1 score helps to determine which classifier is better.F1 score for Vgg16, Densenet121 , MobileNetv2 is 84% while F1score for Xception 81%.

While comparing proposed tuned vgg16 and Mobilenetv2 network with other researchers tuned network for classification of diabetic retinopathy images, it is found that the our networks are performing better (Table 10). For learning rate with Adam optimizer values 0.001, 0.00001 and 0.00002 are found to be optimal. The recommended batch size for small image dataset is 32 and number of epoch should be in multiple of 10. This work contributes towards the improvement of the results achieved for diabetic retinopathy.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Saeedi, Pouya, et al (2019) Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas. *Diabetes research and clinical practice* 157, pp 107843.

[2] Asiri, N. , et.al (2019) Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey. *Artificial intelligence in medicine*, *99*, pp 101701

[3] Rupa Patel, Anita Chaware (2021). Computer Vision and Recognition Systems Using Machine and Deep Learning Approaches , In: The Institution of Engineering and Technology ,pp 253-276

[4] K. Simonyan, A. Zisserman (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[5] K. He, X. Zhang et.al (2015) Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385

[6] C. Szegedy, W. Liu, et.al(2015) Going Deeper with Convolutions, In :The IEEE Conference on Computer Vision and Pattern Recognition

[7] F. Chollet et. al (2016) Xception: Deep Learning with Depth wise Separable Convolutions

[8] Mishra, Supriya; Hanchate, Seema; Saquib, Zia (2020). Diabetic Retinopathy Detection using Deep Learning,IEEE 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics , Bengaluru ,pp 515–520. doi:10.1109/ICSTCEE49637.2020.9277506

[9] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q.(2017) Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* ,pp. 4700-4708. http://doi.org/10.1109/CVPR.2017.243

[10] Howard, A. G., Zhu, M., Chen, et.al., " Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861,2017.

[11] Mark, S. et al., "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp 4510-4520.

[12] Zoph, B., Vasudevan, V., Shlens, J. and Le, Q.V.(2018) Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* pp. 8697-8710.

[13] Weerts , H.J., Mueller, A.C. and Vanschoren, J.(2020). Importance of tuning hyperparameters of machine learning algorithms. *arXiv preprint arXiv:2007.07588*.

[14] Jacek M. Zurada (1994) Introduction to Artificial Neural System, Jaico Publishing House.

[15] Jin H. et al.(2022) Hyperparameter Importance for Machine Learning Algorithms. arXiv preprint arXiv:2201.05132

[16] A. Gulcu ,Z. KUş,(2020),Hyper-Parameter Selection in Convolutional Neural Networks Using Microcanonical Optimization Algorithm, in IEEE Access, vol. 8, pp. 52528-52540, doi: 10.1109/ACCESS.2020.2981141

[17] Jafar A, Myungho L. (2020) Hyperparameter Optimization for Deep Residual Learning in Image Classification. In2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C) ,pp. 24-29

[18] Yu T, Zhu H. (2020) Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689.

[19] Amirabadi, M.A., Kahaei, M.H. et al.(2020) Novel suboptimal approaches for hyperparameter tuning of deep neural network [under the shelf of optical communication]. Physical Communication, 41, p.101057.

[20] Lerman, P.M., (1980) Fitting segmented regression models by grid search. Journal of the Royal Statistical Society: Series C (Applied Statistics), 29(1), pp.77-84.

[21] James Bergstra , Bengio (2012), Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).

[22] Bergstra, J., et al.(2011) Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, *24*.

[23] Loshchilov, I. and Hutter, F., (2016) CMA-ES for hyperparameter optimization of deep neural networks. *arXiv preprint arXiv:1604.07269*.

[24] Mockus, J., (2012). *Bayesian approach to global optimization: theory and applications* (Vol. 37). Springer Science & Business Media.

[25] Pratt, H., Coenen, F., Broadbent, D.M., Harding, S.P. and Zheng, Y., 2016. Convolutional neural networks for diabetic retinopathy. *Procedia computer science*, *90*, pp.200-205

[26] Pradhan A, Dey BK, Nath RK, Sarma B.(2020) Transfer learning based classification of diabetic retinopathy stages. In2020 International Conference on Computational Performance Evaluation (ComPE),IEEE ,pp. 813-819.

[27] Vimukthi, Y., Kodikara, N. and Nanayakkara, L., (2020). A Deep Learning Based Approach for the Classification of Diabetic Retinopathy in Human Retina.

[28] Mohammadian, S., Karsaz, A. and Roshan, Y.M., 2017, November. Comparative study of fine-tuning of pre-trained convolutional neural networks for diabetic retinopathy screening. In 2017 24th National and 2nd International Iranian Conference on Biomedical Engineering (ICBME) (pp. 1-6). IEEE.

[29] https://www.kaggle.com/c/aptos2019-blindness-detection/data

[30] Arora, M. and Pandey, M., (2019), February. Deep neural network for diabetic retinopathy detection. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) ,pp. 189-193, IEEE.

[31] Rupa Patel and Anita Chaware,(2020),Transfer Learning with Fine-Tuned MobileNetV2 for Diabetic Retinopathy, In 2020 International Conference for Emerging Technology (INCET) , Belgaum, India**,** pp. 1-4, IEEE. doi: 10.1109/INCET49848.2020.9154014

[32] Rupa Patel and Anita Chaware (2021) Quantizing MobileNet Models for Classification Problem ",INDIACom-2021; 15th INDIACom; 2021 8th International Conference on Computing for Sustainable Global Development,organised by BVICAM, New Delhi**.**

[33] Asiri, N., Hussain, M., Al Adel, F. and Alzaidi, N., (2019.) Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey. *Artificial intelligence in medicine*, *99*, p.101701

[34] Wang YE, Wei GY, Brooks D.,(2019) Benchmarking tpu, gpu, and cpu platforms for deep learning. arXiv preprint arXiv:1907.10701.

[35] Rawat, Waseem & Wang, Zenghui. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation. 29. 1-98. 10.1162/NECO_a_00990

[36] Aswathi, T., Swapna, T.R. and Padmavathi, S., (2021), Transfer Learning approach for grading of Diabetic Retinopathy. In Journal of Physics: Conference Series (Vol. 1767, No. 1, p. 012033). IOP Publishing.

[37] Rambabu, B. ., Vikranth, B. ., Anupkanth, S. ., Samya, B. ., & Satyanarayana, N. . (2023). Spread Spectrum based QoS aware Energy Efficient Clustering Algorithm for Wireless Sensor Networks . International Journal on Recent and Innovation Trends in Computing and Communication, 11(1), 154–160. https://doi.org/10.17762/ijritcc.v11i1.6085