Analysis of MLP, CNN, and Transfer Learning Using VGG-16 for CIFAR-10 Dataset

Shahid Mahmood[1], Moneeb Gohar[1], Athar Waqar Alvi[1], Waleed Ahmad[2], Muhammad Kashif Khattak[*2], Anum Mushtaq[2]
[1]Bahria University Islamabad, Pakistan
[2]University of Poonch Rawalakot, AJK, Pakistan
[*]**Correspondence:** mkashifkhattak@upr.edu.pk

Citation | Mahmood. S, Gohar. M, Alvi. A. W, Ahmad. W, Khattak. M. K, Mushtaq. A, "Analysis of MLP, CNN, and Transfer Learning Using VGG-16 for CIFAR-10 Dataset", IJIST, Vol. 6 Issue. 4 pp 1826-1838, Oct 2024
Received | Oct 02, 2024 **Revised |** Oct 26, 2024 **Accepted |** Oct 30, 2024 **Published |** Oct 31, 2024.

Artificial Neural Networks (ANN) are becoming the core domain of Artificial Intelligence. Generally, Machine learning and specifically, deep learning gained popularity in problem-solving by virtue of Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and transfer learning approach. Transfer learning is becoming a powerful and successful technique for a variety of computer vision and image analysis applications due to its capability of reusing well-known proven architectures and their weights. Identification of optimum architecture and classifier along with pre-trained architectures is one of the challenging tasks in achieving optimum accuracy in various image analysis tasks. This paper investigates the performance of MLP, CNN, and transfer learning approaches using VGG-16 by tweaking hyperparameters and classifier architecture. The investigations and critical analysis revealed that MLP and CNN architectures have achieved about 55 % and 80 % validation accuracy on test data. Further experiments using VGG-16 architecture with MLP as a classifier have achieved more than 93 % accuracy on standard specification hardware for image classification on the CIFAR-10 dataset.

Keywords: Artificial Neural Networks (ANN); Convolutional Neural Network (CNN); Multi-Layer Perceptron (MLP); CIFAR-10 Dataset; VGG-16.

Introduction:

Artificial intelligence (AI) is influencing almost every field of life through the implementation of information technologies. In the current era, AI generally refers to machine learning (ML) and more precisely ML refers to Deep learning (DL) which is Artificial Neural Networks (ANN) with Multi-Layer Perceptron (MLP). Single-layer perceptron is mostly used for solving linear problems however, additional layers may be placed among the input layer and the output layer to solve non-linear problems. These network architectures with multiple hidden layers are called Multi-Layer Perceptron (MLP). MLP neural networks remain successful and accurate for classification problems when input is ambiguous and variable. Generally, image classification is considered one of the most popular and authentic problems to gauge efficiency for MLP, CNN, and Transfer Learning using pre-trained architectures VGG-16. However, according to the literature and our experiments, MLP is not meant for efficiently solving image classification problems. Specifically, on dataset CIFAR-10 MLP has shown about 55% accuracy in 300 epochs. Later on, the same problem was tried to solve using CNN, and optimizing various hyper-parameters achieved much better accuracy i.e., more than 80% on the same dataset CIFAR-10. Further experiments revealed that by the Transfer Learning approach using pre-trained architectures, VGG-16 accuracy can be further improved by more than 93%. The main objectives of this study are as follows:

- We want to compare the performance of MLP, CNN, and Transfer learning on the CIFAR-10 dataset using pre-trained architectures specifically VGG-16 for image classification.
- We want to explore the effects of different hyper-parameter settings on accuracy under various conditions including a number of layers, activation functions, neuron counts, and batch sizes.
- We want to identify the optimal configuration of each model for solving image classification tasks and evaluate the efficiency and accuracy of each approach.

In this paper, initial architectures for all three approaches are derived from the literature review. In the literature, only those MLP, CNN, and Transfer Learning approaches using pre-trained architectures VGG-16 are reviewed which are used for solving the same types of problems.

Literature Review:

The literature survey was mostly covered from 2018 to 2022 to understand which architecture of MLP and CNN can solve the image classification problem [1][2][3][4][5]. Implementation of MLP started with input, middle, and output layers and one output layer. In the initial stage, a literature survey revealed that a number of hidden layers can be determined by trial and error [2][6][7][8][9]. However, for CNN some of the old schemes were also covered. Some proposed schemes were only focused on the utilization of a limited amount of memory [10]. In 2017, a scheme was proposed by Aydogdu et. al to compare 3 architectures, namely 6-layer CNN, ResNet18, and ResNet34 using their age estimation accuracies [11]. One of the schemes was proposed in 2012 in this scheme Krizhevsky et.al proposed a CNN-based architecture (Alexnet). This scheme required 60 million parameters and approximately 50000 neurons to represent image features [12]. Other literature also helped in finalizing the initial architecture of CNN [13][14]. Accordingly, CNN has been designed to solve image classification problems on dataset CIFAR-10 with variation in hyperparameters.

A recently proposed study was performed which has shown that the accuracy is inversely proportional to the filter size [15]. In this paper, the accuracy of CNN is analyzed for image classification by employing CNN on the CIFAR-10 dataset. Literature suggested that VGG-16 can be used as pre-trained architecture for improvements in accuracy [16][17]. A study reported more than 90% accuracy while using VGG-16 as a feature extractor [18]. Similarly, other literature also concluded that VGG-16 can be used for image classification and the classifier part can be MLP [19]. This dataset contains 60,000 images and 10 classes, each a section containing 6000 images of each class. The accuracy of the MLP is analyzed by a varying number of Convolutional Filters (32, 64, and 128 of size 3x3), activation functions on the convolutional Layer and number of neurons, and varying batch size of training data. The padding parameter is not changed and keeps the value 'same'. It means that the system automatically pads as much as needed to preserve the spatial dimensions. It means that after applying convolution the output size of the image will remain the same as the input size of the image. Keeping the above studies in view our study presents the following novel contributions to the area:

- It gives a systematic comparison of MLP, CNN, and Transfer Learning (VGG-16) specifically for the CIFAR-10 dataset and highlights the limitations of MLP in image classification.
- Our study provides an extensive range of hyper-parameter variations in the context of CIFAR-10 for MLP and CNN, which have not been comprehensively analyzed together in existing literature.
- The findings reveal that MLP struggles with image classification, achieving only 55% accuracy after extensive epochs, while CNN and VGG-16 showcase significant improvements, with VGG-16 exceeding 93%. This insight is valuable for future research in model selection for image classification tasks.

Background:

This is an experimental and analytical study, in which the performance of MLP and CNN is investigated and analyzed by changing various parameters. The experiments are performed on two base architectures of MLP and CNN. Initially, the base architecture is set with the help of a literature review and existing proposed schemes for solving the same types of problems. In this section, we will explore the basic architectures used for the study.

MLP Basic Architecture:

In our study, we started our experiments with one input layer, middle layers, and one output layer with activation functions relu, sigmoid, and SoftMax. Later on, increased the number of hidden layers to four, six, and eight. The input layer contains 3072 (32x32x3) neurons because each image is 32x32 and multiplied with 3 for color images. The middle layer contains 512 neurons, which is revealed from the literature review for solving the same type of problem [6]. The output layer contains ten neurons because our dataset contains ten classes. This is how the initial structure of MLP is formulated as shown in 1 below.

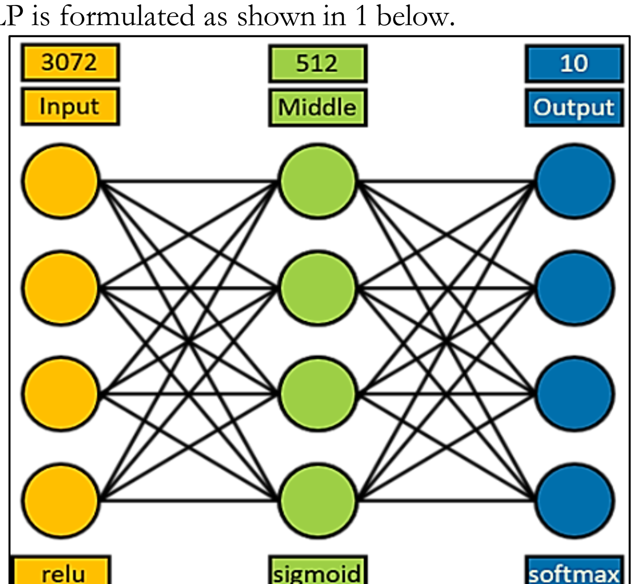

Figure 1. Basic Structure of MLP

CNN Basic Architecture:

The basic architecture of CNN consists of two convolutional layers for image feature extraction, one fully connected layer, and one output layer. This architecture is utilized as base architecture by keeping in view the processing capacity of the hardware and then during the experimentation phase parameters are changed one by one. The first convolutional layer consists of 32 filters with an activation function relu. The second Convolutional layer consists of 64 filters with anactivation function relu. The fully connected or dense layer consists of 128 neurons with activation function relu. The Output layer contains ten neurons and the activation function is SoftMax because the dataset contains ten classes. CNN architecture may be depicted as (Conv – Pool – Conv – Pool – FC – Softmax). The basic architecture is shown as 2 below:

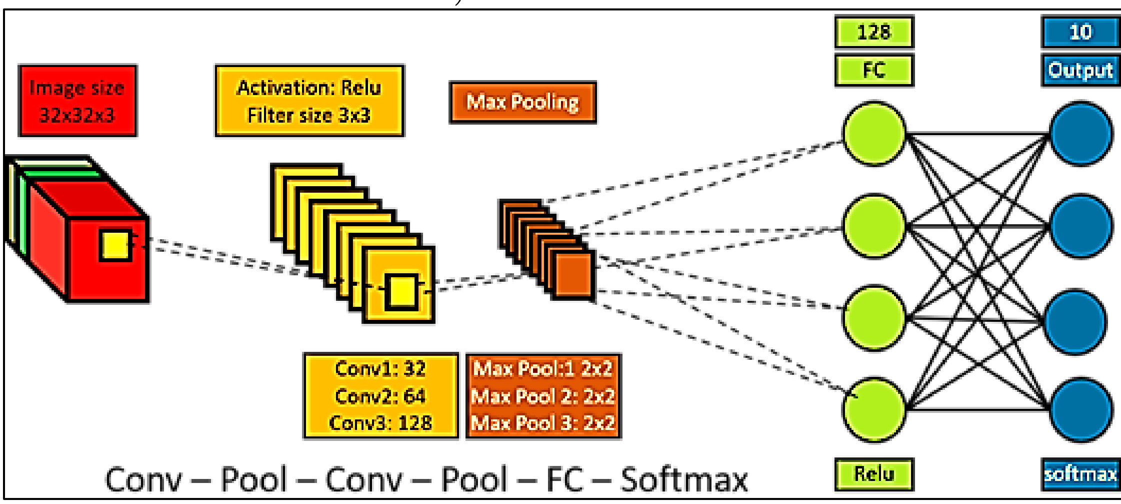

Figure 2. Basic Structure of CNN

Transfer Learning Using VGG-16 Architecture:

The basic architecture of VGG-16 consists of consists of sixteen convolutional layers. It contains only 3x3 convolutions and many filters which require 138 million parameters.
For training [20]. Its training requires a lot of processing and time. Therefore, pre-trained architecture has been used. Weights of the feature extraction part are used while two Fully connected layers are used for image classification. The experiments are performed by changing various numbers of layers and other hyperparameters. Input image size is modified as per VGG-16 and the output layer is used along with softmax for image classification of ten classes. This architecture is utilized as base architecture by keeping in view the processing capacity of the hardware and then during the experimentation phase parameters are changed one by one. The basic architecture is shown as 2 below:

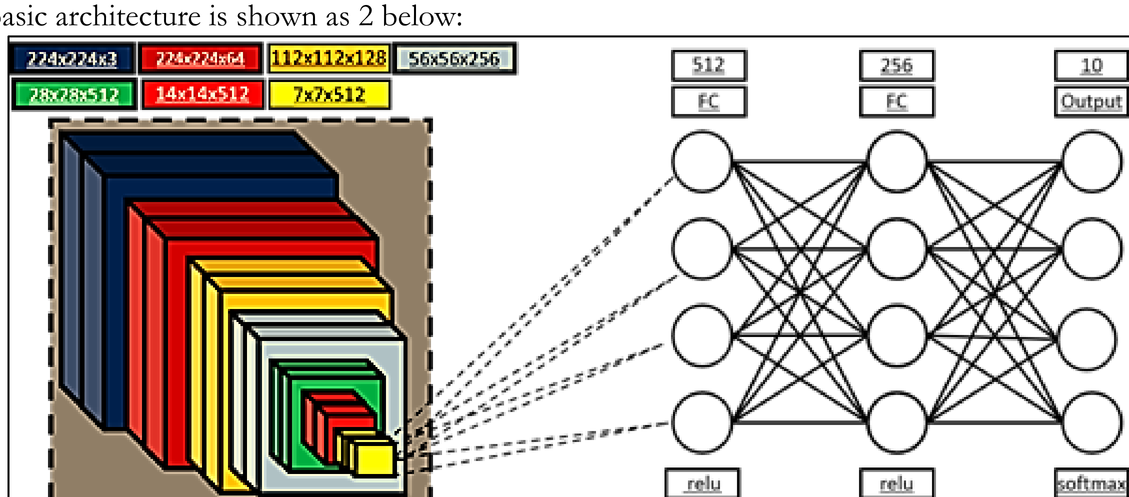

Figure 3: Transfer Learning using VGG-16 architecture

Methods:

Experiments are performed by tweaking with hyper-parameters to achieve optimum accuracy for image classification on the CIFAR-10 dataset. The CIFAR-10 dataset contains 60,000 images, out of these 50000 are used in the training and 10,000 in the test set. The following experiments are performed in this study

- Comparison of Changing of activation functions in MLP and CNN
- Increasing number of hidden Layers in MLP, Convolutional Layers in CNN, and Classifier part of VGG-16
- Comparison of Increasing Batch Size in MLP, CNN, and VGG-16
- Increasing number of Neurons at hidden layer - MLP
- Changing Filter size (3x3 and 5x5) - CNN
- Increasing Batch Size in MLP, CNN, and VGG-16
- Changing the number of filters in CNN

Initially, base architectures were finalized for MLP and CNN as shown in 1 and 2 respectively. All mentioned experiments for both MLP and CNN are performed with these base architectures. Only the first experiment of changing the activation function is performed for 300 epochs and all remaining experiments are performed for 30 epochs only. It was observed that a major chunk of performance was observed in the initial 30 epochs and performance in terms of accuracy was not much improved after 30 epochs. Therefore, all remaining experiments are performed for 30 epochs. This number of 30 epochs is also selected due to the processing capacity of the hardware being used. The experiments were started on MLP architecture with changing activation function and 300 number of epochs but accuracy was not increased more than 55%.

Table 1. MLP - two experiments for 300 epochs

Activation Functions	Loss	Acc	Val loss	Val acc
Relu, Relu and SoftMax	1.107165	0.604140	1.437880	0.514000
Sigmoid, Sigmoid and SoftMax	.444463	0.842060	1.839087	0.547400

There are two types of activation functions that are used at the input and hidden layer that is relu and sigmoid. The activation function at the output layer is SoftMax because there are ten output classes that are required to be predicted. In other words, there are ten types of images contained in the Cifar-10 Dataset therefore activation functions relu and sigmoid cannot be used because both are used for binary classification problems. In this initial model, accuracy was not improved after 300 epochs for both relu and sigmoid. After running 300 epochs with each activation function (relu and sigmoid) it was determined that optimum accuracy is achieved after 30 epochs. As shown in Table 1.

Similarly, the base architecture i.e., (Conv – Pool – Conv – Pool – FC – Softmax) of CNN was started with two convolutional layers. One fully connected (FC) layer and one output layer. The experiments were performed for 300 epochs by changing activation functions only and the final results are shown below in Table 2.

Table 2. CNN - two experiments for 300 epochs

Activation Functions	Loss	Acc	Val Loss	Val Acc
Relu and Softmax	0.129039	0.956325	1.143812	0.763200
Sigmoid and Softmax	0.170970	0.939025	1.405599	0.690900

In the above experiments relu as activation function, has achieved 76% performance and the sigmoid 69% performance approximately. Keeping in view the performance of MLP and CNN after changing the activation function it was decided to change a number of hidden layers for both architectures one by one.

In MLP, the number of hidden layers is increased to two, four, six, and eight but accuracy is still not achieved. At this MLP model was at the initial stage i.e., input, muddle, and output layer with relu as activation function. More precisely, the accuracy increased very slightly to six layers and even went down when the number of layers increased to eight as shown in Table 3.

Table 3. MLP experiments for the impact of hidden layers

No. of Conv Layers	Loss	Acc	Val loss	Val acc
Two Layers	1.209291	0.568600	1.294840	0.539500
Four Layers	1.315512	0.526040	1.379602	0.513500
Six Layers	1.356751	0.517020	1.361899	0.514800
Eight Layers	2.302760	0.099480	2.302685	0.100000

It was established that by increasing the number of hidden layers in MLP the accuracy cannot be improved. Accuracy remained at about 51% up to six layers but became 10% when the number of layers was increased to eight.

Similarly, the number of convolutional layers in CNN is also increased. At this stage, models reverted back to the initial base architecture with two convolutional layers (Conv– Pool – Conv – Pool – FC – SoftMax). Similarly, with three convolutional layers (Conv – Pool- Conv – Pool – Conv – Pool– FC – SoftMax). One additional layer of 128 filters is added and accuracy is improved to optimum i.e., 80% as shown in Table 4, and experiments are performed for 30 epochs.

Table 4. CNN - experiments for the impact of convolutional layers

No. of Conv Layers	Loss	Acc	Val Loss	Val Acc
Two Layers	0.205517	0.927925	0.985787	0.756100
Three Layers	0.484919	0.828700	0.617790	0.801900

The experiments in transfer learning are performed by freezing the convolutional part of VGG-16 which works as a feature extractor and changing the number of fully connected layers which works as a classifier part. VGG-16 gives an output of 7x7x512, after this first fully connected layer of 512 Neurons is added as FC-1. The second fully connected layer of 256 Neurons is faded as FC-2. Similarly, FC-3 and FC-4 are the third and fourth layers added in the classifier part, which consists of 128 and 64 neurons respectively. The following results for VGG-16 are performed for 10 epochs as it was determined that accuracy was not much increased after 10 epochs however accuracy reached 93 percent on 29 epochs as shown in the results section. This has increased accuracy as shown in Table 5.

Table 5. VGG-16 - experiments for the impact of fully connected layers

No. of Conv Layers	Loss	Acc	Val Loss	Val Acc
One Layers	0.0347	0.9891	0.3394	0.9183
Two Layers	0.0336	0.9890	0.3877	0.9156
Three Layers	0.0430	0.9868	0.3922	0.9137
Four Layers	0.0581	0.9850	0.3921	0.9220

At this stage, MLP was also analyzed in parallel when accuracy was not improved even after increasing the number of neurons. The batch size of training data was changed from 128, 256, 512, and 1024. This also has not increased noteworthy accuracy as shown in Table 6.

Table 6. MLP - experiments for the impact of batch size

Batch Size	Loss	Acc	Val Loss	Val Acc
128	1.209291	0.568600	1.294840	0.539500
256	1.190482	0.574000	1.302161	0.537800
512	1.197028	0.573240	1.272801	0.547300
1024	1.233234	0.557260	1.311221	0.527700

Similarly, the batch size of training was also changed for CNN as shown in Table 7 Batch size parameter is also analyzed in experiments performed in transfer learning using VGG-16. Results are shown as shown in Table 8. Further investigations and experiments are performed on MLP architecture by increasing the number of neurons at the middle layer. In the initial model, there were 512 neurons at the middle layer but increased to 1024 and 2048 number of neurons. This option also could not work out in terms of improving accuracy as shown in Table 7. In MLP architecture impact was analyzed after increasing the number of neurons in hidden layers. The maximum accuracy is observed at approximately 54% when the number of neurons is 1024. The impact is shown below in Table 9.

Table 7. CNN - experiments for the impact of batch size

Batch Size _	Loss	Acc	Val loss	Val acc
64	0.749295	0.732525	0.793265	0.728900
128	0.469884	0.829850	0.747334	0.756100
256	0.817036	0.707675	0.792271	0.729400

In CNN experiment was performed to analyze the impact of the Number of Filters and maximum accuracy was achieved at 75%. The impact after 30 epochs is shown below in Table 10. In all the above experiments both architectures of MLP, CNN, and transfer learning approach using VGG-16 are analyzed with respect to the percentage of accuracy. It was determined that MLP and CNN architectures have achieved about 55% and 80 % validation accuracy on test data. While in transfer learning using VGG-16 architecture with MLP as a classifier has achieved more than 93 %.

Table 8. VGG-16 - experiments for the impact of batch size

Batch Size	Loss	Acc	Val Loss	Val Acc
64	0.0336	0.9890	0.3877	0.9156
128	0.0392	0.9878	0.4112	0.9122
256	0.0565	0.9819	0.4349	0.9020

Table 9. Experiments for the impact of a number of neurons at the hidden layer

No of Neurons	Loss	Acc	Val Loss	Val Acc
512	1.209291	0.568600	1.294840	0.539500
1024	1.203366	0.570100	1.282822	0.540700
2048	1.307405	0.534120	1.344401	0.516100

Table 10. CNN - Experiments for the impact of a number of filters

No. of Filters	Loss	Acc	Val Loss	Val Acc
Conv1 32 and Conv2 64	0.469884	0.829850	0.747334	0.756100
Conv1 64 and Conv2 128	0.686536	0.751675	0.773568	0.742400

Experiments for the Impact of Activation Functions on MLP and CNN:

Experiments of changing activation functions were performed on the architectures of MLP and CNN. There are two experiments performed on each architecture to observe the impact of the activation function on the accuracy of solving image classification problems on the same dataset. In the first experiment of MLP on the model training and testing were performed for 300 epochs with the activation functions (relu, relu, and softmax) at the input, middle, and output layers. In the second experiment of MLP all other parameters remained the same with only a change in activation functions i.e., (sigmoid, sigmoid, and softmax), and the same experiment was run for 300 epochs. Similarly, In the first experiment of CNN on the model training and testing were performed for 300 epochs with the activation functions (Conv-1 relu, Conv-2 relu, FC relu, and softmax). In the second experiment of MLP all other parameters remained the same with only change in activation functions i.e., Conv-1 sigmoid, Conv-2 sigmoid, FC sigmoid, and softmax) and the same experiment is run for 300 epochs.

MLP - Experiments for Impact of Hidden Layers (Two, Four, Six, and Eight):

There are four experiments performed to observe the impact of increasing the hidden layer for the accuracy of solving image classification problems on the dataset. The initial setup remained the same and experiments were performed with one input layer, middle layers, and one output layer with activation functions relu, sigmoid, and softmax. All the additional fully connected hidden layers of 512 neurons are increased with the activation function sigmoid. As mentioned earlier that started with two Layers and an output layer and then gradually increased by two additional layers in each iteration. During these experiments, the impact of time taken was also observed.

CNN - Experiments for Impact of Convolutional Layers (Two, and Three):

There are two experiments performed on CNN architecture to observe the impact of increasing the convolutional layer for the accuracy of solving image classification problems on the dataset. The initial setup remained the same and experiments were performed with (Conv-1 relu, Conv-2 relu, FC relu, and softmax). One additional convolutional layer of 128 filters is added to the model. The second experiment is performed on (Conv-1 relu, Conv-2 relu, Conv-3 relu, FC relu, and softmax. During these experiments, accuracy was observed to improve.

Experiments for Impact of Number of Neurons at Hidden Layer (512, 1024 and 2048):

There are three experiments performed to observe the impact of increasing the number of neurons at the hidden Layer for accuracy in solving image classification problems on the dataset. The initial setup remained the same and experiments were performed with one input layer, middle layers, and one output layer with activation functions relu, sigmoid, and softmax. The number of Neurons increased at the middle layers only.

MLP -Experiments for Impact of Batch Size (128, 256, 512 and 1024):

There are four experiments performed to observe the impact of increasing the batch size of the training dataset for the accuracy of solving image classification problems on the dataset. By increasing the batch size number of steps is decreased in each iteration. The objective of this experiment was to increase the accuracy of image detection and fast convergence to optimum accuracy values.

CNN -Experiments for Impact of Batch Size (64, 128 and 256):

There are three experiments performed to observe the impact of increasing the batch size of the training dataset for the accuracy of solving image classification problems on the dataset. By increasing the batch size number of steps is decreased in each iteration. The objective of this experiment was to increase the accuracy of image detection and fast convergence to optimum accuracy values.

VGG-16 -Experiments for Impact of Batch Size (64, 128 and 256):

There are three experiments performed for transfer learning using VGG-16. By observing the impact of increasing the batch size for accuracy of solving image classification problems on the dataset. By increasing the batch size number of steps is decreased in each iteration. The objective of this experiment was to increase the accuracy of image detection and fast convergence to optimum accuracy values. There is no major impact seen on the accuracy of results.

Results and Analysis:

Results are analyzed for all types of experiments in terms of improved accuracy and reduction of loss. The accuracy with all parameters changing and observed results is discussed with respect to changing parameters. Optimum accuracy for MLP is observed at approximately 55% with 300 epochs and sigmoid as activation function at two hidden layers and SoftMax at the output layer. 55% accuracy was achieved for 1024 neurons but this was computationally extensive as it was taking too much time as shown in 6. The same accuracy was also achieved with base architecture as shown in 4 and 8 in just 30 epochs of MLP as it was referred to in literature. CNN

Impact and Analysis of Activation Functions on MLP and CNN:

In MLP architecture activation function of the input and middle layer is changed which has slightly improved accuracy but is not significant. The objective of changing the activation function was to improve accuracy and reduce loss but significant change was not observed major accuracy around 50 percent was achieved at twenty to thirty epochs. Later on, 270 epochs have only achieved four to five percent. The experiment was performed for 300 epochs as shown in the figure below. In CNN architecture activation function has also shown slight improvement in the case of relu as compared to sigmoid. However, sigmoid has taken more time than relu as shown in 5.

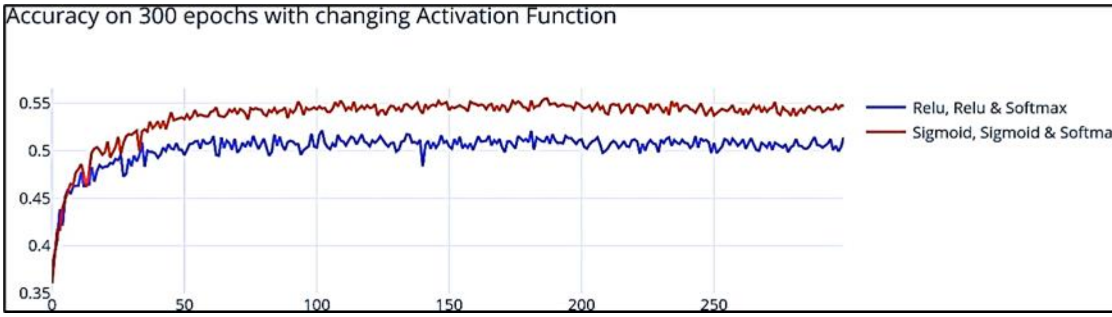

Figure 4. MLP - Accuracy on 30 epochs with changing Activation Function

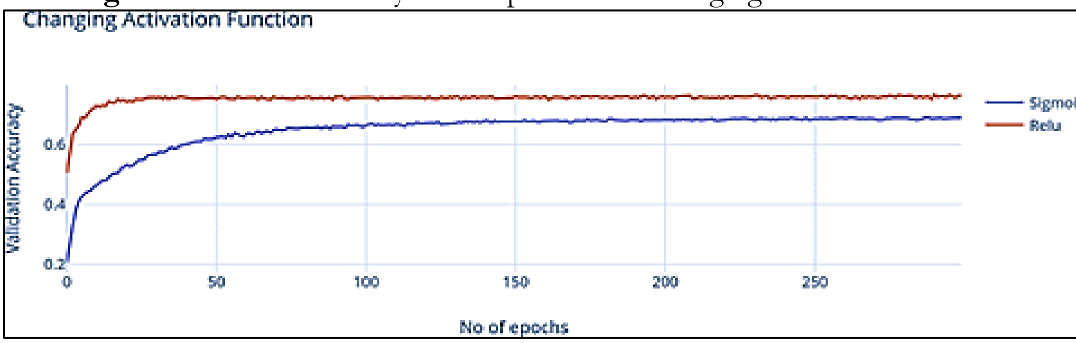

Figure 5. CNN - Accuracy on 30 epochs with changing Activation Function

Impact and Analysis Hidden Layers in MLP and Convolutional Layers in CNN:

Hidden layers were increased because accuracy was not achieved after 300 epochs. Therefore, keeping in view the complexity of the problem hidden layers were increased to four hidden layers, six hidden layers, and eight hidden layers. This experiment has increased the time required for each epoch from 15 seconds to 20 seconds and 25 seconds approximately. This variation also did not work significantly as shown in the figure below.

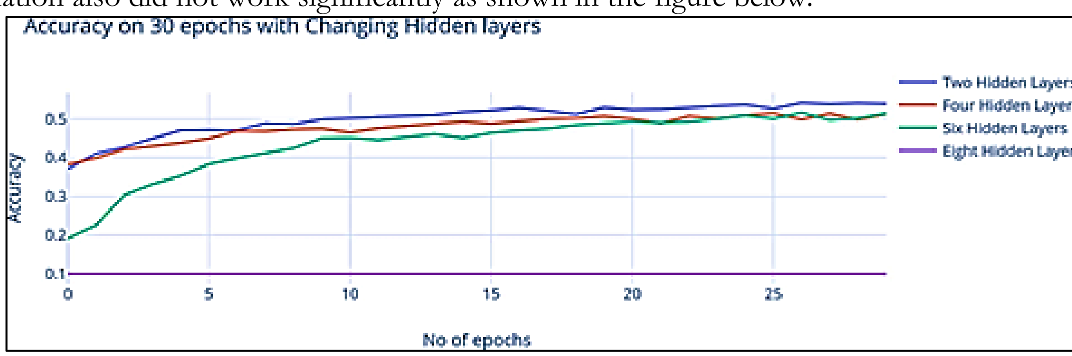

Figure 6. MLP-Impact and analysis number of neurons at hidden Layer (512, 1024, and 2048)

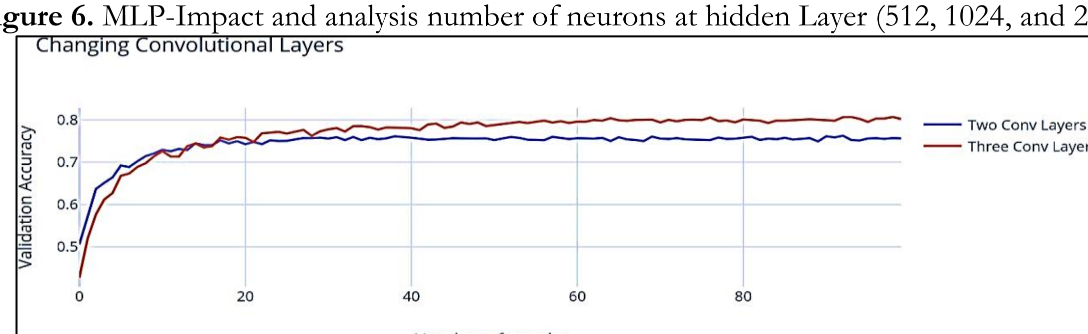

Figure 7. CNN - Impact and Analysis Changing Convolutional Layers

In CNN, the experiments were performed for two convolutional layers (with 32 and 64 filters) and three convolutional layers (with 32, 64 filters, and 128 filters). The results have shown an accuracy of approximately 80%. However, increasing convolutional layers to four convolutional layers (with 32, 64, and 256 filters) has reduced accuracy to 73%. In Transfer Learning, the experiments were performed by changing fully connected (FC) layers. This is the classification part of the architecture. FC-1 consists of 512 neurons. FC-2 consists of 256 neurons. FC-3 consists of 256 neurons. FC-4 consists of 128 neurons. The results have shown an accuracy of approximately 92%.

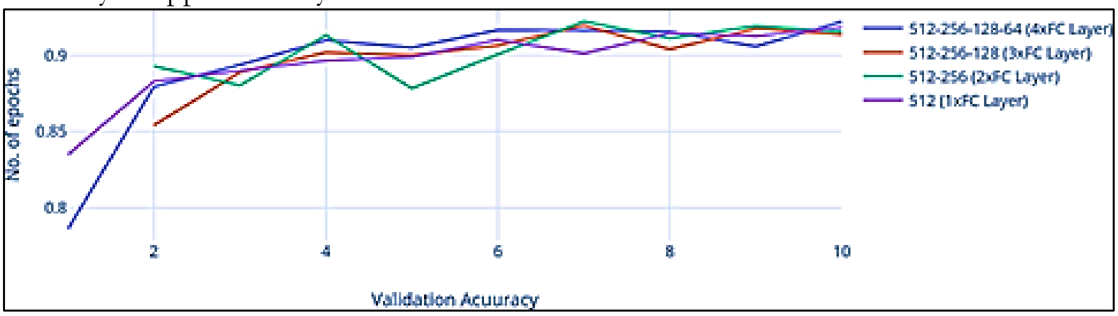

Figure 8. CNN - Impact and Analysis Changing Convolutional Layers

MLP-Impact and Analysis Number of Neurons at Hidden Layer (512, 1024 and 2048):

The initial model remained the same and experiments performed by increasing the number of Neurons increased at middle layers only from 512, 1024, and 2048. There is no significant change in accuracy observed even though accuracy reduced from 53 percent to 51 percent. Similarly, the loss is also increased from 1.29 to 1.34 as shown in the figure below.

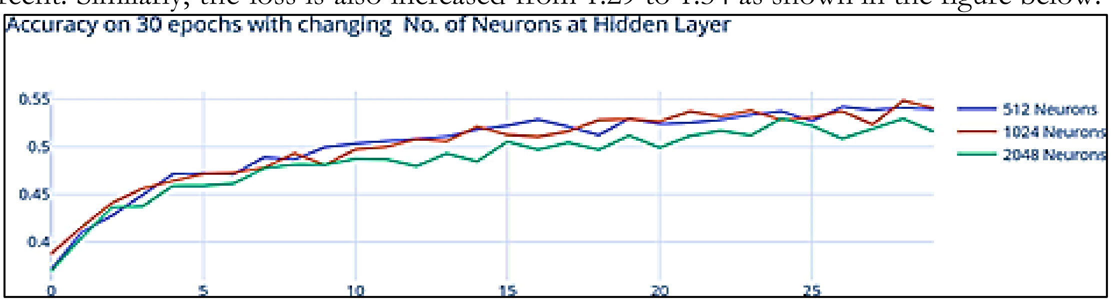

Figure 9. Accuracy on 30 epochs with changing No of Neuron

MLP-Impact and Analysis Batch Size (128, 256, 512 and 1024):

In view of the above, when accuracy was not much improved, the batch size of training data was increased. Initially, when batch size increased from 128 to 256 and 512 it improved accuracy from 53 percent to 54 percent but later on when increased batch size to 1024 the accuracy again decreased to 53 percent and similarly no major change in loss is also observed as shown 10.

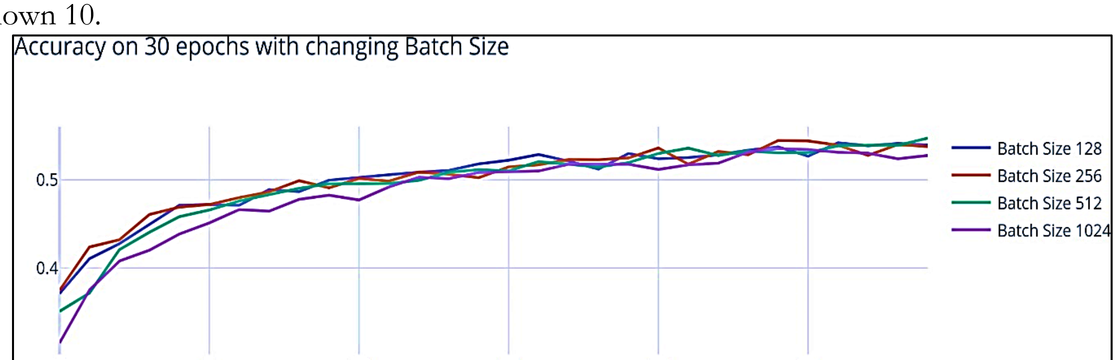

Figure 10. MLP-Accuracy on 30 epochs with changing Batch Size

CNN-Impact and Analysis Batch Size (64, 128 and 256):

Similarly, the accuracy of CNN was also observed by changing batch sizes. A batch size of 128 has shown an optimum accuracy of approximately 75%. But when batch size was increased to 256 accuracy was reduced to approximately 73% as shown in 11. Therefore, batch sizes were not more than 256.

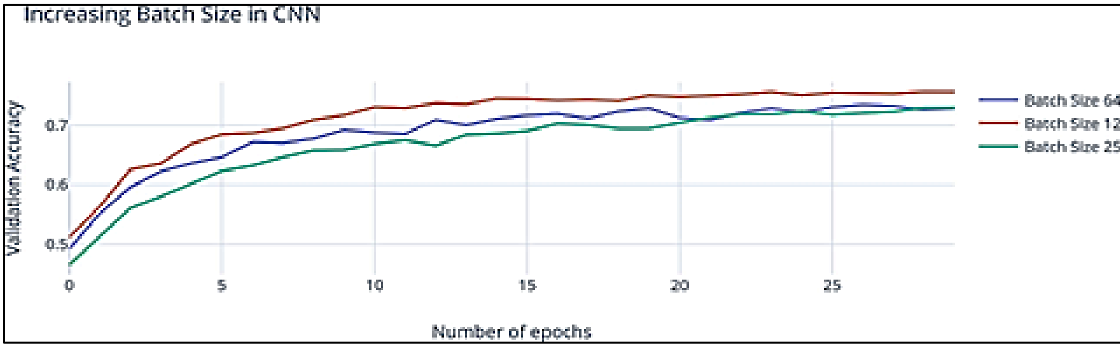

Figure 11. CNN-Accuracy on 30 epochs with changing Batch Size

VGG-16-Impact and Analysis Batch Size (64, 128 and 256):

Similarly, the accuracy of VGG-16 was also observed by changing batch sizes. All three batch sizes have shown an optimum accuracy of approximately 92% as shown in 12. Therefore, batch sizes were not more than 256.

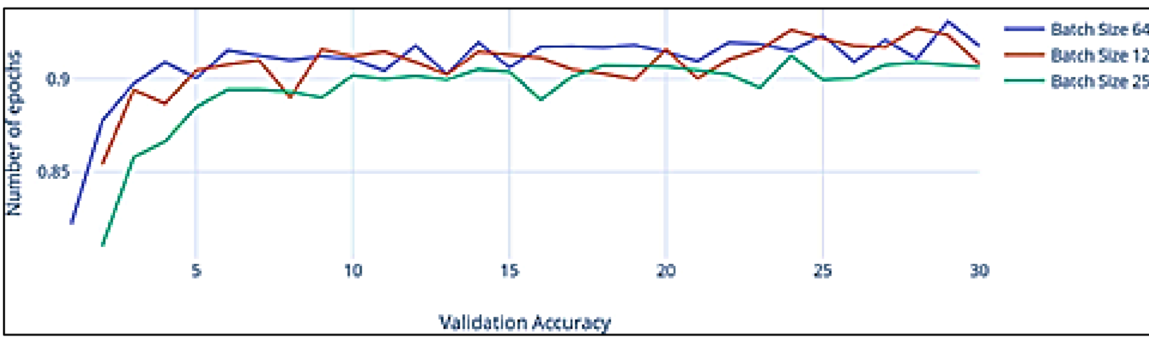

Figure 12. VGG-16-Accuracy on 30 epochs with changing Batch Size

The Final architecture for optimum accuracy using pre-trained weights of VGG-16 is with two fully connected layers of 512 and 256 neurons as shown in 4.

Conclusion:

In this comparative and analytical experimentation, it is concluded that transfer learning using VGG-16 as a feature extractor has shown better accuracy as compared to multi-layer Perceptron (MLP) and Convolutional Neural Networks (CNN). MLP can classify images in a given dataset CIFAR-10 with an accuracy of almost 55 percent. An optimum accuracy of about 50 percent was achieved in the initial twenty to thirty epochs. Later on, other epochs do not affect accuracy significantly. Similarly, changing activation functions, increasing the number of neurons, and increasing the number of hidden layers just add processing overhead without significant improvements in accuracy and loss. Therefore, MLP is not a suitable solution for solving image classification problems. The same image classification problem foe CIFAR-10 is also solved with CNN and it is observed that more than 80% accuracy is achieved in 30 epochs only. Further investigations reveal that pre-trained VGG-16 can achieve more than 93/The same dataset CIFAR- 10.

Specifications:

- The experiments were performed on the following specifications of hardware. Complete source code and results are uploaded on Git Hub. https://github.com/Shahidmscs/Deep-Learning/tree/main
- OS:Microsoft Windows 10 Pro Model: HP Laptop 15 − bs 0xx
- Type: x64 − based PC

- I n t e l (R) Core (TM) i 3 −7100U
- CPU @ 2. 4 0 GHz, 2401 Mhz, 2 Core (s)
- 4 Logical Processor (s) RAM: 8. 0 0 GB

Author's Contribution: All the authors contributed equally to this paper.

Conflict of Interest: Authors have no conflict of interest.

References:

[1] Q. Jiang, L. Zhu, C. Shu, and V. Sekar, "Multilayer perceptron neural network activated by adaptive Gaussian radial basis function and its application to predict lid-driven cavity flow," Acta Mech. Sin. Xuebao, vol. 37, no. 12, pp. 1757–1772, Dec. 2021, doi: 10.1007/S10409-021-01144-5/TABLES/3.

[2] E. Avuçlu and F. Başçiftçi, "New approaches to determine age and gender in image processing techniques using multilayer perceptron neural network," Appl. Soft Comput., vol. 70, pp. 157–168, Sep. 2018, doi: 10.1016/J.ASOC.2018.05.033.

[3] R. Román-Godínez, E. Zamora, H. Sossa, and A. Info, "A Comparative Study of Dendrite Neural Networks for Pattern Classification," Int. J. Comb. Optim. Probl. Informatics, vol. 12, no. 3, pp. 8–19, Sep. 2021, Accessed: Oct. 30, 2024. [Online]. Available: https://ijcopi.org/ojs/article/view/252

[4] M. Lin, W. Jing, D. Di, G. Chen, and H. Song, "Multi-Scale U-Shape MLP for Hyperspectral Image Classification," IEEE Geosci. Remote Sens. Lett., vol. 19, 2022, doi: 10.1109/LGRS.2022.3141547.

[5] C. Shah, Q. Du, and Y. Xu, "Enhanced TabNet: Attentive Interpretable Tabular Learning for Hyperspectral Image Classification," Remote Sens. 2022, Vol. 14, Page 716, vol. 14, no. 3, p. 716, Feb. 2022, doi: 10.3390/RS14030716.

[6] S. Ajala, "Artificial neural network implementation for image classification using cifar-10 dataset," vol. 10, 2021.

[7] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li, "S2-MLP: Spatial-Shift MLP Architecture for Vision," Proc. - 2022 IEEE/CVF Winter Conf. Appl. Comput. Vision, WACV 2022, pp. 3615–3624, 2022, doi: 10.1109/WACV51458.2022.00367.

[8] A. Farzad and T. A. Gulliver, "Log message anomaly detection with fuzzy C-means and MLP," Appl. Intell., vol. 52, no. 15, pp. 17708–17717, Dec. 2022, doi: 10.1007/S10489-022-03300-1/METRICS.

[9] P. Viertel and M. König, "Pattern recognition methodologies for pollen grain image classification: a survey," Mach. Vis. Appl., vol. 33, no. 1, pp. 1–19, Jan. 2022, doi: 10.1007/S00138-021-01271-W/FIGURES/7.

[10] R. C. Calik and M. F. Demirci, "Cifar-10 Image Classification with Convolutional Neural Networks for Embedded Systems," Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA, vol. 2018-November, Jul. 2018, doi: 10.1109/AICCSA.2018.8612873.

[11] M. F. Aydogdu, V. Celik, and M. F. Demirci, "Comparison of Three Different CNN Architectures for Age Classification," Proc. - IEEE 11th Int. Conf. Semant. Comput. ICSC 2017, pp. 372–377, Mar. 2017, doi: 10.1109/ICSC.2017.61.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.

[13] S. Divya, B. Adepu, and P. Kamakshi, "IMAGE ENHANCEMENT AND CLASSIFICATION OF CIFAR-10 USING CONVOLUTIONAL NEURAL NETWORKS," Proc. - 4th Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2022, 2022, doi: 10.1109/ICSSIT53264.2022.9716555.

[14] H. Zhang and X. Luo, "The Role of Knowledge Creation-Oriented Convolutional Neural Network in Learning Interaction," Comput. Intell. Neurosci., vol. 2022, no. 1, p.

6493311, Jan. 2022, doi: 10.1155/2022/6493311.
[15] O. M. Khanday, S. Dadvandipour, and M. A. Lone, "Effect of filter sizes on image classification in CNN: a case study on CFIR10 and Fashion-MNIST datasets," IAES Int. J. Artif. Intell., vol. 10, no. 4, pp. 872–878, Dec. 2021, doi: 10.11591/ijai.v10.i4.pp872-878.
[16] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," Proc. - 3rd IAPR Asian Conf. Pattern Recognition, ACPR 2015, pp. 730–734, Jun. 2016, doi: 10.1109/ACPR.2015.7486599.
[17] R. Chitic, F. Leprévost, and N. Bernard, "Evolutionary algorithms deceive humans and machines at image classification: an extended proof of concept on two scenarios," J. Inf. Telecommun., vol. 5, no. 1, pp. 121–143, Jan. 2021, doi: 10.1080/24751839.2020.1829388.
[18] C. S. and S. Parikh, "Comparison of cnn and pre- trained models: A study", [Online]. Available: https://www.researchgate.net/publication/359850786_Comparison_of_CNN_and_Pre-trained_models_A_Study?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6ImxvZ2luIiwicGFnZSI6InNlYXJjaCIsInBvc2l0aW9uIjoicGFnZUhlYWRlciJ9fQ
[19] T. Alshalali and D. Josyula, "Fine-tuning of pre-trained deep learning models with extreme learning machine," Proc. - 2018 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2018, pp. 469–473, Dec. 2018, doi: 10.1109/CSCI46756.2018.00096.
[20] C. Alippi, S. Disabato, and M. Roveri, "Moving Convolutional Neural Networks to Embedded Systems: The AlexNet and VGG-16 Case," Proc. - 17th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks, IPSN 2018, pp. 212–223, Oct. 2018, doi: 10.1109/IPSN.2018.00049.