Deeper Networks for Image Classification using Pytorch

Ganesh Kumaran Masilamani - 200434339
Department of Electronics Engineering and Computer Science
Queen Mary University Of London
London, United Kingdom
ec20049@qmul.ac.uk

Abstarct---***Deep Learning is the most recent advancement in the field of Artificial Intelligence, which is used worldwide. This helps the system to achieve better Accuracy. Image Classification has become much feasible using Deep Learning techniques, in which we can process Large datasets. Nowadays most of the Image Classification uses Convolutional Neural Networks. In this paper, Image Classification is performed using the VGG16, ResNet, GoogleNet, AlexNet, MobileNet in MNIST and CIFAR10 Dataset. The MNIST Dataset is the handwritten data between 0 to 9 with 60,000 grayscaled images of 28×28 pixels, and the CIFAR10 Dataset contains 60,000 Colour Images of 10 classes of pixel size 32×32. After experimentation it is found that, in MNIST data all the models achieved about 99% accuracy. In CIFAR10 data ResNet achieved better accuracy of 88.84% and loss of 0.0004%.***

Keywords--- VGG16, ResNet, GoogleNet, MNIST Dataset, CIFAR10 Dataset, Convolutional Neural Networks.

1. INTRODUCTION

The field of Deep Learning and Computer vision in Artificial Intelligence is rapidly growing. This improvement has made image classification much easier with the larger dataset, where larger datasets can be processed with better accuracy. Inspite of many techniques such as Approximate Nearest Neighbour (ANN) being used for Image Classification, using Deep Learning Models classifies the images with the better accuracy than other techniques.

X-Ray images are processed using ResNet in [1] to identify the COVID-19 affected patients. This analysis is done using the X-Ray which is taken for the PCR test and the Artificial intelligence is used to identify the patients more accurately so that the spread can be reduced.

In paper [2], COVID-19 is detected using the different VGG models likely, VGG16 and VGG19. Same process is carried out as in [1], but with different Deep Neural Network. The literatures show that there are lot of advantages in using the Deep network models for the Classification purposes.

2. RELATED WORKS

The Development in the field of deep learning had helped in investigation of large datasets in Image Classification.

In the paper by Ke Zhang[3], DenseNet is used for the Image classification, in which weights are changed and the model is formed. It increases the representation power by recalibrating the features channel-wise. The authors finally combined the MFR-DenseNet and the CFR-DenseNet with ILFR-DenseNet. From this effectiveness of the Networks are measured. This paper concludes that CIFAR100 dataset works better with DenseNet.

The paper [4], uses pretrained models, which saves the time for building and training the model. It also gives the better results. In this paper the authors used VGG16, VGG10, ResNet and InceptionV3 models in classification of the indian food items. This ends with the conclusion that the InceptionV3 model gives better accuracy and minimise the loss to the greater extent.

Yukun Huang.Et.al [5], deals with the high-resolution Multispectral Images, in which a new network is developed from the Scratch. It is ResNet with the pyramidal Residual network which ends up in improving the Classification performance. In the paper by Mian Mian Lau [6], Deep Convolutional Feedforward Neural Network is developed. This is created to monitor the performance of the deep networks. Fully connected layer can be increased to maximise the analysing power efficiency.

Ahmet Demir [7], proposed a Inception-ResNet-V2 model to perform Image Classification.

LeakyReLU is the activation function which was used in this paper, and classification is performed. Secondly, Maxpooling layers are created and the classification is performed once again. Same is done with the Average pooling layers and the results are compared. Approximately 93% accuracy is achieved finally.

VGG16 Network is used in the paper which was proposed by Dewa Made Sri Arsa [8]. In this they have used VGG16 with the Random Forest classifier. This classification was performed in the public batik dataset. After experimentation Accuracy, Precision, F-score were noted. This proposed model outperformed the existing one, which produces accuracy of 97%.

The paper by Shyava Tripathi [9], proposes a Small Convolutional Neural Networks to perform Image Classification, which is much easier to implement and observe better classification accuracy on any Image datasets.

Arti P.Et.al [10] in their research have used transfer learning with the pretrained Deep Networks for image classifcation. For this, authors have used the IRMA dataset which consists of 6406 X-ray images. After experimentation the AlexNet model acheives best accuracy of 0.93 among other models.

The paper by Arpana Mahajan [11], proposes a ResNet model, where after training SVM classifier is used to extract the features with minimal loss in the model. It is the fastest method to extract the features from the image categorical data and hence used in the classification of such datasets.

Rui Yang.Et.al [12] proposed a DenseNet architecture which is otherwise known as the Multi-Scale Input Spatial Pyramid Pooling Fusion Networks. It is used in the classification of the Climate Zones. LCZ42 is the dataset which was used here. This architecture achieves the accuracy of 90.79% in the test set.

The Next paper was proposed by Emine Cengil [13], in which transfer learning using pretrained weights was used in the models such as AlexNet, GoogleNet, VGG16, DenseNet, ResNet. In this paper VGG16 model outperformed other deep convolutional models and achieved the accuracy of 93.53%.

In the paper by Mehmet Sevi.Et.al [2], VGG models are used of the classification of COVID-19 patients. From this the precision, Recall and the F-score of the system were calculated. These are classified with the multiclass classification.

Yian Seo paper [14], describes the Image classification using pre-trained Convolutional Neural Networks. In this paper the authors used the GoogleNet on the fine tuned fashion dataset. The Network in this work achieved 62% accuracy on the testset after 10 folds.

As like [2], same COVID-19 detection is performed using the X-Ray images in the paper by Zehra Karhan [1], but in this ResNet is used in the place of VGG models. After experimentation it is found that the maximum accuracy that can be reached is 99.5% in 5 fold cross validation.

In paper by Qiang Li [15], Deep Networks are used in the classification of the Image Datasets, Three datasets are created based on the texture, shape and the measurement scale. The data are processed with different Networks such as Alexnet, VGGNet, GoogleNet. Among these AlexNet performed better in all the aspects.

The paper by Minjun Zhu [16] describes Image classification that is performed using the CIFAR dataset in VGG, DenseNet, LSDNE using the Greedy algorithm. Of all this LSDNE gives better accuracy and has more advantages and the loss is reduced over 14% than other models.

3. MODEL DESCRIPTION

In this paper, we have used different types of deep neural networks for evaluating the effectiveness of the different deep networks models for image classification on MNIST and CIFAR10 Datasets.

3.1 Model Architecture

(I) VGG-16

VGG16 otherwise known a OxfordNet, is the Convolutional Deep Network which is named after the Visual Geometry Group from oxford. It is easy to implement in practical. VGG16 consists of 16 layers which has approximately of 138 million parameters. It is one of the popular Architecture for Image Classification.

VGG16 Model is built with the sequence of the layers. Input dimensions are fixed with 244×244 image size. The first layer is the Convolutional layer which is the 3×3 convolutional layer with padding, which is to maintain the resolution in the model. The second layer will be the ReLU function which is used for the activation and it often tends to be Non-linear and the final layer would be the Maxpooling layer which is used for downsampling.

The value of 0.2 is fixed instead of the default value which is 0.5 in the dropout layer.

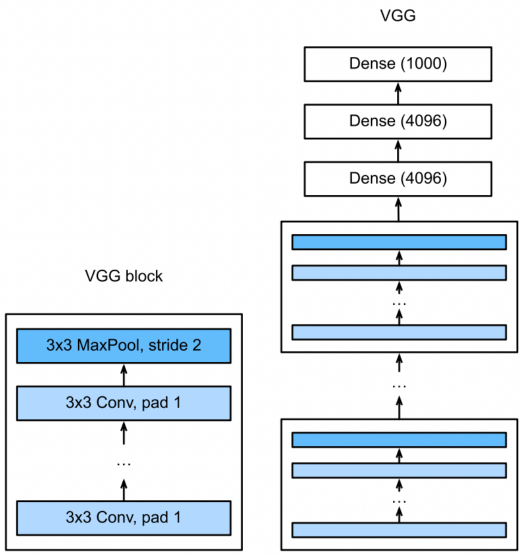


Fig 3.1.1. VGG16 Architecture

(II) ResNet

The Basic building block of the ResNet is known as Residual block that builds on the pyramidal cells in the cortex, that can be achieved by utilizing the skip connections and the jumping over some of the neural layers. ResNet uses Batch Normalisation as its root. This improves the input of the residual network which increases the performance of the network. The idea of residual network is based on identity shortcut connection, in which the network can skip some layers.

This will not affect the accuracy of the model since layers only stack the image mapping. This is because other than stacking layers do anything on accuracy.

The Dataset is fed into the ResNet block. The residual network consists of 4 residual blocks. The first layer of the residual block is 3×3 convolutional layer and then the data needs to be normalized so that, it is passed through the Batch-normalization. After that activation function is used. For this ReLU activation is used. This process is continued in the loop in the residual block.

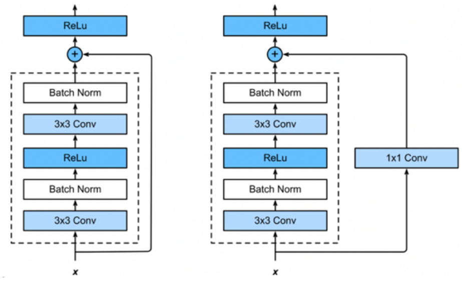


Fig 3.1.2. Residual Block Architecture

Finally, ResNet is constructed using the 4 Residual Blocks in which Network skips some layers. The Residual block is connected with 7×7 Convolutional Layer, 3×3 Maxpooling layer and finally it is passed through the Global average pooling layer. This helps the data not getting overfitted by enforcing correspondences among mapping and the features .

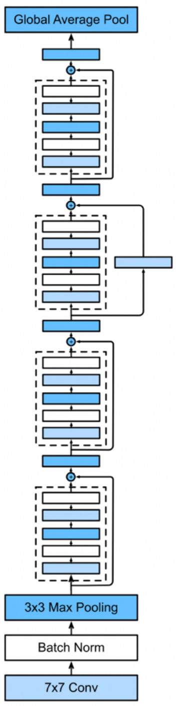


Fig 3.1.3. ResNet Architecture

(III) GoogleNet

This Architecture has 22 deep Network layers with 27 pooling layers. GoogleNet contains 9 Inception modules which is splitted into 2,5,2 in 3 layers, These inception modules are connected to the global average pooling layer.

The Inception Block consists of 4 layer in which the first layer has the input connected with the 1×1 Convolutional network. The second layer has the input connected with the 1×1 convolutional network and then it is passed through the 3×3 convolutional network. When coming to the third layer same happens like the second layer, but instead of 3×3 it is connected to the 5×5 convolutional Network, and the final layer consists of 3×3 Max pooling layer and the 1×1 convolutional network. Finally all the four layers are concatenated; this forms the inception block.

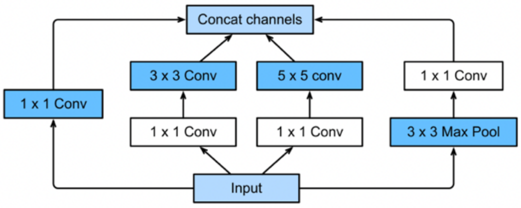


Fig 3.1.4. Inception Block Architecture

Then, the GoogleNet architecture is formed as like ResNet.

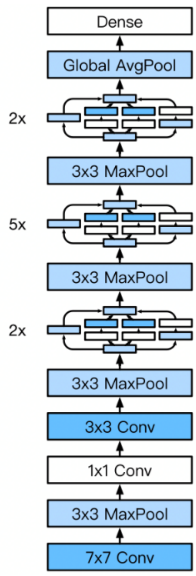


Fig 3.1.5. GoogleNet Architecture

Dataset is passed through the 7×7 convolutional layer and passed through the 3×3 Max pooling layer and transferred through the 1×1 and 3×3 convolutional layer and as per the GoogleNet architecture it is passed through the 3×3 max pooling layer. Next step is to pass the inception block two times in the network and again passed with 3×3 maxpooling layer. Then it is passed with the Inception block 5 times and with Maxpooling layer. Inception block is passed two times and finally it is passed with the global average pooling to make all the layers fully connected and forms the Network.

4. EXPERIMENTATION

4.1 Datasets

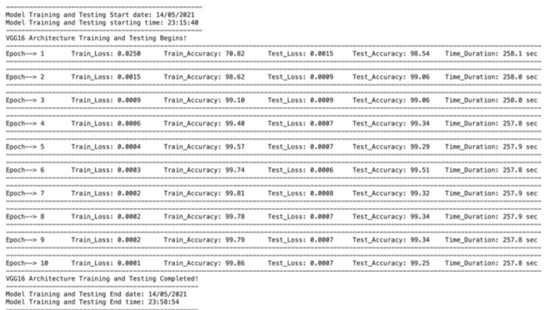
Two Datasets are used. One is the MNIST dataset which contains 60000 samples splitted into 50000 training samples and the 10000 test samples, which is of 28×28 Image pixel size. This dataset is available in the MNIST database website.

The second dataset we will be using is the CIFAR10. This dataset is used for the further investigation of the models. CIFAR10 dataset is available in the Toronto.edu site which contains 60000 images separated among 10 classes of 32×32 images size.

4.2 Testing Results

4.2.1 VGG16 on MNIST Dataset

VGG16 model is trained for 10 epochs on MNIST data with the learning rate of 0.1 and the following results are observed,



```
Model Training and Testing Start date: 14/05/2021
Model Training and Testing starting time: 23:15:40
VGG16 Architecture Training and Testing Begins!
Epoch--> 1    Train_Loss: 0.0250    Train_Accuracy: 70.82    Test_Loss: 0.0015    Test_Accuracy: 98.54    Time_Duration: 258.1 sec
Epoch--> 2    Train_Loss: 0.0015    Train_Accuracy: 98.62    Test_Loss: 0.0009    Test_Accuracy: 99.06    Time_Duration: 258.0 sec
Epoch--> 3    Train_Loss: 0.0009    Train_Accuracy: 99.10    Test_Loss: 0.0009    Test_Accuracy: 99.06    Time_Duration: 258.0 sec
Epoch--> 4    Train_Loss: 0.0006    Train_Accuracy: 99.40    Test_Loss: 0.0007    Test_Accuracy: 99.34    Time_Duration: 257.8 sec
Epoch--> 5    Train_Loss: 0.0004    Train_Accuracy: 99.57    Test_Loss: 0.0007    Test_Accuracy: 99.29    Time_Duration: 257.9 sec
Epoch--> 6    Train_Loss: 0.0003    Train_Accuracy: 99.74    Test_Loss: 0.0006    Test_Accuracy: 99.51    Time_Duration: 257.8 sec
Epoch--> 7    Train_Loss: 0.0002    Train_Accuracy: 99.81    Test_Loss: 0.0008    Test_Accuracy: 99.32    Time_Duration: 257.9 sec
Epoch--> 8    Train_Loss: 0.0002    Train_Accuracy: 99.78    Test_Loss: 0.0007    Test_Accuracy: 99.34    Time_Duration: 257.9 sec
Epoch--> 9    Train_Loss: 0.0002    Train_Accuracy: 99.79    Test_Loss: 0.0007    Test_Accuracy: 99.34    Time_Duration: 257.8 sec
Epoch--> 10   Train_Loss: 0.0001    Train_Accuracy: 99.86    Test_Loss: 0.0007    Test_Accuracy: 99.25    Time_Duration: 257.8 sec
VGG16 Architecture Training and Testing Completed!
Model Training and Testing End date: 14/05/2021
Model Training and Testing End time: 23:58:54
```

Fig 4.2.1.1. VGG16 Architecture Results

VGG model achieves the accuracy of 99.86% in the training data and 99.25% in the test set, which makes the model to function more accurately and the prediction would be accurate. In MNIST data all the models which we trained for 10 epochs reached 99% accuracy on both train and test data which predicts more accurately.

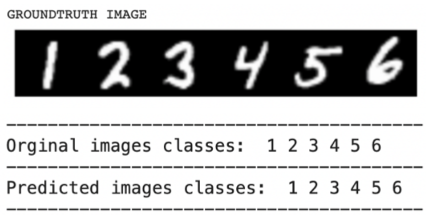


Fig 4.2.1.2. Original and Predicted image classes for MNIST in VGG16

The individual accuracies of the image classes other than 3 and 3 achieves 99% which is the maximum among all.

4.2.2 ResNet on MNIST Dataset

The MNIST data is trained on the ResNet model for 10 epochs with the learning rate of 0.1 and the following results are observed,

```
Model Training and Testing Start date: 14/05/2021
Model Training and Testing starting time: 23:15:46
ResNet Architecture Training and Testing Begins!
Epoch--> 1    Train_Loss: 0.0073    Train_Accuracy: 94.31    Test_Loss: 0.0013    Test_Accuracy: 98.67    Time_Duration: 42.7 sec
Epoch--> 2    Train_Loss: 0.0012    Train_Accuracy: 98.83    Test_Loss: 0.0008    Test_Accuracy: 99.10    Time_Duration: 42.7 sec
Epoch--> 3    Train_Loss: 0.0008    Train_Accuracy: 99.24    Test_Loss: 0.0009    Test_Accuracy: 99.08    Time_Duration: 42.8 sec
Epoch--> 4    Train_Loss: 0.0005    Train_Accuracy: 99.50    Test_Loss: 0.0007    Test_Accuracy: 99.32    Time_Duration: 42.8 sec
Epoch--> 5    Train_Loss: 0.0004    Train_Accuracy: 99.65    Test_Loss: 0.0009    Test_Accuracy: 99.12    Time_Duration: 42.7 sec
Epoch--> 6    Train_Loss: 0.0003    Train_Accuracy: 99.74    Test_Loss: 0.0012    Test_Accuracy: 98.97    Time_Duration: 42.8 sec
Epoch--> 7    Train_Loss: 0.0002    Train_Accuracy: 99.79    Test_Loss: 0.0008    Test_Accuracy: 99.29    Time_Duration: 42.8 sec
Epoch--> 8    Train_Loss: 0.0002    Train_Accuracy: 99.83    Test_Loss: 0.0007    Test_Accuracy: 99.50    Time_Duration: 42.7 sec
Epoch--> 9    Train_Loss: 0.0002    Train_Accuracy: 99.82    Test_Loss: 0.0006    Test_Accuracy: 99.50    Time_Duration: 42.8 sec
Epoch--> 10   Train_Loss: 0.0001    Train_Accuracy: 99.90    Test_Loss: 0.0007    Test_Accuracy: 99.44    Time_Duration: 42.9 sec
ResNet Architecture Training and Testing Completed!
Model Training and Testing End date: 14/05/2021
Model Training and Testing End time: 23:23:14
```

Fig 4.2.2.1. ResNet Architecture Results

The ResNet Model achieves the train accuracy of 99.90% and the test accuracy of 99.44% with 0.0001 train loss and 0.0007 test loss.

In this model, the accuracies of 7,8 and 9 achieves 100% , which tells that all the predictions were correct.

4.2.3 GoogleNet on MNIST Dataset

We have trained the GoogleNet model for 10 epochs on MNIST data with the learning rate of 0.1 and the following results are observed,

```
Model Training and Testing Start date: 14/05/2021
Model Training and Testing starting time: 23:15:52
GoogleNet Architecture Training and Testing Begins!
Epoch--> 1    Train_Loss: 0.0413    Train_Accuracy: 54.46    Test_Loss: 0.0039    Test_Accuracy: 96.48    Time_Duration: 43.5 sec
Epoch--> 2    Train_Loss: 0.0039    Train_Accuracy: 96.51    Test_Loss: 0.0023    Test_Accuracy: 97.79    Time_Duration: 43.4 sec
Epoch--> 3    Train_Loss: 0.0020    Train_Accuracy: 98.08    Test_Loss: 0.0013    Test_Accuracy: 98.73    Time_Duration: 43.4 sec
Epoch--> 4    Train_Loss: 0.0013    Train_Accuracy: 98.69    Test_Loss: 0.0015    Test_Accuracy: 98.58    Time_Duration: 43.5 sec
Epoch--> 5    Train_Loss: 0.0010    Train_Accuracy: 99.04    Test_Loss: 0.0017    Test_Accuracy: 98.43    Time_Duration: 43.5 sec
Epoch--> 6    Train_Loss: 0.0008    Train_Accuracy: 99.20    Test_Loss: 0.0012    Test_Accuracy: 98.81    Time_Duration: 43.4 sec
Epoch--> 7    Train_Loss: 0.0007    Train_Accuracy: 99.33    Test_Loss: 0.0011    Test_Accuracy: 98.96    Time_Duration: 43.5 sec
Epoch--> 8    Train_Loss: 0.0005    Train_Accuracy: 99.49    Test_Loss: 0.0012    Test_Accuracy: 99.00    Time_Duration: 43.4 sec
Epoch--> 9    Train_Loss: 0.0004    Train_Accuracy: 99.55    Test_Loss: 0.0017    Test_Accuracy: 98.41    Time_Duration: 43.5 sec
Epoch--> 10   Train_Loss: 0.0004    Train_Accuracy: 99.64    Test_Loss: 0.0014    Test_Accuracy: 98.81    Time_Duration: 43.4 sec
GoogleNet Architecture Training and Testing Completed!
Model Training and Testing End date: 14/05/2021
Model Training and Testing End time: 23:23:26
```

Fig 4.2.3.1. GoogleNet Architecture Results

This Model achieves the train accuracy of 99.64% and the test accuracy of 98.81% with 0.0004 train loss and 0.0014 test loss.

From the above figure, we can observe that the accuracy of 1,7 and 9 achieves 100% with no flaw predictions. From the accuracy we can say that the inception layers works properly which gives the better accuracy, in otherwords we can say that it helps in decreasing the loss to the great extent.

4.3 Further Evaluation

The Further validation is performed on the CIFAR10 Dataset

4.3.1 VGG16 on CIFAR10 Dataset

The model is trained on the CIFAR data and the results are shown below.

```
Epoch--> 28   Train_Loss: 0.0021    Train_Accuracy: 90.41    Test_Loss: 0.0044    Test_Accuracy: 82.94    Time_Duration: 24.2 sec
Epoch--> 29   Train_Loss: 0.0020    Train_Accuracy: 90.94    Test_Loss: 0.0036    Test_Accuracy: 85.45    Time_Duration: 24.2 sec
Epoch--> 30   Train_Loss: 0.0019    Train_Accuracy: 91.46    Test_Loss: 0.0036    Test_Accuracy: 85.81    Time_Duration: 24.1 sec
Epoch--> 31   Train_Loss: 0.0018    Train_Accuracy: 91.97    Test_Loss: 0.0040    Test_Accuracy: 84.78    Time_Duration: 24.1 sec
Epoch--> 32   Train_Loss: 0.0017    Train_Accuracy: 92.22    Test_Loss: 0.0038    Test_Accuracy: 85.21    Time_Duration: 24.1 sec
Epoch--> 33   Train_Loss: 0.0017    Train_Accuracy: 92.18    Test_Loss: 0.0048    Test_Accuracy: 83.00    Time_Duration: 24.1 sec
Epoch--> 34   Train_Loss: 0.0016    Train_Accuracy: 92.85    Test_Loss: 0.0045    Test_Accuracy: 84.43    Time_Duration: 24.1 sec
Epoch--> 35   Train_Loss: 0.0016    Train_Accuracy: 92.98    Test_Loss: 0.0040    Test_Accuracy: 85.36    Time_Duration: 24.1 sec
Epoch--> 36   Train_Loss: 0.0015    Train_Accuracy: 93.22    Test_Loss: 0.0035    Test_Accuracy: 86.27    Time_Duration: 24.1 sec
Epoch--> 37   Train_Loss: 0.0015    Train_Accuracy: 93.43    Test_Loss: 0.0045    Test_Accuracy: 83.46    Time_Duration: 24.0 sec
Epoch--> 38   Train_Loss: 0.0014    Train_Accuracy: 93.53    Test_Loss: 0.0040    Test_Accuracy: 85.20    Time_Duration: 24.1 sec
Epoch--> 39   Train_Loss: 0.0013    Train_Accuracy: 94.14    Test_Loss: 0.0036    Test_Accuracy: 86.15    Time_Duration: 23.7 sec
Epoch--> 40   Train_Loss: 0.0013    Train_Accuracy: 94.18    Test_Loss: 0.0041    Test_Accuracy: 85.52    Time_Duration: 24.2 sec
VGG16 Architecture Training and Testing Completed!
Model Training and Testing End date: 14/05/2021
Model Training and Testing End time: 23:59:24
```

Fig 4.3.1.1. VGG16 Results on CIFAR10 data

This model is trained for 40 epochs on the CIFAR10 data which produces the final train accuracy of 94.18% and test accuracy of 85.52% with the train loss of 0.0013.

```
GROUNDTRUTH IMAGE
```

```
Orginal images classes:  horse dog ship plane ship bird
Predicted images classes:  horse dog ship plane ship deer
```

Fig 4.3.1.2. Original and Predicted image classes for CIFAR10 in VGG16

The individual accuracies of each image classes in CIFAR10 data. Among this, the image class for the plane achieves the maximum accuracy of 95% followed by car of 93%.

4.3.2 ResNet on CIFAR10 Dataset

The CIFAR10 data is trained on the ResNet model for 40 epochs with the learning rate of 0.1 and the following results are observed,

This Model produces the good accuracy but during prediction it predicts some data wrongly. When coming to accuracy this model achieves the Train accuracy of 97.91% and the test accuracy of 88.84% .

In the Resnet model, the CIFAR10 data predicts some classes incorrectly, although it achieves the maximum accuracy on the test set of 88.84%, which is higher than both the VGG16 and GoogleNet model test accuracy on the CIFAR10 dataset. It can be inferred that ResNet model works much good in CIFAR data.

```
Epoch--> 28    Train_Loss: 0.0009    Train_Accuracy: 96.12    Test_Loss: 0.0042    Test_Accuracy: 86.23    Time_Duration: 59.5 sec
Epoch--> 29    Train_Loss: 0.0008    Train_Accuracy: 96.33    Test_Loss: 0.0034    Test_Accuracy: 88.71    Time_Duration: 59.5 sec
Epoch--> 30    Train_Loss: 0.0008    Train_Accuracy: 96.43    Test_Loss: 0.0032    Test_Accuracy: 89.55    Time_Duration: 59.4 sec
Epoch--> 31    Train_Loss: 0.0007    Train_Accuracy: 96.65    Test_Loss: 0.0034    Test_Accuracy: 88.98    Time_Duration: 59.5 sec
Epoch--> 32    Train_Loss: 0.0007    Train_Accuracy: 96.97    Test_Loss: 0.0053    Test_Accuracy: 85.40    Time_Duration: 59.5 sec
Epoch--> 33    Train_Loss: 0.0006    Train_Accuracy: 97.16    Test_Loss: 0.0034    Test_Accuracy: 89.66    Time_Duration: 59.4 sec
Epoch--> 34    Train_Loss: 0.0006    Train_Accuracy: 97.13    Test_Loss: 0.0038    Test_Accuracy: 88.13    Time_Duration: 59.4 sec
Epoch--> 35    Train_Loss: 0.0006    Train_Accuracy: 97.29    Test_Loss: 0.0037    Test_Accuracy: 89.12    Time_Duration: 59.4 sec
Epoch--> 36    Train_Loss: 0.0006    Train_Accuracy: 97.51    Test_Loss: 0.0069    Test_Accuracy: 83.18    Time_Duration: 59.4 sec
Epoch--> 37    Train_Loss: 0.0006    Train_Accuracy: 97.47    Test_Loss: 0.0034    Test_Accuracy: 89.95    Time_Duration: 59.4 sec
Epoch--> 38    Train_Loss: 0.0005    Train_Accuracy: 97.73    Test_Loss: 0.0036    Test_Accuracy: 89.60    Time_Duration: 59.4 sec
Epoch--> 39    Train_Loss: 0.0005    Train_Accuracy: 97.78    Test_Loss: 0.0036    Test_Accuracy: 89.49    Time_Duration: 59.4 sec
Epoch--> 40    Train_Loss: 0.0005    Train_Accuracy: 97.91    Test_Loss: 0.0040    Test_Accuracy: 88.84    Time_Duration: 59.4 sec
ResNet Architecture Training and Testing Completed!
Model Training and Testing End date: 15/05/2021
Model Training and Testing End time: 00:43:03
```

Fig 4.3.2.1. ResNet Results on CIFAR10 data

In ResNet model, the image class of ship achieves 98% accuracy followed by the class of car which is of 97% and the deer of 95%.

4.3.3 GoogleNet on CIFAR10 Dataset

The GoogleNet model was trained for 40 epochs on CIFAR10 data with the learning rate of 0.1 with the batch size of 32 on the SGD optimizer. This model gives less test accuracy when compared to other developed deep models. The following results are observed,

```
Epoch--> 28    Train_Loss: 0.0046    Train_Accuracy: 79.54    Test_Loss: 0.0052    Test_Accuracy: 78.29    Time_Duration: 41.3 sec
Epoch--> 29    Train_Loss: 0.0044    Train_Accuracy: 80.25    Test_Loss: 0.0077    Test_Accuracy: 70.24    Time_Duration: 41.3 sec
Epoch--> 30    Train_Loss: 0.0042    Train_Accuracy: 81.33    Test_Loss: 0.0063    Test_Accuracy: 74.82    Time_Duration: 41.4 sec
Epoch--> 31    Train_Loss: 0.0041    Train_Accuracy: 81.97    Test_Loss: 0.0059    Test_Accuracy: 75.33    Time_Duration: 41.3 sec
Epoch--> 32    Train_Loss: 0.0039    Train_Accuracy: 82.63    Test_Loss: 0.0055    Test_Accuracy: 77.46    Time_Duration: 41.3 sec
Epoch--> 33    Train_Loss: 0.0038    Train_Accuracy: 83.41    Test_Loss: 0.0049    Test_Accuracy: 79.23    Time_Duration: 41.3 sec
Epoch--> 34    Train_Loss: 0.0036    Train_Accuracy: 83.81    Test_Loss: 0.0058    Test_Accuracy: 76.38    Time_Duration: 41.3 sec
Epoch--> 35    Train_Loss: 0.0034    Train_Accuracy: 84.66    Test_Loss: 0.0047    Test_Accuracy: 80.54    Time_Duration: 41.2 sec
Epoch--> 36    Train_Loss: 0.0034    Train_Accuracy: 85.03    Test_Loss: 0.0055    Test_Accuracy: 77.80    Time_Duration: 41.3 sec
Epoch--> 37    Train_Loss: 0.0032    Train_Accuracy: 86.06    Test_Loss: 0.0053    Test_Accuracy: 78.80    Time_Duration: 41.3 sec
Epoch--> 38    Train_Loss: 0.0031    Train_Accuracy: 86.29    Test_Loss: 0.0069    Test_Accuracy: 74.61    Time_Duration: 41.3 sec
Epoch--> 39    Train_Loss: 0.0030    Train_Accuracy: 86.88    Test_Loss: 0.0062    Test_Accuracy: 77.75    Time_Duration: 41.4 sec
Epoch--> 40    Train_Loss: 0.0029    Train_Accuracy: 87.29    Test_Loss: 0.0058    Test_Accuracy: 77.00    Time_Duration: 41.4 sec
GoogleNet Architecture Training and Testing Completed!
Model Training and Testing End date: 15/05/2021
Model Training and Testing End time: 00:31:02
```

Fig 4.3.3.1. GoogleNet Results on CIFAR10 data

GoogleNet with CIFAR10 data achieves the accuracy of 80%, which is 87.29% for train data and 77.00% for test data with the train loss of 0.0029 and 0.0058 of the loss occur during the testing of the model

```
Orginal images classes:  horse dog ship plane ship bird
Predicted images classes:  horse cat ship plane ship deer
```

Fig 4.3.3.2. Original and Predicted image classes for CIFAR10 in GoogleNet

There are two wrong predictions among the six in the GoogleNet model which is low among the other two models trained in the CIFAR10 dataset.

The individual accuracies of each image classes in CIFAR10 data in the GoogleNet. After comparing, the image class for the truck achieves the highest accuracy of 96% followed by the class cat of 90%.

5. CONCLUSION

Model	Dataset Used	No of Epochs	Train Loss	Final Train Accuracy	Test Loss	Test Accuracy	Training Time(Approx)
VGG16	MNIST	10	0.0001	99.86	0.0007	99.25	43 Minutes
ResNet	MNIST	10	0.0001	99.90	0.0007	99.44	7 Minutes
GoogleNet	MNIST	10	0.0004	99.64	0.0014	98.81	7 Minutes
VGG16	CIFAR10	40	0.0013	94.18	0.0041	85.52	16 Minutes
ResNet	CIFAR10	40	0.0005	99.91	0.0040	88.84	40 Minutes
GoogleNet	CIFAR10	40	0.0029	87.29	0.0058	77.00	28 Minutes

Fig 5.1 Comparison Table Between all the Developed Deep Network Models

The Deep Networks becomes the latest advancement in the field of Deep Learning and Artificial Intelligence, which helps in accurate prediction of data with the help of many layers of processing. The above figure shows the comparison between all the accuracies and the time constraints of the various models developed and trained in the different datasets.

In MNIST data, the developed models which includes VGG16, ResNet, reaches 99% test accuracy and GoogleNet reaches nearly to 99% which is 98.81%, which leads to accurate prediction of data in that specific dataset. Whereas when the model is futher evaluated in the CIFAR10 dataset, VGG and the GoogleNet models achieves 80% (approx) test accuracy but the ResNet model achieves 90% accuracy which is higher the both the above specified model. In CIFAR10 dataset, the ResNet model work much better in means of achieving better accuracy which also leads to the drop in the loss to the larger extent than both VGG and the GoogleNet models. From this, it is concluded that the ResNet model developed works better than other developed models.

REFERENCES

[1] Zehra Karhan, Fuat Akal, "COVID-19 Classification using Deep Learning in Chest X-Ray Images", Medical Technologies Congress (TIPTEKNO-2020), DOI: 10.1109/TIPTEKNO50054.2020.9299315

[2] Mehmet Sevi, Ilhan Aydin, "COVID-2019 Detection Using Deep Learning Methods", International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI-2020), DOI:10.1109/ICDABI51230.2020.932562.

[3] Ke Zhang, Yurong Guo, Qiaolin Ding, "Multiple Feature Reweight DenseNet for Image Classification", IEEE Access-2018 (Volume:7),DOI:10.1109/ACCESS.2018.2890127.

[4] J R Rajayogi, G Manjunath, G Shobha, "Indian Food Image Classification with Transfer Learning", 4th International Conference on Computational Systems and Information technology for Sustainable Solution (CSITSS-2019), DOI:1109/CSITSS47250.2019.9031051.

[5] Yukun Huang, Jingbo Wei, Wenchao Tang, Chaoqi He, "Pyramid Convolutional Neural Networks and Bottleneck Residual Modules for Classification of Multispectral Images", IEEE International geoscience and Remote Sensing Symposium (IGARSS-2020), DOI:1109/IGARSS39084.2020.9324314.

[6] Mian Mian Lau, Jonathan Sien Phang, King Hann Lim, "Convolutional Deep Feedforward Network for Image Classification", 7th International Conference on Smart Computing & Communications (ICSCC-2019), ISBN:978-1-7281-1558-0, DOI:1109/ICSCC.2019.8843642.

[7] Ahmet Demir, Feyza Yilmaz, "Inception-ResNet-v2 with Leakyrelu and Averagepooling for More Reliable and Accurate Classification of Chest X-ray Images", Medical Technologies Congress (TIPTEKNO-2020), DOI:10.1109/TIPTEKNO50054.2020.9299232.

[8] Dewa Made Sri Arsa, Anak Agung Ngurah Hary Susila, "VGG16 in Batik Classification based on Random Forest", International Conference on Information Management and Technology (ICIMTech-2019), DOI:1109/ICIMTech.2019.88-43844.

[9] Shyava Tripathi, Rishi Kumar, "Image Classification using Small Convolutional Neural Networks", 9th International Conference on Cloud Computing, Data Science & Engineering (CONFLUENCE-2019), ISBN: 978-1-5386-5934-2, DOI:1109/CONFLUENCE.2019.8776982.

[10] P Arti, Abhishek Agrawal, A Adishesh, V M Lahari, KrupaB Niranjana, "Convolutional Neural Network Models For Content Based X-Ray Image Classification", IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE-2019), ISBN:978-1-7281-4500-6, DOI:1109/WIECON-ECE48653.20-19.9019943.

[11] Arpana Mahajan, Sanjay Chaudhary, "Categorical Image Classification Based on Representational Deep Network(RESNET)", 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA-2019), DOI:1109/ICECA.2019.8822133.

[12] Rui Yang, Yun Zhang, Zhenyuan Ji, Weibo Deng, "MSPPF-Nets: A Deep Learning Architecture for Remote Sensing Image Classsification", IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS-2019), DOI:1109/IGARSS.2019.8899068.

[13] Emine Cengil, Ahmet Cinar, "Multiple Classification of Flower Images using Transfer Learning", International Artificial

intelligence and Data Processing Symposium(IDAP-2019), DOI:10.1109/IDAP.2019.8875953.

[14] Yian Seo, Kyung-shik Shin, “Image Classification of Fine-grained Fashion Image Based on Style Using Pre-trained Convolutional Neural Network”, IEEE 3rd International Conference on Big Data Analysis (ICBDA-2018), DOI: 10.1109/ICBDA.2018.8-367713.

[15] Qiang Li, Yingjan Yang, Yingwei Guo, Wei Li, Yang Liu, Yan Kang, “Performance Evaluation of Deep Learning Classification network for Image Features”, IEEE Access-2020(Volume:9),DOI:10.1109/ACCESS.2020.3048956.

[16] Minjun Zhu, Qinghua Chen, “Big Data Image Classification Based on Distributed Deep Representation Learning Model”, IEEE Access-2020, (Volume:8), Page(s):133890 – 133904, ISBN:978-1-7281-7336-0, DOI:1109-/ACCESS.2020.3011127.