PERFORMANCE ANALYSIS OF CNN APPROACHES FOR IMAGE CLASSIFICATION OVER MNIST DATASET

[1]Karan Sharma, [2]Anita Ganpati
[1]M.Tech. Student, [2]Professor
Department of Computer Science,
Himachal Pradesh University, Shimla, India

Abstract: *Deep Learning (DL) helps in learning from experiences and understanding the real world in terms of hierarchy. Image Classification is the key application of the deep learning. Using deep learning, system can recognize objects, faces or scene like a human eye and performs much better. Image Classification can be achieved by machine learning or deep learning. Unlike machine learning, there is no need of feature selection and feature extraction in deep learning. While using machine learning, after a certain extent of training model reached a saturation point whereas this is not a case for deep learning. So, utilization of deep learning for image classification is more effective than other approaches. In deep learning, for image classification Convolutional Neural Networks (CNNs). This study is carried out to evaluate the performance of different CNN approaches for image classification over MNIST dataset The MNIST dataset is a collection of handwritten digit images, contains 10000 testing images and 60000 training images. These models are evaluated in python using Keras and TensorFlow libraries.*

Keywords **– Deep Learning, Convolutional Neural Networks, Image Classification, MNIST, TensorFlow, Keras.**

I. INTRODUCTION

1.1 DEEP LEARNING

Deep learning is a sub-field of machine learning that is based on learning from the experience and understanding of the real world in terms of hierarchy. In deep learning, the learning can be supervised or unsupervised. Deep learning works like the brain. Its models are almost identical to the nervous system of the human brain [1]. A deep learning model works in the layered structure. Each layer is connected to the another and passing information from previous to another one. Each layer in deep learning learns to translates its input data into a bit more abstract and compound description. High dimensional data can be processed with deep learning [2]. Unlike machine learning, there is no saturation point in deep learning. So, deep learning models perform well. Also, in machine learning feature extraction is performed by the human, but deep learning extracts the feature itself.

1.2 IMAGE CLASSIFICATION

Image Classification is a process of predicting the image on the basis of some class of data whose class labels are unspecified. Image classification can be supervised or unsupervised [5]. Image classification includes a process that helps analyzing a dataset by generating some rules for grouping which are further used for classifying the future unseen data. There is generally a two-step process is used in image classification:

1. Learning: In this step, a model is built by describing some of the predetermined set of data classes. This step is called as training step, in which an algorithm is implemented to build a model by analysis of dataset from a training dataset which consist of images and labels.
2. Classification: In this step, the extracted model from the learning step is tested with the whole new test dataset for the sake of measuring and analyzing the performance of extracted trained model.

1.3 CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural network is a subset of deep neural network which is generally used in computer vision and automatic speech recognition (ASR) [4]. It consists of input, output and multiple hidden layers. Typically, the hidden layers of convolutional neural network contain convolution layer, fully connected layer, pooling layer and normalization layer. Convolutional neural network recognizes the images by transfiguring the original image data through multiple layers [6]. Convnets are identical to visual cortex of human body in working. Each hidden layer detects a particular feature set like edges, lines or a pattern. The higher layers detect much complex features. Convnets are widely used for classification of images, detection of objects, facial recognition and image recognition [3] [10].

II. EXPERIMENATAL SETUP

2.1 Methodology Followed

Different convolutional neural networks approaches are implemented for analysis of performance over MNIST dataset [9]. The parameters used for performance evaluation are accuracy and loss. Also, precision, f1 score and recall is calculated for each model.

2.2 Dataset Used

MNIST (Modified National Institute of Standards and Technology) dataset is used in this study. The MNIST dataset is a collection of handwritten digit images, contains 10000 testing images and 60000 training images. MNIST is a subpart of NIST dataset [9].

2.3 Tools Used

For this evaluation models are implemented in python programming language using Keras and Tensorflow libraries. TensorFlow is an opensource machine learning library by google [7] and Keras is an opensource library for neural networks [8].

2.4 Cases

Case 1: In this case, the model consists of 3 convolution layers, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The total number of trainable parameters are 93322.

Case 2: In this case, the model consists of 3 separable convolution layers, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The total number of trainable parameters are 44787.

Case 3: In this case, the model consists of 3 transpose convolution layers, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The total number of trainable parameters are 466058.

Case 4: In this case, the model consists of 1 convolution layer, 1 separable convolution layer, 1 transpose convolution layer, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The total number of trainable parameters are 241066.

Case 5: In this case, the model consists of 3 transpose convolution layers, 3 dropout layers, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The dropout rate is 0.1. The total number of trainable parameters are 466058.

Case 6: In this case, the model consists of 3 transpose convolution layers, 3 dropout layers, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The dropout rate is 0.05. The total number of trainable parameters are 466058.

Case 7: In this case, the model consists of 3 transpose convolution layers, 3 dropout layers, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The dropout rate is 0.025. The total number of trainable parameters are 466058.

Case 8: In this case, the model consists of 3 transpose convolution layers, 3 dropout layers, 2 max-pooling layers, 1 flatten layer and 2 dense layers. The dropout rate is 0.2. The total number of trainable parameters are 466058.

III. RESULTS AND DISCUSSION

In the training phase, it is observed that after 6 epochs, case 7 gives the highest accuracy. The case 7 has dropout rate of 0.025. In terms of loss case 3 and case 7 gives the lowest loss rate after 6 epochs.

3.1 Training Accuracy Result

Fig. 4.1 and Table 4.1 depicts the training accuracy results.

Table 4.1: Training Accuracy Result

Epochs	1	2	3	4	5	6
Case 1	0.9554	0.9849	0.9895	0.9923	0.9937	0.9950
Case 2	0.9153	0.9737	0.9808	0.9846	0.9871	0.9885
Case 3	0.9610	0.9873	0.9907	0.9938	0.9941	0.9955
Case 4	0.9444	0.9848	0.9887	0.9913	0.9933	0.9944
Case 5	0.9570	0.9857	0.9898	0.9921	0.9934	0.9944
Case 6	0.9606	0.9865	0.9909	0.9927	0.9940	0.9951
Case 7	0.9622	0.9871	0.9905	0.9932	0.9946	0.9958
Case 8	0.9597	0.9857	0.9890	0.9908	0.9927	0.9932

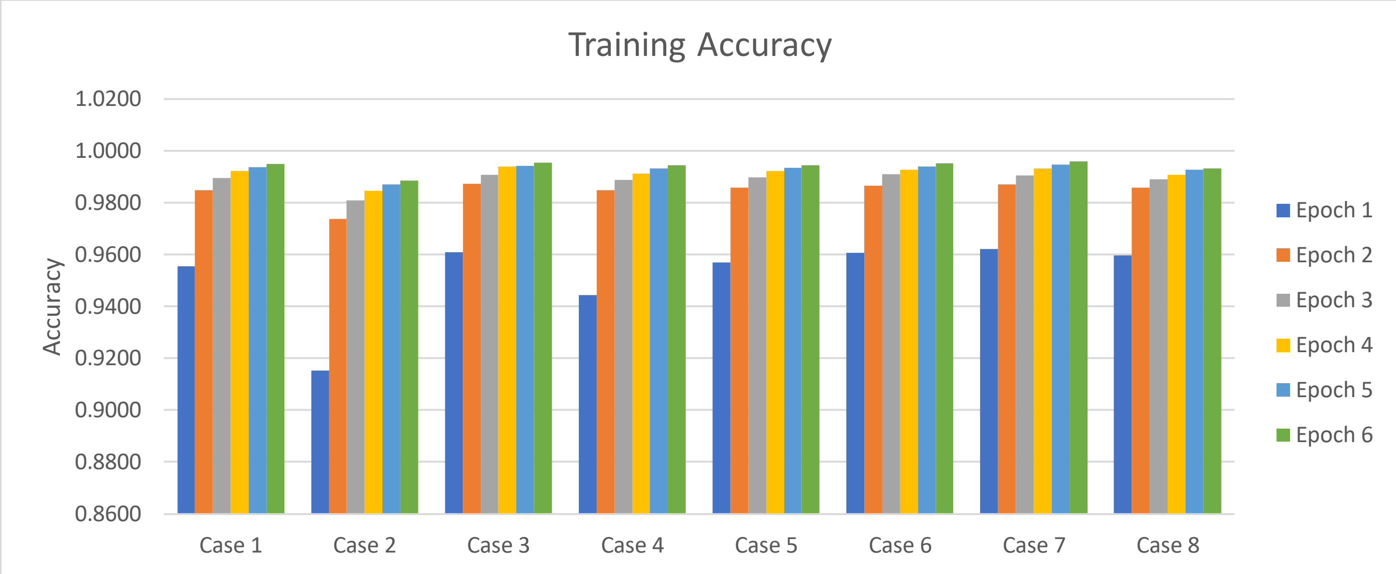

Fig. 4.1: Graphical Representation of Training Accuracy

3.2 Training Loss Result

Fig. 4.1 and Table 4.1 depicts the training loss results.

Table 4.2: Training Loss Result

Epochs	1	2	3	4	5	6
Case 1	0.1455	0.0479	0.0335	0.0253	0.0194	0.0153
Case 2	0.2712	0.0816	0.0596	0.0485	0.0407	0.0352
Case 3	0.1251	0.0418	0.0294	0.0201	0.0194	0.0136
Case 4	0.1746	0.0490	0.0353	0.0270	0.0214	0.0174
Case 5	0.1342	0.0459	0.0331	0.0249	0.0208	0.0175
Case 6	0.1274	0.0429	0.0286	0.0237	0.0189	0.0145
Case 7	0.1231	0.0427	0.0288	0.0223	0.0164	0.0137
Case 8	0.1296	0.0469	0.0335	0.0292	0.0228	0.0214

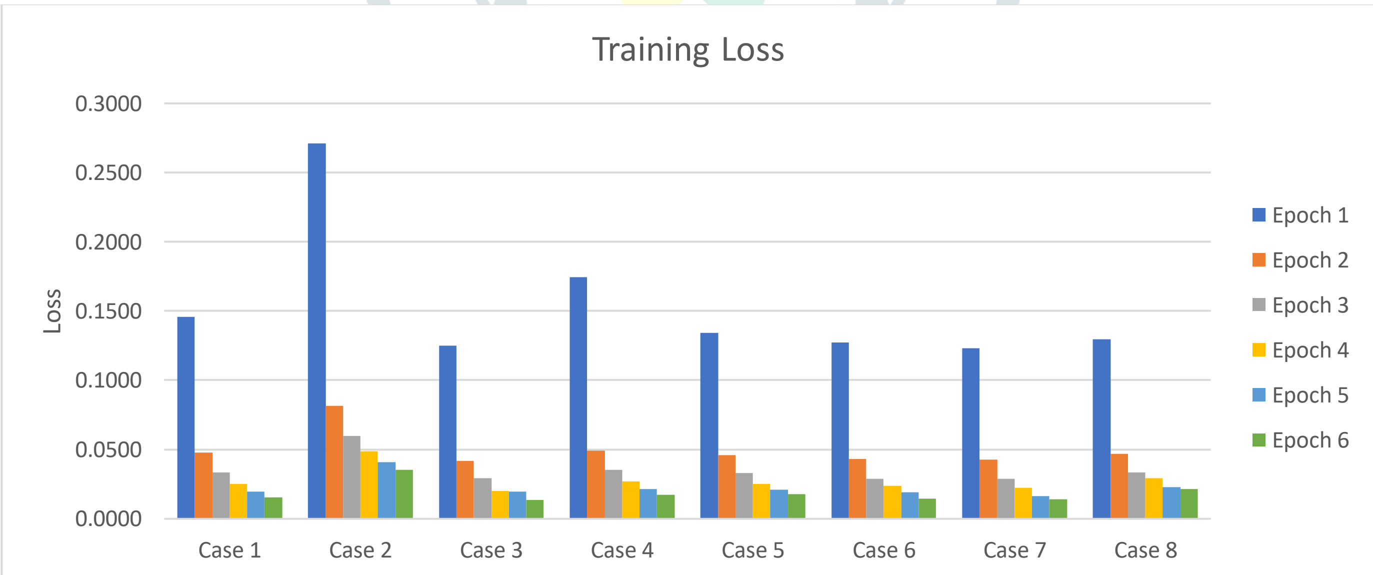

Fig. 4.2: Graphical Representation of Training Loss

3.3 Testing Results

Fig. 4.3, 4.4 and Table 4.3 depicts the testing accuracy and loss results. In testing results, it is observed that, case 8 performs better with highest accuracy of 0.9923. Case 8 has dropout rate of 0.2. Case 3 has second highest accuracy. Also, case 8 has the lowest testing loss rate.

Table 4.3: Testing Accuracy and Loss Result

Case	Testing Accuracy	Testing Loss
Case 1	0.9911	0.0311
Case 2	0.9859	0.0454
Case 3	0.9921	0.0287
Case 4	0.9915	0.0276
Case 5	0.9920	0.0266
Case 6	0.9878	0.0426
Case 7	0.9917	0.0292
Case 8	0.9923	0.0248

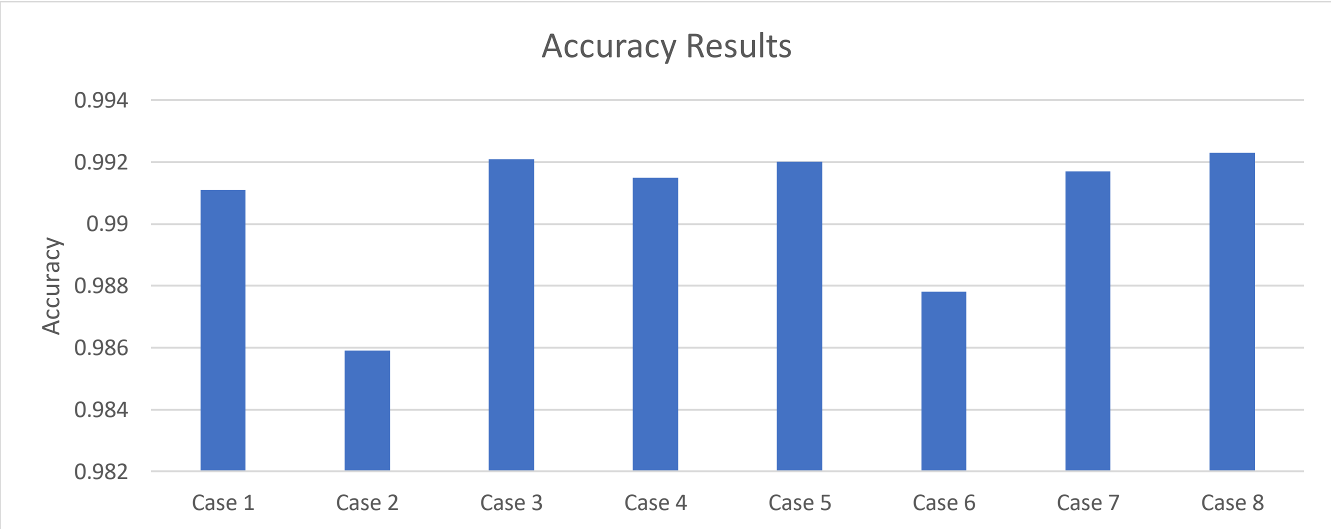

Fig. 4.3: Testing Accuracy Graph

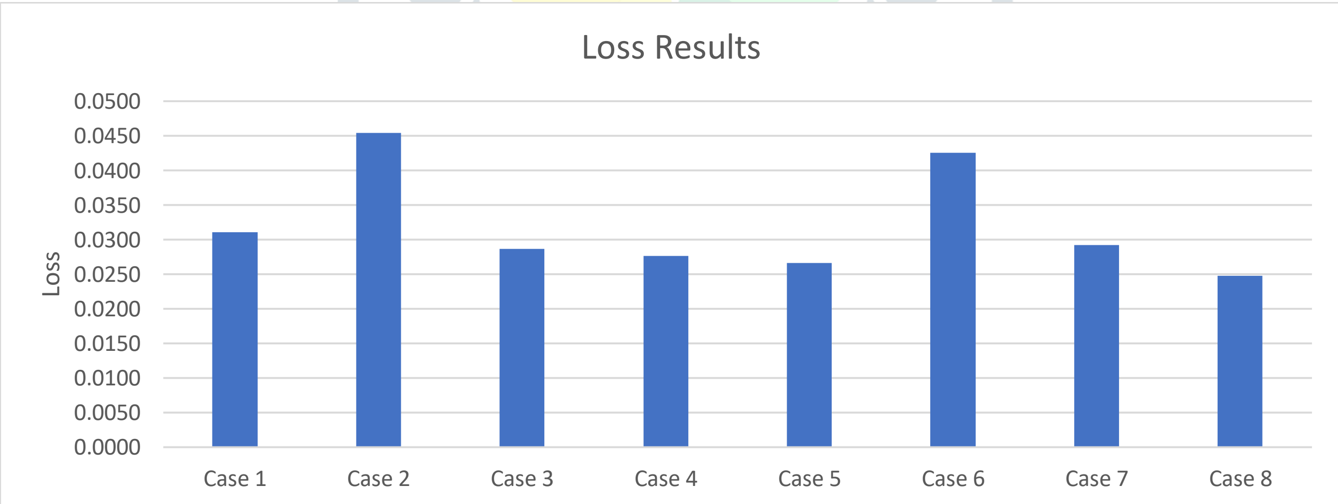

Fig. 4.4: Testing Loss Graph

From the overall result, it has been observed that, case 8 has performed better overall with the highest testing accuracy rate and the lowest loss rate. Case 8 has dropout rate of 0.2 and has transpose convolution layer instead of convolution layer. Although the training accuracy is not the highest in case 8 but in terms of trained model testing accuracy is much important than training accuracy.

3.4 Results for Case 8

Fig. 4.5 and Table 4.4 depicts the results for the case 8. The results included the confusion matrix and Classification report. The classification report shows the precision, recall and F1-score. From the values of these metrics we can clearly demonstrate that the results for case 8 were better than the other cases.

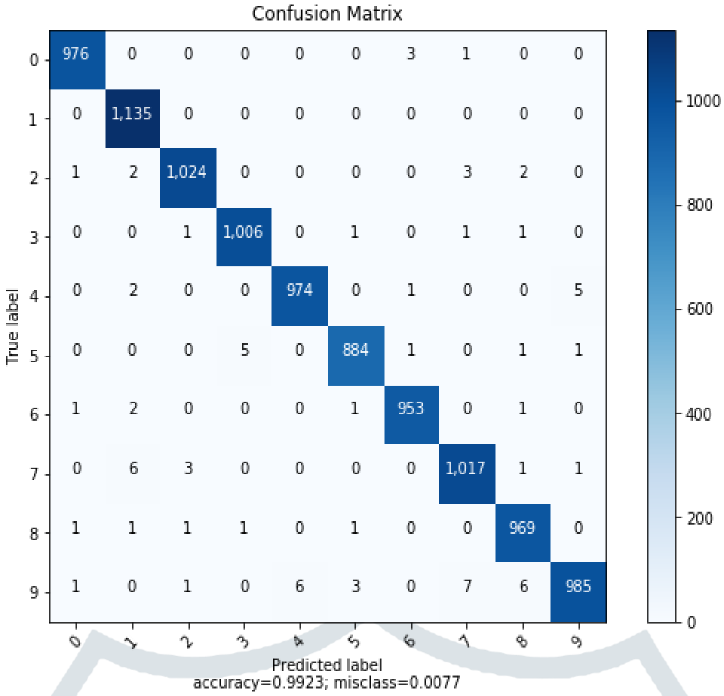

Fig. 4.5: Confusion Matrix for Case 8

Table 4.4: Classification Report of Case 8

Classification Report			
Classifier	**Precision**	**Recall**	**F1-Score**
0	1	1	1
1	0.99	1	0.99
2	0.99	0.99	0.99
3	0.99	1	1
4	0.99	0.99	0.99
5	0.99	0.99	0.99
6	0.99	0.99	0.99
7	0.99	0.99	0.99
8	0.99	0.99	0.99
9	0.99	0.98	0.98
Weighted Average	0.99	0.99	0.99

IV. CONCLUSION AND FUTURE SCOPE

In this paper, the classification of images has been carried out on MNIST dataset. The CNN has been used to carry out these experimentations with different approaches. Total eight cases have been created using combination of different layers. The experiments are employed for each of these cases. It has been found that,

- The performance of prediction model for image classification increases when using the transpose convolution layer instead of traditional convolution layer.
- When we use the drop-out layer with transpose convolution layer the performance of our model has been further enhanced.

So, it can be concluded that, the performance of our proposed prediction model for image classification works better for the MNIST dataset. For future work, the proposed model can be used on the different datasets with variable drop-out rate and number of epochs.

REFRENCES

[1] Jürgen Schmidhuber, “Deep learning in neural networks: An overview”, Neural Networks, Volume 61, January 2015, pp. 85-117.

[2] Grégoire Montavon, Wojciech Samek & Klaus-Robert Müller, “Methods for interpreting and understanding deep neural networks”, Digital Signal Processing, Volume 73, February 2018, pp. 1-15.

[3] Q.V. Le, A tutorial on deep learning part 2: autoencoders, convolutional neural networks and recurrent neural networks. Google Brain, 2015, pp.1-20.

[4] A Krizhevsky., I. Sutskever & G.E. Hinton, “Imagenet classification with deep convolutional neural networks”, In Advances in neural information processing systems, 2012 ,pp. 1097-1105.

[5] Haralick, Robert M., and Karthikeyan Shanmugam. "Textural features for image classification." IEEE Transactions on systems, man, and cybernetics 6 (1973): 610-621.
[6] Oquab, Maxime, et al. "Learning and transferring mid-level image representations using convolutional neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
[7] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016.
[8] Chollet, F., & others. (2015). Keras.
[9] The MNIST Dataset available at: http://yann.lecun.com/exdb/mnist/
[10] https://www.tensorflow.org/beta/tutorials/images/intro_to_cnns