International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 07 | Jul 2023 www.irjet.net p-ISSN: 2395-0072

Classification of Images Using CNN Model and its Variants

Narayan Dhamala[1], Krishna Prasad Acharya[2]

[1]*Teaching Assistant, Department of Computer Science & Application, Mechi Multiple Campus, Jhapa, Nepal*
[2] *Assistant Professor, Department of Computer Science & Application, Mechi Multiple Campus, Jhapa, Nepal*

Abstract - Image classification is a method of assigning a label to an image and it is suitable to use deep learning for this task due to spatial nature of image which can leverage the massively parallel structure to learn various features. In this research, a Convolution Neural Networks (CNN) model is presented with three configurations. The first configuration is simple and other two configurations are improvement of first configuration by using techniques to prevent over fitting. The training and testing is performed on a CIFAR-10 dataset which consists of 60000 sets of images of 10 different objects. During comparison of variants of model using different performance matrices it is observed that the dropout regularization technique can significantly make the model more accurate. It also shows that lower batch size can give better result than higher batch size.

Key Words: ***Convolution Neural Network, Epos, Pooling, relu, softmax.***

1.INTRODUCTION

Image classification is considered as an intelligent task of classifying images into different classes of given objects based on features. The classification problem can be binary or multiple classifications. Examples of binary classification are classifying between cat or dog images, absence or presence of cancer cells in the medical images etc. Similarly, multiple classifications include classifying cat or dog images, different animals, digit recognition etc. Image classification is used in the field of computer vision for analyzing various image data to get useful insight. It can differentiate between the given images based on tiny details which could be missed even by expert humans in the given domain It is often misunderstood with object recognition. Object recognition is a boarder term which is a combination of computer vision tasks including image classification to detect and recognize different objects in the image. So, the main difference is that image classification only deals with classifying images into different types or classes while object recognition involves detection of various objects in the image and recognizing them. Image classification involves training the machine learning model using large data sets of images. The model learns the pattern and features present in various classes of objects and can predict the class of object from previously unseen image.

1.1 Convolution Neural Network (CNN)

CNN is a type of Deep Neural Networks which is mainly used for solving image recognition related problems. It a variant of Multi-Layer Perceptron (MLP) also called fully connected neural networks. In such networks, each layer in one layer is connected to every neuron in next layer. CNNs can form complex patterns from simpler ones which makes them more effective in recognition of images with lots of features. In this CNN architecture, the three different layers are present which are: convolution layer, pooling layer and fully connected layer.

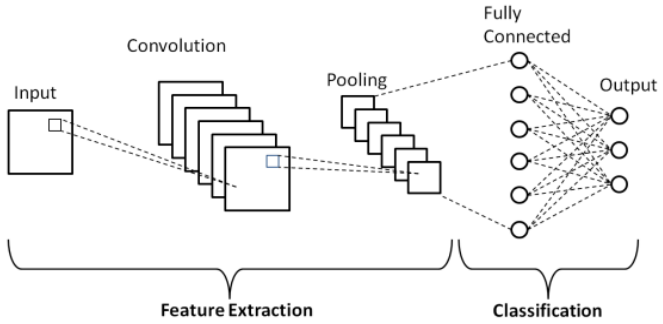

Figure 1: Basic architecture of a CNN

1.1.1 Convolutional layer

It is first and most significant layer. This layer is responsible for learning features from the input image. It takes image as input and applies a kernel (filter) to an image and produces the output. This operation is called convolution. It helps to reduce shape of the image while retaining its features. Different filters can be stacked together to extract many features. In CNN, an image with a shape (no. of images) x (image height) x (image width) is passed through Convolution layer which in turn produces a feature map of shape (no. of images) x (feature map height) x (feature map width) x (feature map channels).

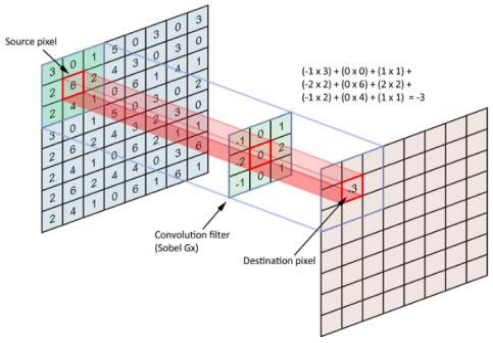

Figure 2: Convolution operation

1.2.2 Pooling layer

It is related to down sampling the feature maps by summarize -ng the presence of features in patches of the feature map. The pooling is used to further reduce the size of the feature map. The size of pooling filter is always in the form of 2 by 2 matrixes with a stride of 2 pixels.

The two mostly used pooling methods are: max pooling and average pooling. In max pooling, the 2 by 2 filter is slided over the feature map and find out the maximum value in the box. In an average pooling, the pooling filter of size 2 by 2 is slided over the feature map and the average value in the box is chosen.

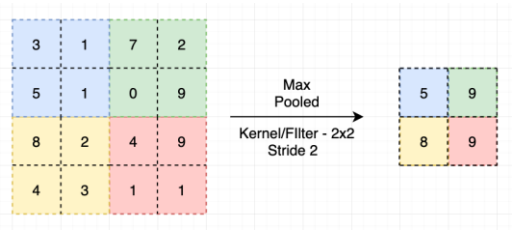

Fig 1-3: Max-pooling operation

1.2.3 ReLU layer

It is a rectified linear unit which removes negative values from activation map by setting them to zero. It increases non-linear properties of the network. It applies the activation function f(x) = max (0, x). It helps to overcome the limitations of other activation functions like sigmoid and tanh. Layers deep in large networks using these nonlinear activation functions fail to receive useful gradient information.

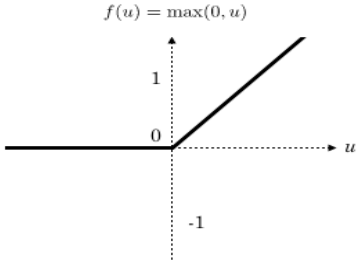

Figure 4: ReLU activation function

1.2.4 Fully Connected layer

This layer takes the output produced from convolution and pooling method to classify the input image into different classes (labels). The result from previous layer is flattened into a single vector of values, each representing a probability that a certain feature belongs to a label. For example, in handwriting recognition system the labels can be letters from A to Z. If the input image is of letter 'P' then the probabilities of features representing circle and a line should be high.

1.3 Problem statement

The different machine learning algorithms used in detection and recognition of objects are: Support Vector Machines (SVM), K-Nearest Neighbor (KNN), Naïve Bayes (NB), Binary Decision Tree (BDT) and Discriminant Analysis (DA). Most of these traditional machine learning algorithms are not suitable for problems which are non-linear in nature. They can often provide inaccurate model and suffer from high error rate. So, for solving such problems a type of machine learning technique called deep learning is becoming more popular. Although neural networks were developed very early, they were not explored extensively due to lack of sufficient computing power. Due to advancements in computer hardware and rise of accelerated processing units such as GPU and TPU such neural networks are being used to achieve greater performance in various intelligent tasks. Similarly, it can be combined with traditional techniques to make them more powerful. It massively parallel network similar to the biological network of neurons. Due to the spatial nature of images, they can be more efficiently processed using neural networks. It becomes harder for the neural networks to find underlying patterns of features by analyzing each and every pixel in the image. So, this simple way of using Neural Networks cannot perform well in the high-quality images having lots of features. So, in order to solve the above problem, a more advanced type of neural networks called CNN can be used.

1.4 Objectives

The main objectives of this research are as follows:

•To implement Convolution Neural Network model and it's variants for image classification.

•To compare and analyze the different variants of the model using different performance metrics like: classification accuracy rate, error rate, sensitivity, specificity, precision, F1 score and effect of batch size on the given image dataset.

2. Literature Review

In 2021, Ruchika Arora et.al [1] proposed an optimized CNN for image classification of Tuberculosis images using efficient tuning of hyper parameters based on hyper band search optimization approach. The tuberculosis diseases in chest X-ray images are trained using NLM china dataset and also tested on them. The efficient hyper parameters are chosen by trial and error method and according to the experience of the designer. The experiment shows that usage of hyper parameters on a given data set using CNN method achieves 91.42 % accuracy.

The Alex Krizhevsky et al.built ImageNet [2] CNN forILSVRC-2012 contest where the task was to classify 1.2 million high resolution images into 1000 different classes. The authors were able to achieve top-1 and top-5 error rates of 37.5% and 17.0 % on the test data. It is considered to be better result

than previous state-of-art. The CNN model consists of five convolutional layers some of which are followed by max pooling layers and three fully connected layers. This model uses efficient GPU implementation to make training faster. It also utilizes dropout regularization technique to reduce over fitting. This paper showed that a deep CNN is capable of achieving state of art result using purely supervised learning.

The Md. Anwar Hossain et al. [3] proposed a CNN image classification which was trained using CIFAR-10 dataset. The architecture of the CNN model consists of three blocks of Convolution and ReLU layer. The first block is followed by Max pooling, second is followed by Average pooling and last one is followed by a fully connected layer. The author has implemented this model using MatConvNet. In the experiment, the maximum accuracy of 93.47% was yielded with batch size 60, no. of epochs 300 and learning rate 0.0001.

For ImageNet challenge 2014, The Karen Simonyan et al. [4] at VGG team came up with more accurate CNN architectures which not only achieve the state-of-the-art accuracy on ILSVRC classification and localization tasks, but are also applicable to other image recognition datasets. The authors present different configuration of the network which differs only in depth from network 8 convolutional layers with 3 FC layers to 16 convolutional layers with 3 FC layers. The main highlight of this network is that it used very small filters throughout the network which helps to achieve significant improvement.

GoogLeNet[5] was able to push the limit of the CNN depth with a 22 layers structure. It is found that the deeper and wider layer helps to improve accuracy. The main hallmark of this architecture is the improved utilization of computing resources in the network. The authors were able to increase the depth simultaneously keeping the computational requirement constant by improving the architecture. The paper discusses the idea of applying dimension reduction wherever computational requirements increase. The authors have presented a type of network configuration called inception network which consists of modules stacked upon each other. Thus, this paper shows the strength of inception architecture and provides evidence that it can achieve similar result as more computationally expensive networks.

The Shivam et al. [6] performs image classification using five different architecture of image classification that are made based on varying convolution layer, fully connected layers and filter size. To perform the experiment the different hyper parameters like: activation function, optimizer, learning rate, dropout rate and size of batch are considered. The result shows that the choice of activation function, dropout rate and optimizer has influence on the accuracy of the architecture. The CNN Model gives 99% of accuracy for MNIST dataset but lower than that for Fashion-MINST dataset this may be due to complex nature of data in the Fashion-MINST dataset.

3. Methodology

3.1 Data Collection

The dataset used in this research is CIFAR-10 dataset. It is one of the most widely used dataset for machine learning. It consists of 60,000 images with a 32 x 32 color images in 10 classes, with 6,000 images per class. The ten different classes in the datasets are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Since the resolution of the image is low, it can be used to test new machine learning algorithms quickly with very less computing power.

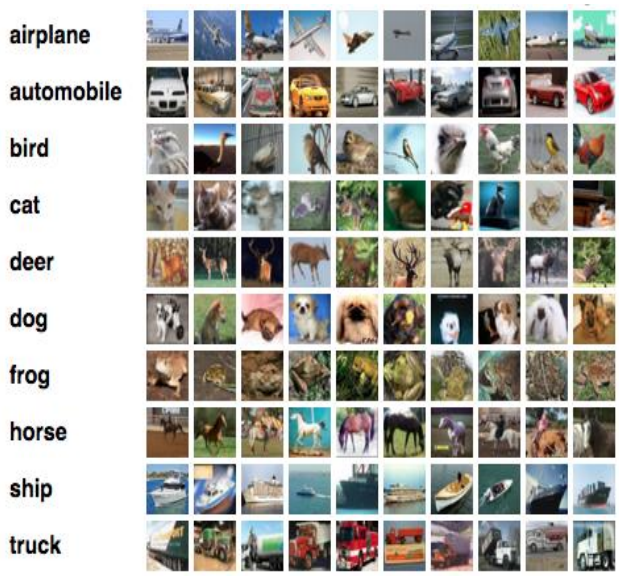

Figure 5: Some images on CIFAR-10 dataset

3.2 Tools used

Following are the hardware and software tools used to implement this research

3.2.1 Hardware requirements

CPU: Intel x64_86
RAM: 4GB
HDD: 1 TB

3.2.2 Software requirements

Operating System: Windows 10 64bit
Developed In: Python 3.7
Libraries Used: Keras, Scikit-Learn, Matplotlib, Seaborn

3.3 Data preprocessing

It is necessary to apply some preprocessing on the data before providing it into the network. The dataset of 60,000 images is split into 40,000 training set, 10,000 for validation

during training and 10,000 for testing. The pixel data in x values is converted into floating point value and normalized into range 0-255. The output or y value consists of categorical data which are converted into numeric values.

3.4 Architecture of CNN model

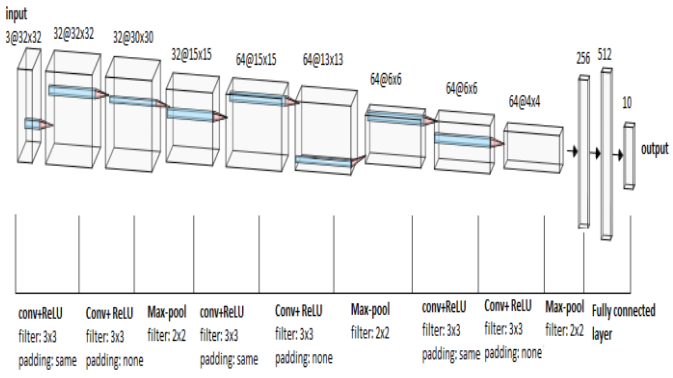

Figure 6: The architecture of CNN Model

The network of the model is structured in 3-block VGG style having sequence of CONV-CONV-POOL-CONV-CONV-POOL-CONV-CONV-POOL-FC-FC. Each block consists of two convolution layer (CONV) followed by a max pooling layer (POOL). There are two fully connected layers (FC) at the end of the network.

.

The first layer of the network is a convolutional layer in which the input images are fed. Each image is of size n*n=32*32 with depth of 3 for each color channels. It uses f*f=3*3 kernel for convolution operation. It uses ReLU (Rectified LinearUnit) activation function. The stride is 1 which means the filter window is moved by 1 pixel at a time. since our 32*32 image is reduced to 30*30, zero padding around input layer is applied such that output image size is same as input. In this case padding=1 is applied. Next, 32 filters are applied in this layer. This layer produces feature maps of size 32@32*32 where first 32 is no. of feature maps and 32 in 32*32 is given as: ((n+2p-f)/s) +1. So, in this case, ((32+2*1-3)/1) +1 =32. The no. of feature maps in next layers can be calculated using same formula.

The second layer is also a convolutional layer which takes input from previous layer and produces feature maps of size 32@30x30. It also applies 32 filters of 3*3 size. It also has stride=1 but it doesn't apply any zero padding.

The third layer is pooling layer which applies max-pooling using 2*2 filter with stride=2. It produces output feature maps of size 32@15*15. This layer doesn't apply any activation function.

The fourth layer is a convolutional layer similar to first layer. It applies 64 filters of size 3x3and produce output feature maps of size 64@15*15. It has stride, padding and activation function same as first layer.

The fifth layer is a convolution layer same as second layer. It produces feature maps of size 64@13*13. Like second layer, it doesn't have padding around boundary. It has stride, padding and activation function same as second layer.

The sixth layer is pooling layer same as third layer. It also applies max-pooling using 2x2 filter with stride=2. It produces output feature maps of size 64@6*6.

Next three layers are similar as fourth, fifth and sixth layers respectively. The seventh layer which is convolutional layer produces feature maps of size 64@6*6 which is fed into next convolutional layer that outputs feature maps of 64@4*4. Next, the ninth layer which is pooling layer produces output of size 64@2*2.

The next layer is a fully connected layer which converts the 3D array into a 1D array of size 2*2*64 = 256. It is also called flattened layer. It uses ReLU activation function. Finally, the last layer is also a fully connected layer which has 10 nodes for representing each class in CIFAR-10 dataset. It uses soft-max activation function.

3.5 Variants of the model

3.5.1 Baseline model

This model is built using the architecture presented in previous section. It is a 3 block VGG style model which is modular and easy to implement. It uses Stochastic Gradient Descent (SGD) for optimizing model with learning rate of 0.001. This variant of the model doesn't use any techniques for the improvement of the model. It is trained using batch size of 32 and 64 and no. of epochs 100. Batch size refers the total number of training examples used before the model is updated. The number of epos is defined as the numbers of times the algorithms will work in the given training datasets. Generally the higher number of epos is chosen and can later reduce it to a number at which optimal performance is achieved. Higher number of epochs can lead to over-fitting

3.5.2 Improved model I

This model uses same architecture as baseline model but it applies a technique called dropout regularization. This technique randomly removes certain nodes from the network. It helps to prevent the condition of over fitting in our model which causes the model to memorize or fit so closely to training data that it performs poorly on unseen or test dataset. This model applies dropout rate of 25% after each block CONV-CONV-MAX and 5% before final fully connected layer.

3.5.3 Improved model II

This model further improves the previous model by applying image augmentation technique which augmentation has been used which increases size of training data by adding more images obtained by transforming training images. In this case the height and width of image is shifted randomly between 1-10%.

3.6 Comparative Criteria

The variants of the model will be analyzed based on the following performance metrics.

3.6.1 Classification accuracy rate

$$\text{Classification accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Where, True Positives (TP) = actual positives which are predicted positive

True Negatives (TN) = actual negatives which are predicted negative

False Positives (FP) = actual positives which are predicted negative

False Negatives (FN) = actual negatives which are predicted positive

3.6.2 Error rate

$$\text{Error rate} = \frac{(FP + FN)}{(TP + TN + FP + FN)}$$

3.6.3 True positive rate (TPR) or Sensitivity or Recall

$$TPR = \frac{TP}{(TP + FN)}$$

3.6.4True negative rate (TNR) or Specificity

$$TNR = \frac{TN}{(TN + FP)}$$

3.6.5 Positive Predictive Value (PPV) or Precision

$$\text{PPV or Precision} = \frac{TP}{(TP+FP)}$$

3.6.6F_1Score

For any general value β:

$$F_\beta = \frac{(1+\beta 2)\,(PPV * TPR)}{(\beta 2 * PPV + TPR)}$$

For β=1:

$$F_1 \text{ Score} = \frac{(PPV * TPR)}{(PPV + TPR))}$$

4. Result, Analysis and Comparison

In this section, the variants of the CNN model are compared along with other training parameters like batch size and no. of epochs. The comparison is done based on performance matrices: classification accuracy rate, error rate, TPR and TNR. The Table 4 shows performance metrics of variants of the model based on batch size 32 and 64. The variants of the model were trained and tested up to 100 epochs. For each configuration of model, the performance metrices has been presented.

Table -1: Performance metrics of different variants of model

Variant of model	Batch size	No. of epochs	Classification accuracy rate (in %)	Error rate (in %)	TPR (in %)	TNR (in %)	PPV (in %)	F1 Score (in %)
Baseline model	64	100	94.104	5.896	70.520	96.724	70.445	35.232
	32	100	94.568	5.432	72.840	96.982	73.073	36.406
Improved model I	64	100	96.528	3.472	82.640	98.071	82.961	41.320
	32	100	96.684	3.316	83.420	98.158	83.466	41.681
Improved model II	64	100	96.490	3.510	82.450	98.050	82.448	41.125
	32	100	96.986	3.014	84.930	98.326	85.267	42.394

4.1 Comparison result of classification accuracy rate

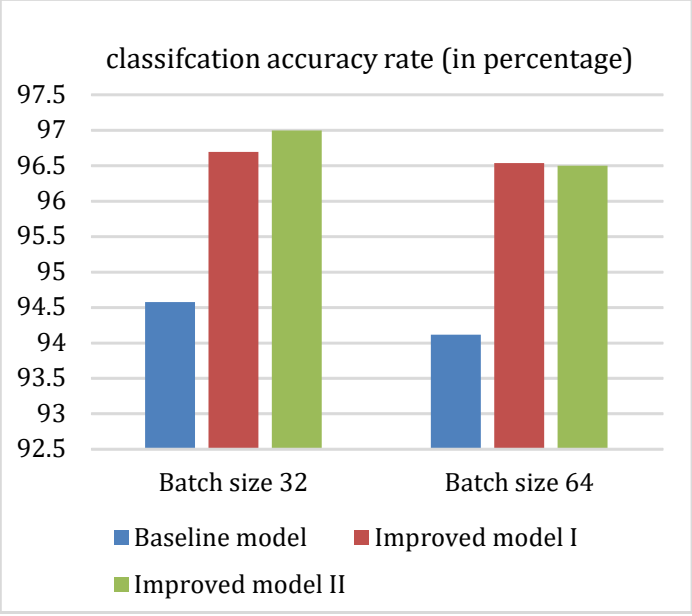

Figure 7: Comparison graph of classification accuracy rate

The graph on figure 7 shows that the accuracy rate is higher for model with lower batch size i.e. 32 in comparison with higher batch size i.e. 64 for first two variant. In slightly decreases on last variant for batch size 64.For batch size 32, the rate was increased by 2.116 % from baseline to improved model I. Similarly, it was further increased by 0.302% in improved model II. For batch size 64, the rate was increased by 2.424% from baseline to improved model I. Similarly, it was further increased by 0.038% in improved model II.

4.2 Comparison result of error rate

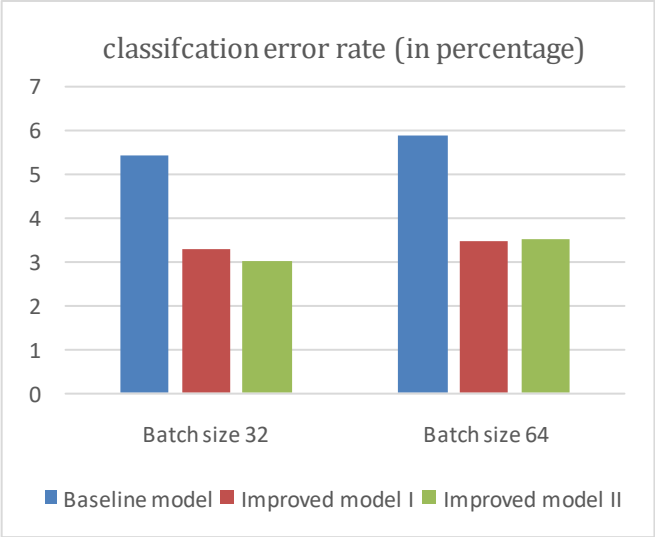

Figure: 8 Comparison graph of classification error rate

Based on figure: 8, it is clear that the second variant of the model significantly decreases the error rate and it decreases further for batch size 32 but increases slightly for batch size 64 in final variant.

4.3 Comparison result of True Positive Rate

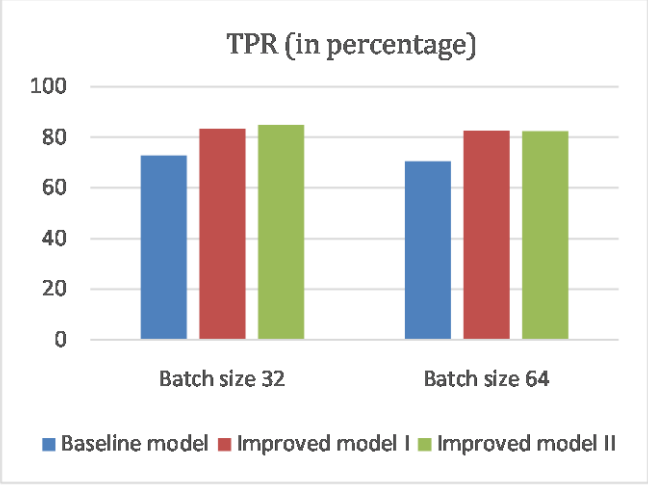

Figure 9: Comparison graph of TPR

Based on graph on above image, the TPR increases from first variant to final except when batch size is 64. It increases by 10.58 % from baseline model to improved model I and again slightly increases by 1.51 % in Improved model II when batch size is 32. For batch size 64, It increases by 12.12 % from baseline model to improved model I but slightly decreases by 0.19 % in Improved model II.

4.4 Comparison result of True Negative Rate

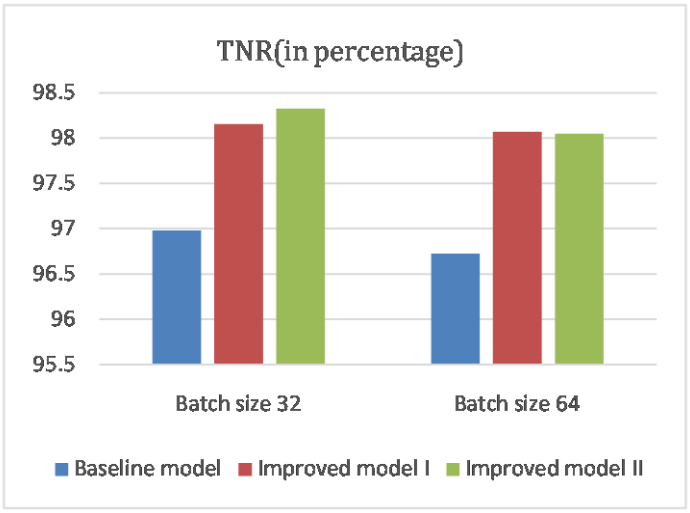

Figure 10: Comparison graph of TNR

The graph on the figure 10 shows that the TNR increases in each successive variant by 1.176 % and0.168 % when the batch size is 32. In case of batch size 64, it increases by 1.347 % from first to second variant but decreases by 0.021 % on third variant.

4.5 Comparison result of Positive Predictive Value

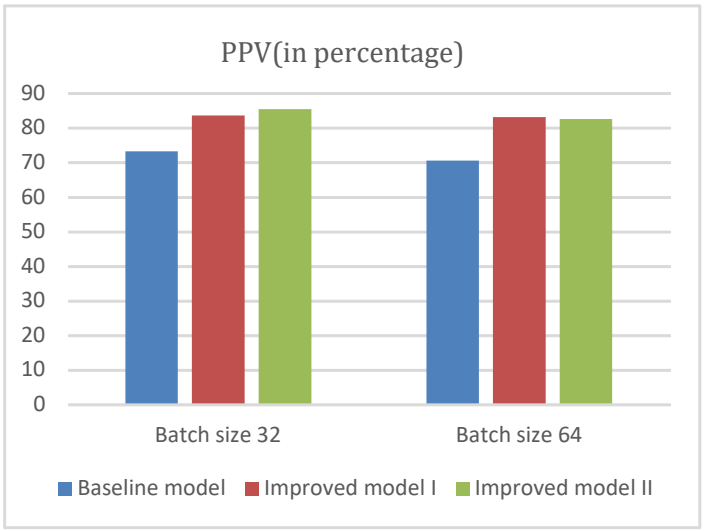

Figure 11: Comparison graph of PPV

The graph on the figure 11 shows that the PPV increases in each successive variant by 10.393 % and further 1.801 % when the batch size is 32. In case of batch size 64, it increases by 12.516 % from first to second variant but decreases by 0.513 % on third variant.

4.6 Comparison result of F_1 score

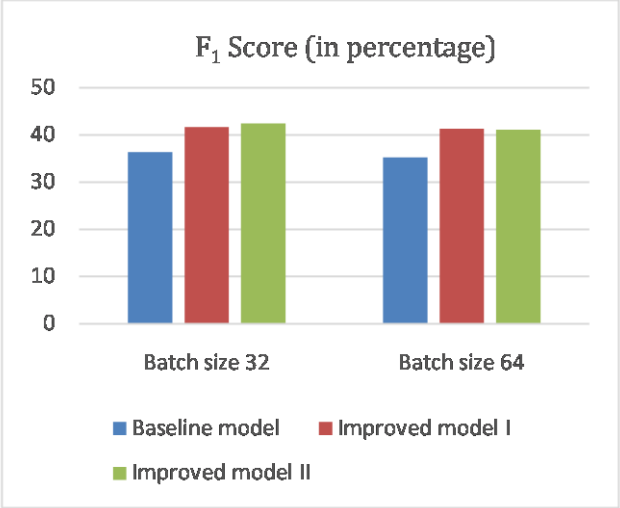

Figure 12: Comparison graph of F_1 Score

Based on the figure 12, the F1 score increases by 5.275 % in improved model I and again by 4.287 % on Improved model II when the batch size is 32. In case of batch size 64, it increases by 6.088 % from first to second variant but decreases by 0.195 % on third variant.

5.Conclusion

In this research, a CNN model is built with three variation or configuration. It is tested by solving image classification problem using CIFAR-10 dataset. The first variant is baseline model and two other improved variants use techniques like image augmentation and dropout regularization. The result from this research shows that the dropout technique greatly increased the performance of the model while that of image

augmentation slightly improved the model. It also shows that the performance can be decreased if some parameter is not optimal as such in improved model II when batch size is 64. This research also demonstrates that the deep learning can be applied to solve vision related tasks. It can be also concluded that the parameters of the model should be tweaked and tested using hit and trial method to get optimal result from the network and lower batch size gives more classification accuracy as compared to that of higher batch size.

6. Future Works

It can be further developed and extended to work on other datasets. This model can be further fine-tuned to get better results by adjusting various parameters. It can give us more accurate results by using even more datasets and deeper network. It was also realized that high performance GPU computing is necessary to train model to achieve significant results. This model can be further extended and developed to solve other more complex tasks such as classification of medical images to detect diseases like cancer, cataract, pneumonia etc. Beside this it can be developed various other image classification tasks in the future.

REFERENCES

[1] Ruchika Arora, Indu Saini and Neetu Sood, "Efficient Tuning of Hyper-parameters in Convolutional Neural Network for Classification of Tuberculosis Images," Proceedings of International Conference on Women Researchers in Electronics and Computing (WREC 2021) April 22–24, 2021, DOI: 10.21467/proceedings.114.

[2] Krizhevsky, Alex, Sutskever, Ilya, and Hinton,Geoffrey. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25 (NIPS'2012). 2012.

[3] Hossain, Md. Anwar &Sajib, Md. (2019). Classification of Image using Convolutional Neural Network (CNN). Global Journal of Computer Science and Technology. 19. 13-18. 10.34257/GJCSTDVOL19IS2PG13.

[4] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[5] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[6] Kadam, Shivam&Adamuthe, Amol & Patil, Ashwini. (2020). CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset. Journal of scientific research. 64. 374-384. 10.37398/JSR.2020.640251.