

Shreeyantra College
Tribhuvan University
Institute of Science and Technology



A Project Report On
“Image Classification using CNN”

In Partial Fulfilment of Requirements for the Bachelor Degree in Computer Science and Information
Technology

Under the supervision of:

Mr. Bishal Shrestha
Academic Coordinator

Submitted to:

Department of Computer Science and Information Technology
Shreeyantra College
Damak, Jhapa, Nepal

Submitted by:

Subash Rai (11343/073)
Pankaj Rai (11352/073)
Ashish Niroula (11338/073)
Kamal Chandra Upreti (11350/073)

Date: 27th Aug, 2021

Shreeyantra College
Affiliated to Tribhuvan University

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Computer Science and Information Technology, IOST, Shreeyantra College**, a project report entitled **“Image Classification using CNN”** submitted by **Subash Rai, Pankaj Rai, Ashish Niroula and Kamal Chandra Upreti**. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfilment of Bachelor’s degree of Computer Science and Information Technology.

Signature of the Supervisor	Signature of the HOD/Coordinator
Signature of the External Examiner	Signature of the Internal Examiner

ACKNOWLEDGEMENT

We wish to express our gratitude and appreciation to many individuals who have helped us to complete this project. This project would not have been successful without their help and support.

First of all, our sincere thanks go to **Shreeyantra college** for providing the necessary support and infrastructure needed to prepare this project.

We are grateful to our project supervisor **Mr. Bishal Shrestha** who helped us and provided the necessary guidance the project by providing ideas and knowledge needed in this project. We are really indebted to his supportive role and encouragement which really helped us throughout intense period of this project.

We are also thankful to all the teachers at our department who provided many valuable knowledge and immense support which made this project successful. We are also delighted to have very helpful colleagues and seniors for their constant support.

Lastly, we would to thank all the individuals who have directly or indirectly helped us to make this project successful.

Project Members

Subash Rai (11343/073)

Pankaj Rai (11352/073)

Ashish Niroula (11338/073)

Kamal Chandra Upreti (11350/073)

ABSTRACT

Image classification is a task 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 project, a Convolutional Neural Networks (CNN) model is presented with three configurations. The first configuration is simple and other two configurations further improve it by using techniques to prevent over fitting. The model is trained and tested on CIFAR-10 dataset which consists of 60000 sets of images of 10 different objects. During comparison of variants of model using different performance metrics 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.

Keywords: *Convolutional Neural Network, CIFAR-10 dataset, relu, softmax, keras*

Table of Contents

ACKNOWLEDGEMENT	i
ABSTRACT.....	ii
Table of Contents	iii
List of Figures	v
List of abbreviations	vi
CHAPTER 1	1
1. Introduction.....	1
1.1 Introduction to image classification.....	1
1.2 Convolutional Neural Network (CNN).....	1
1.2.1 Convolutional layer.....	2
1.2.2 Pooling layer	2
1.2.3 ReLU layer.....	3
1.2.4 Fully Connected layer	3
1.3 Problem statement.....	3
1.4 Objectives	4
CHAPTER 2	5
2. Literature Review.....	5
CHAPTER 3	7
3. Methodology	7
3.1 Data Collection	7
3.2 Tools used	7
3.2.1 Hardware requirements	7
3.2.2 Software requirements	7
3.3 Data preprocessing.....	8
3.4 Architecture of the CNN model	8
3.5 Variants of the model	9
3.5.1 Baseline model.....	9
3.5.2 Improved model I:.....	9
3.5.3 Improved model II:	9
3.6 Comparative Criteria.....	10
3.6.1 Classification accuracy rate	10
3.6.2 Error rate	10
3.6.3 True positive rate (TPR) or Sensitivity or Recall.....	10
3.6.4 True negative rate (TNR) or Specificity	10
3.6.5 Positive Predictive Value (PPV) or Precision	10

CHAPTER 4	11
4. RESULT, ANALYSIS AND COMPARISONS	11
4.1 Result, Analysis and Comparison	11
4.1.1 Comparison result of classification accuracy rate	11
4.1.2 Comparison result of error rate	12
4.1.3 Comparison result of True Positive Rate	12
4.1.4 Comparison result of True Negative Rate	13
4.1.5 Comparison result of Positive Predictive Value	13
4.1.6 Comparison result of F_1 score	14
CHAPTER 6	15
6. CONCLUSION AND FUTURE WORKS	15
6.1 Conclusion	15
6.2 Future Works	15
REFERENCES	16

List of Figures

Figure	Page
Fig 1-1: Basic architecture of CNN	1
Fig 1-2: Convolution operation.....	2
Fig 1-3: Max-pooling operation.....	2
Fig 1-4: ReLU activation function.....	3
Fig 3-1: Some images on CIFAR-10 dataset	7
Fig 3-2: Architecture of proposed CNN model.....	8
Fig 4-1: Comparison graph of classification accuracy rate.....	11
Fig 4-2: Comparison graph of classification error rate	12
Fig 4-3: Comparison graph of TPR.....	12
Fig 4-4: Comparison graph of TNR	13
Fig 4-5: Comparison graph of PPV	13
Fig 4-6: Comparison graph of F1 Score	14

List of abbreviations

CNN	Convolutional Neural Network
ANN	Artificial Neural Network
DNN	Deep Neural Network
SVM	Support Vector Machines
KNN	K-Nearest Neighbors
NB	Naïve Bayes
BDT	Binary Decision Tree
DA	Discriminant Analysis
CPU	Central Processing Unit
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
API	Application Programming Interface
CIFAR	Canadian Institute For Advanced Research
VGG	Visual Geometry Group
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
MNIST	Modified National Institute of Standards and Technology
MLP	Multi-Layer Perceptron
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent

CHAPTER 1

1.Introduction

1.1 Introduction to image classification

Image classification is considered as the intelligent task of classifying images into different classes of given objects based on features. The classification problem can be binary or multiple classification. Examples of binary classification are classifying between cat or dog images, absence or presence of cancer cells in the medical images etc. Similarly, multiple classification includes classifying cat or dog images, different animals, digit recognition etc. It is widely 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 broader 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.

1.2 Convolutional Neural Network (CNN)

CNN is a type of Deep Neural Networks mostly used for solving image recognition related problems. It is 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, there are three layers. The layers are Convolutional layer, Pooling layer and Fully connected layer.

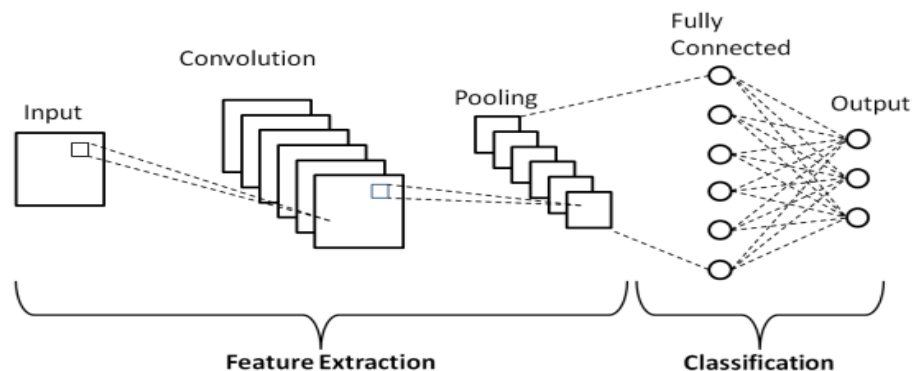


Fig 1-1: Basic architecture of a CNN

1.2.1 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 Convolutional 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).

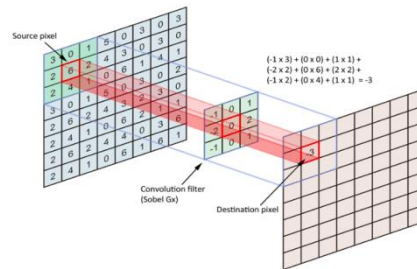


Fig 1-2: Convolution operation

1.2.2 Pooling layer

It is related to down-sampling of feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively. Pooling involves selecting a pooling operation, much like a filter to be applied to feature maps. The size of the pooling operation or filter is smaller than the size of the feature map; specifically, it is almost always 2x2 pixels applied with a stride of 2 pixels.

Max pooling, is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map. The results are down sampled or pooled feature maps that highlight the most present feature in the patch, not the average presence of the feature in the case of average pooling. This has been found to work better in practice than average pooling for computer vision tasks like image classification.

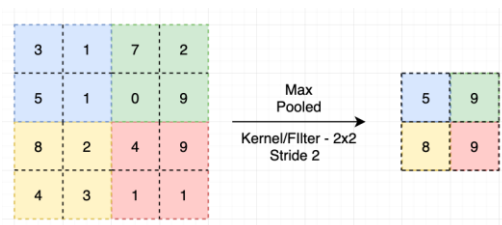


Fig 1-3: Max-pooling operation

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.

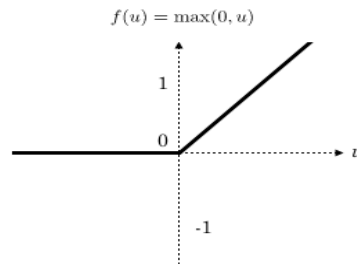


Fig 1-4: ReLU activation function

1.2.4 Fully Connected layer

This layer takes the output produced from convolution and pooling process to classify the input image into different classes(labels). The output 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

There are various machine learning algorithms used in detection and recognition of objects such as 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 state of art 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 tackle this problem, a more advanced type of neural networks called CNN can be used.

1.4 Objectives

The main objectives of this project work are as follows:

- To implement a Convolution Neural Network model 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, recall and f_1 score on the given image dataset.

CHAPTER 2

2. Literature Review

In 1998, LeNET [1] proposed by Yan LeCun et al. was able to achieve 99.05% accuracy on the MNIST which is still state of art among classification networks. The influence of the training set size was measured by training the network with 15000, 30000, and 60000 examples. It showed that even with specialized architectures such as LeNet-5, more training data would improve the accuracy. To show this, it was trained using further 540000 images obtained by randomly distorting original training images. The error rate dropped to 0.8% from 0.95% using this method. The authors compared the result using various classifiers among which Boosted LeNet-4 performed best, achieving a score of 0.7%, closely followed by LeNet-5 at 0.8%

The Alex Krizhevsky et al. built ImageNet [2] CNN for ILSVRC-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 layer 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 overfitting. 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 similar CNN to this project 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. This demonstrated again that going deeper and wider is indeed the right direction to improve accuracy. The main hallmark of this 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] introduced five CNN architectures to solve image classification problem on MNIST and Fashion-MNIST dataset. The authors present five different architectures with varying convolutional layers, filter size and fully connected layers. The experiment was conducted using various hyperparameters such as activation function, optimizer, learning rate, dropout rate and batch size. The result indicates that any CNN model can give 99% accuracy on MNIST dataset. The architecture with 3 convolutional layer and 2 fully connected layers gives better testing accuracy for fashion MNIST dataset.

CHAPTER 3

3. Methodology

3.1 Data Collection

The dataset used in this project is CIFAR-10 dataset. It is one of the widely used dataset for machine learning. It consists of 60,000 32 x 32 color images in 10 classes, with 6,000 images per class. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Since the resolution of the image is low, it can be used to test new machine learning algorithms quickly with very less computing power.

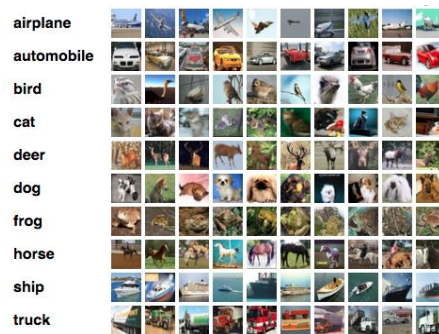


Fig 3-1: Some images on CIFAR-10 dataset

3.2 Tools used

Following are the hardware and software used to implement this project

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, NumPy, Seaborn

3.3 Data preprocessing

It is necessary to apply some preprocessing on the data before feeding into the network. The dataset of 60,000 images is split into 50,000 training set and 10,000 for validation during training as well as for testing. The pixel data in x values is converted into floating point value and normalized into range 0-255. The output or y values consists of categorical data which are converted into numeric values using one hot encoders. It adds 10 new columns in the dataset for each class. The row of the input image has value 1 in the column of corresponding class and 0 elsewhere.

3.4 Architecture of the CNN model

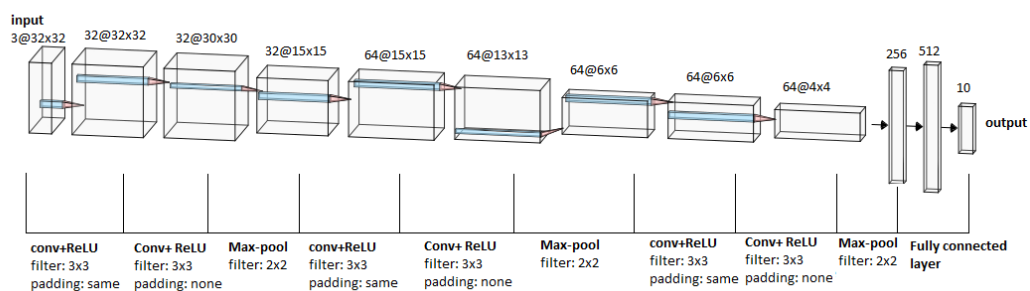


Fig 3-2: Architecture of the CNN model

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 layer (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 $n \times n = 32 \times 32$ with depth of 3 for each color channels. It uses $f \times f = 3 \times 3$ kernel for convolution operation. It uses ReLU (Rectified Linear Unit) 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 $32 @ 32 \times 32$ 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 $32 @ 30 \times 30$. 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 $32 @ 15 \times 15$. 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 3x3 and 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. The batch size is a number of samples processed before the model is updated. The number of epochs is a hyper parameter that defines the number times that the learning algorithm will work through the entire training dataset. We generally choose higher number of epochs and can later reduce it to a number at which optimal performance is achieved. Higher no. 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 overfitting 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

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

3.6.4 True Negative Rate (TNR) or Specificity

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

3.6.5 Positive Predictive Value (PPV) or Precision

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

3.6.6 F₁ Score

For any general value β :

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

For $\beta=1$:

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

CHAPTER 4

4. RESULT, ANALYSIS AND COMPARISONS

4.1 Result, Analysis and Comparison

In this section, the variants of the CNN model mentioned in chapter 3 are compared along with other training parameters like batch size and no. of epochs. The comparison is done based on performance metrics: classification accuracy rate, error rate, TPR and TNR. The table 4-1 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 metrics has been presented.

Table 4-1: 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

4.1.1 Comparison result of classification accuracy rate

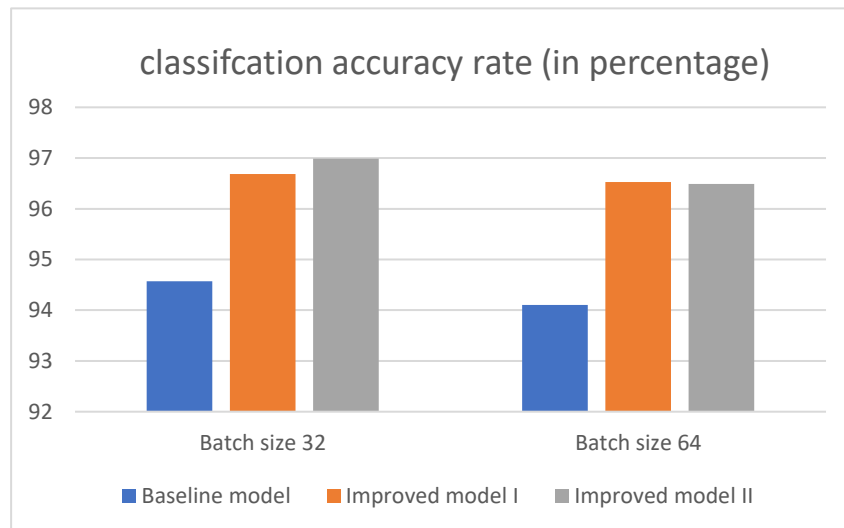


Fig 4-1: Comparison graph of classification accuracy rate

The graph on fig4-1 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.

4.1.2 Comparison result of error rate

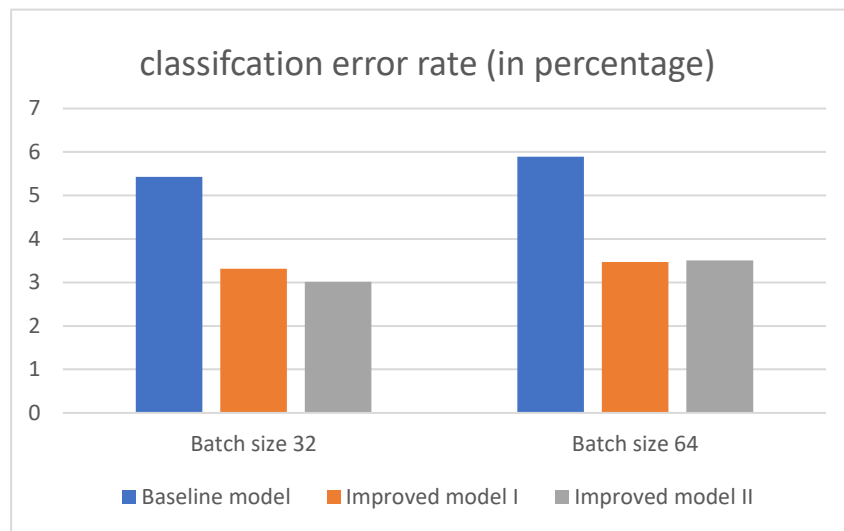


Fig 4-2: Comparison graph of classification error rate

Based on fig4-2, 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.

4.1.3 Comparison result of True Positive Rate

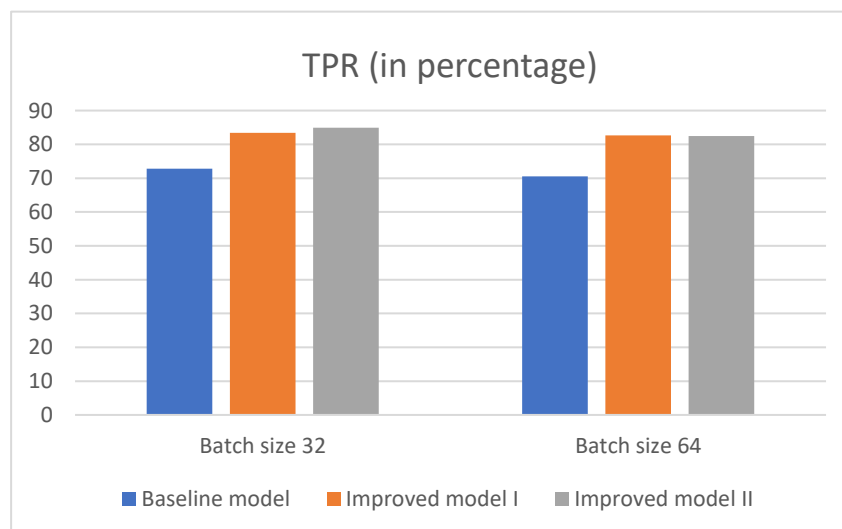


Fig 4-3: Comparison graph of TPR

Based on graph on image Fig 4-3, 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

4.1.4 Comparison result of True Negative Rate

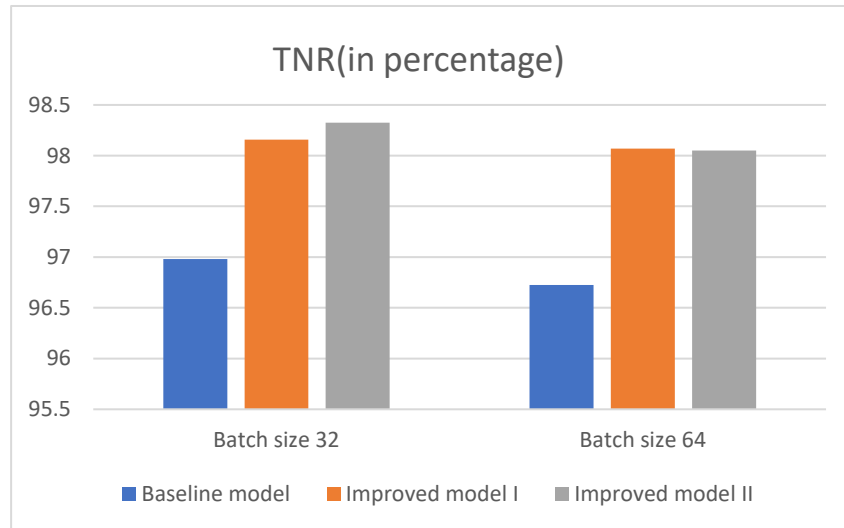


Fig 4-4: Comparison graph of TNR

The graph on the fig 4-4 shows that the TNR increases in each successive variant by 1.176 % and 0.168 % 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.

4.1.5 Comparison result of Positive Predictive Value

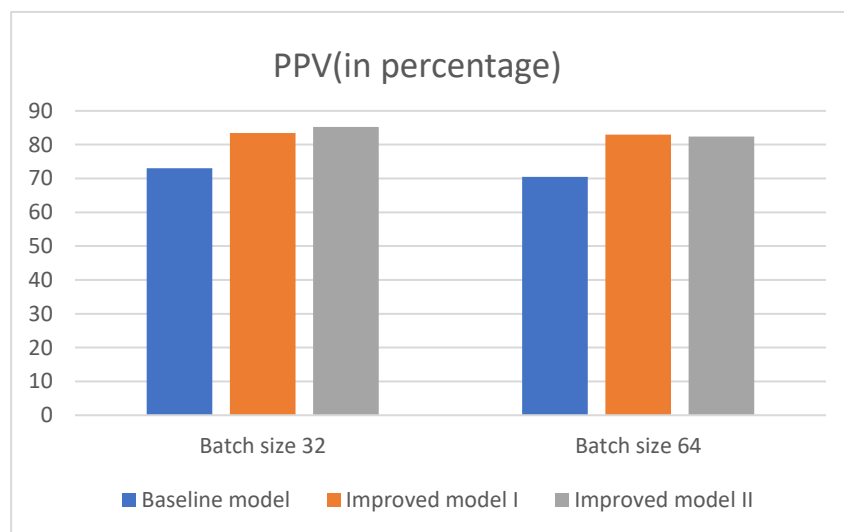


Fig 4-5: Comparison graph of PPV

The graph on the fig 4-5 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.

4.1.6 Comparison result of F_1 score

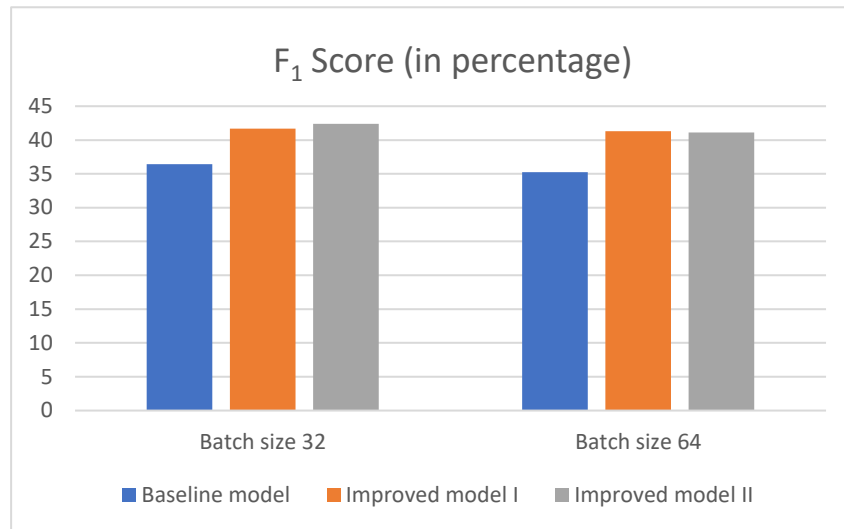


Fig 4-6: Comparison graph of F_1 score

Based on the fig 4, the F_1 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.

CHAPTER 5

5. CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In this project, 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 project shows that dropout technique greatly increased performance of the model while 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 project also demonstrates that 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.

5.2 Future Works

It can be further developed and extended to work on other datasets as well as custom 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 for solving various other image classification tasks in the future.

REFERENCES

- [1] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
- [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.