IMAGE CLASSIFICATION USNG CONVOLUTIONAL NEURAL NETWORK

EKTA BHATIA
STUDENT ID -101227840
CMAIL- ektabhatia@cmail.carleton.ca
Date -12 April 2023

ABSTRACT

In the project, sequential convolutional neural networks with a variety of layers—including input, convolution, max pooling, and density—are defined. The CIFAR data set's picture classification uses the predefined convolutional neural network. There are many uses for image categorization and processing in the healthcare and other obvious areas. The epoch values are monitored to compare various metrics such as training loss, test loss, validation loss, and validation accuracy for different epoch values to train the model effectively and comprehend the comparison while iterating the models. It is easier for the model to learn the patterns when normalisation is done earlier in the training process.

ReLu activation function is chosen because it reduces training time while increasing model accuracy. The Adam optimizer and the Sparse categorical cross-entropy loss function were used to train the model. By repeatedly iterating the importance of epochs using the Keras framework, this study shows the effectiveness of CNN and deep learning for picture classification tasks. Thus, this model is designed to classify the images for CIFAR data set fairly accurate while making comparisons to prevent underfitting and overfitting of the data.

TABLE OF ACRONYMS

CNN	Convolutional Neural Network
VGG	Visual Geometry Group
ReLu	Rectified Linear Activation Function
CIFAR	Canadian Institute for Advanced Research
ResNet	Residual Network
DL	Deep Learning

<u>**TABLE OF CONTENTS**</u>

CHAPTER 1

Introduction

1.1 Motivation for Selecting the Application

I chose the project “Classification of Images from CIFAR -10 Dataset Using Convolutional Neural Networks because image classification is an important task in various domains.

In addition, the CIFAR dataset is a widely used dataset for image classification tasks, consisting of 60,000 32x32 color images in 10 categories. It also has a diverse set of objects in different environments, which makes it suitable for testing the generalizability of deep learning models, and having different sets of objects helps train the model. defines different types of objects. Also, it is freely available and widely used, so many resources and tools can be used with it. For this project, I plan to create a CNN based image classification model using a deep learning framework like TensorFlow. With the ability to capture local information in an image such as edges and corners and integrate them to create high-level features, CNN is uniquely designed to learn the spatial hierarchy of features from raw input data. This allows it to learn complex features from the data. By reducing the spatial size of the feature map, the clustering layer of a CNN can help make the model more efficient and less prone to overfitting. Therefore, CNN was used to classify images from the CIFAR dataset. I used this specific data set and experimented with different tools and functions to compare the test results and achieve test accuracy after reading numerous papers, such as " (Divya, Adepu, & Kamakshi, 2022)classification and enhancement of CIFAR 10 using Convolutional Neural Network," published by IEEE, that provided details on how to achieve accuracy and make detailed comparisons which gave me the insights to manage my project and compare the results.

1.2 Why Deep Learning?

Deep learning has models and algorithm that can learn from large and complex datasets, extract features, and generalize well to new data.

There are several key motivations for using deep learning:

<1> End to end learning: It allows to map raw input to output thereby making the model more accurate and faster.

<2>High Accuracy- The Deep learning models have shown to achieve high performance in various tasks of image classification, natural language processing and speech recognition.

<3>Scalability- Deep learning algorithms can be scaled up to handle larger and more complex data sets.

<4>Automation- Using the deep learning algorithms, it can be used to automatically categorize, classify, and recognize pattern in data.

1.2 OBJECTIVE

A Deep Learning model that can effectively classify images into the respective categories is what the image classification of CIFAR data set using convolutional neural network is going for.
In 10 classes, the CIFAR dataset contains 60000 32x32 images. The objective is to train the CNN model on this dataset such that it can effectively classify the image into appropriate categories.

The project's concise goals can include any of the following:

- Increasing the amount of data and improving the performance of the CNN may include normalisation of data.

- The creation and training of a CNN architecture capable of successfully extracting features from the images in the CIFAR dataset. This might include hyperparameters, including the quantity of layers, the size of the filter, the number of pooling layers.
- Assessing the CNN's performance on the validation and test sets and fine-tuning the model to increase precision and decrease overfitting.
- Gaining insights into how the model makes predictions requires visualising the learned features and evaluating the outcomes.

CHAPTER 2

2.1 Background: related concepts and survey of related solutions

Convolutional Neural Networks (CNNs) are a type of deep learning neural networks that have been demonstrated to produce quality on image classification tasks.

A generalized comparison of related approaches to choose CNN for the CIFAR dataset's image classification. Choosing a CNN for image classification on the CIFAR dataset can be done in several different ways.

- One method is to employ a pre-trained CNN architecture that has been optimised on the CIFAR dataset after being trained on a bigger dataset,
- Other methods include using hyperparameter tuning and architecture search.

In CIFAR datasets, some well-liked CNN architectures are employed for image classification.

- VGG:- It is a type of CNN model which is very efficient and serves as a strong baseline for many applications such as computer vision and is used for object detection extensively. It features are used across many neural networks.
- ResNet: It can be used to solve NLP problem of real world easily. It was used by Microsoft research team and has an exceptionally well computational speed as of a GPU.

- Mobile nets – They are CNN that can be used in mobile devices. Proven to have better results than VGG and has been extremely used along with google API to detect the objects.
- GoogleLE Net -It is an architecture used by google for classification task. To reduce the number of parameters it makes use of heavy pooling layers on top of CNN.

In summary, the choice of CNN architecture for image classification on CIFAR datasets depends on various factors such as accuracy, computational efficiency, and memory usage and training and testing error.

2.2 WHY CNN?

• CNN is specifically made for image identification tasks: CNNs are built to handle structured input data, like photos, unlike other neural network architectures. CNNs are particularly well-suited to image classification problems because they have multiple layers of neurons that learn hierarchical representations of pictures.

• CNN can learn feature automatically making it more adaptable and flexible to varied datasets. As they can learn feature automatically there is no human intervention and need.

• CNNs can tackle large amount of data: To train models efficiently for image classification tasks, a lot of data is frequently needed. Large data sets can be handled by CNN, and it can generalise fresh data.

• High Accuracy -CNN has outperformed other organisations in a variety of image recognition tasks, including those using the CIFAR dataset.

• Sharing Preferences: CNN employs a parameter sharing to reduce the number of parameters in network thereby making it more efficient and easier.

CHAPTER 3: System description:

3.1 Describe your dataset.

A collection of labelled photos called the CIFAR dataset (Canadian Institute for Advanced Study) is utilised for image classification and object recognition applications. The 2004-created dataset has since evolved into a benchmark for testing picture categorization techniques.

The CIFAR dataset comes in two different iterations: CIFAR-10 and CIFAR-100. Ten layers of 6,000 images each make up the 60,000 32x32 colour images that make up the CIFAR-10 dataset. The courses include:

- Cars
- Birds
- Cats
- Deer
- Dogs
- Frogs
- Horses
- Ships
- Trucks

CIFAR dataset is widely used to test and compare the performance of image classification algorithms, and many of the most advanced algorithms have been developed and tested on this dataset.

3.2 ALGORITHM

The algorithm used for classifying the images of CIFAR data set using Convolutional Neural Network is as follows:

1> Import the necessary libraries such as

- Numpy- To perform numerical computation.
- Tensorflow- For building and training neural network.
- Keras- For building and training neural network.
- Matplotlib- To plot the graphs.
- Regularizers – To provide functionality to the weights in the model and to prevent overfitting.

2>Load the CIFAR dataset and assign labels to training and test data and use the load data function to return Numpy arrays.

3>Perform data normalization by scaling the pixel value of training and test data between 0 and 1 which can improve model accuracy and convergence and using the shape tuple to return the array of size of images.

4>Assign the name to 10 classes of CIFAR train Dataset as follows:

'airplane', 'automobile','bird','cat','deer','dog','frog','horse','ship','truck' and visualize the images using matplotlib.

5>Create an empty sequential model object and add the following layers to it as follows:

- Convolutional layer with reLu activation function
- Max pooling layer to reduce the spatial dimensions.

- Additional Convolutional and Max pooling layer to down sample the features from input image
- Droupout layer to reduce the overfitting.

6>Further add the following layers:

Flatten layer-To reduce the output of above layer into 1 -D and feed it into the

Neural netwok

Dense layer-To perform linear transformation on the obtained output

7> Use the optimizer function to compute the accuracy and make the model learn by making use of “adam” optimizer and loss function optimization.

8>Train the CIFAR dataset using the fit function specifying the complete pass for training data to be 20 i.e., 20 epochs and this model makes use of 20 percent data as the validation data set.

9> Plot the training loss, test accuracy, validation loss and validation accuracy as a graphical representation by making use of matplotlib libraries.

10>Next the model can be tested for a sample image and obtain the image classification result for a particular image.

3.3 TRAINING AND TESTING PROCESS

3.3.2 TRAINING PROCESS

1> Load the CIFAR dataset using the import and load data commands.
2> Next the normalization of 80 percent of training dataset is performed to scale the value between 0 and 1 to increase the accuracy.
3> Split the dataset into two sets 80 percent for training and 20 percent for validation.
4> Create an empty sequential model object and add the following layers to it as follows:

   i. Convolutional layer with reLu activation function
   ii. Max pooling layer to reduce the spatial dimensions.
   iii. Additional Convolutional and Max pooling layer to down sample the features from input image
   iv. Dropout layer to reduce overfitting.

5> Feed the dataset to the model defined above and adjust the weights to minimize the loss function.
6> Adam optimizer is used to update the weights adaptively during the training which helps to converge faster and efficiently.

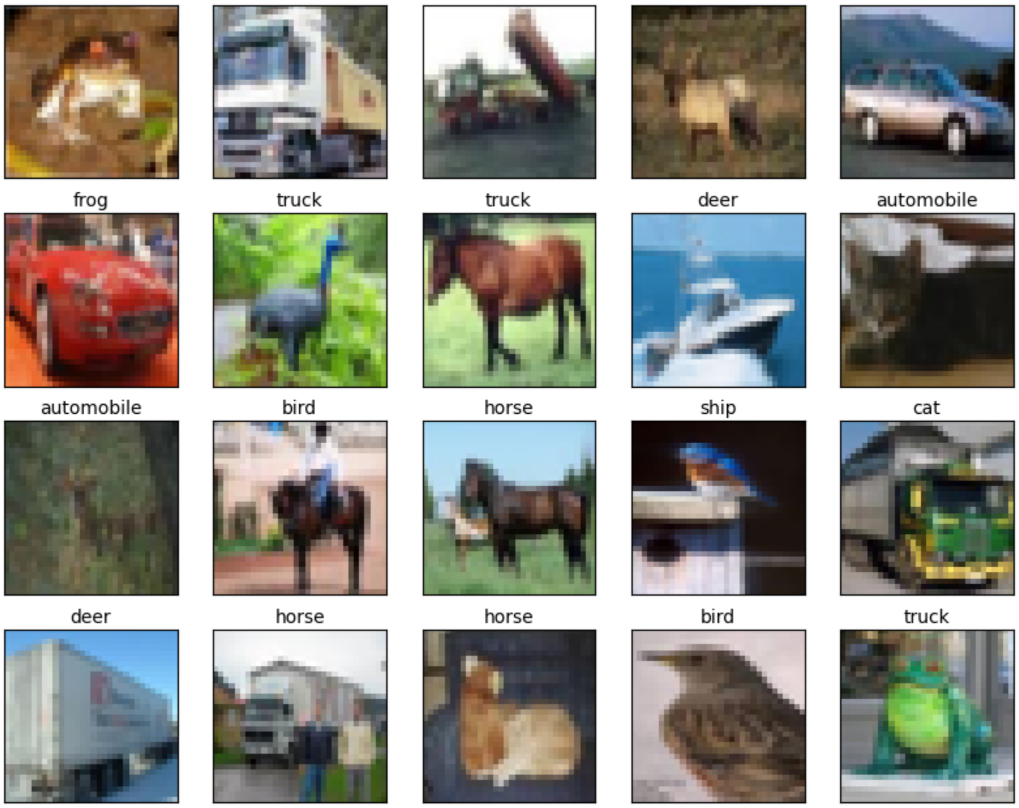
3.3.3 TESTING PROCESS

1> After the training of the data set the data set is tested by using the evaluate function ()
2> The evaluate function returns the test loss and accuracy of the trained model.
3> The test loss and accuracy are then compared by plotting the graph.
4> The accuracy is test by increasing and decreasing the number of epochs.
5> The result of variation of epoch is analysed.

CHAPTER 4

NUMERICAL RESULTS:

4.1 IMAGE WITH THE LABELS



4.2 SEQUENTIAL MODEL

The model is created with the following layers:

1> Convolutional layer with 3X3 relu activation function and with 32 filters.

2>Max pooling layer to reduce the spatial dimension.

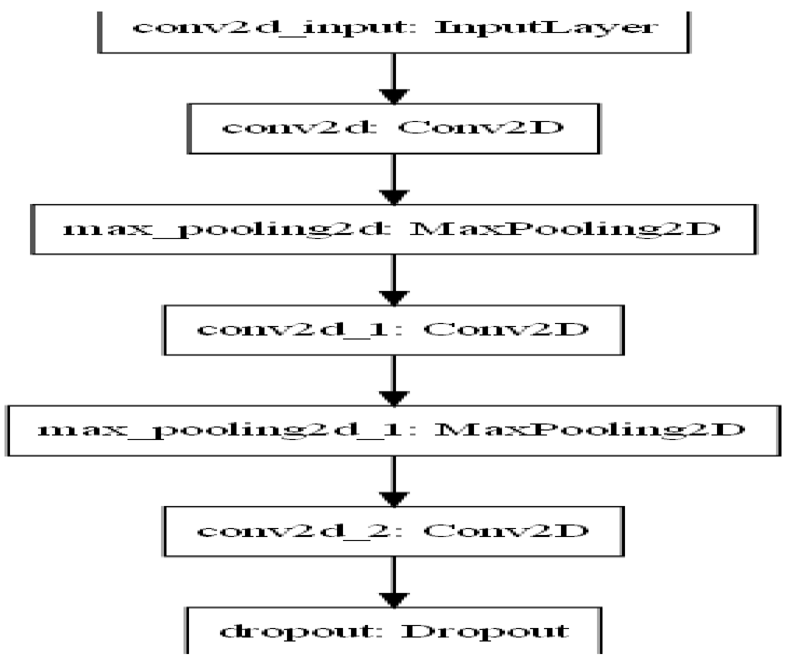
3>Convolutional layer with 64 filters is defined.

4>Droupout layer with rate of 0.4 is defined.

[8]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 64)          36928
_________________________________________________________________
dropout (Dropout)            (None, 4, 4, 64)          0
=================================================================
Total params: 56,320
Trainable params: 56,320
Non-trainable params: 0
_________________________________________________________________
```

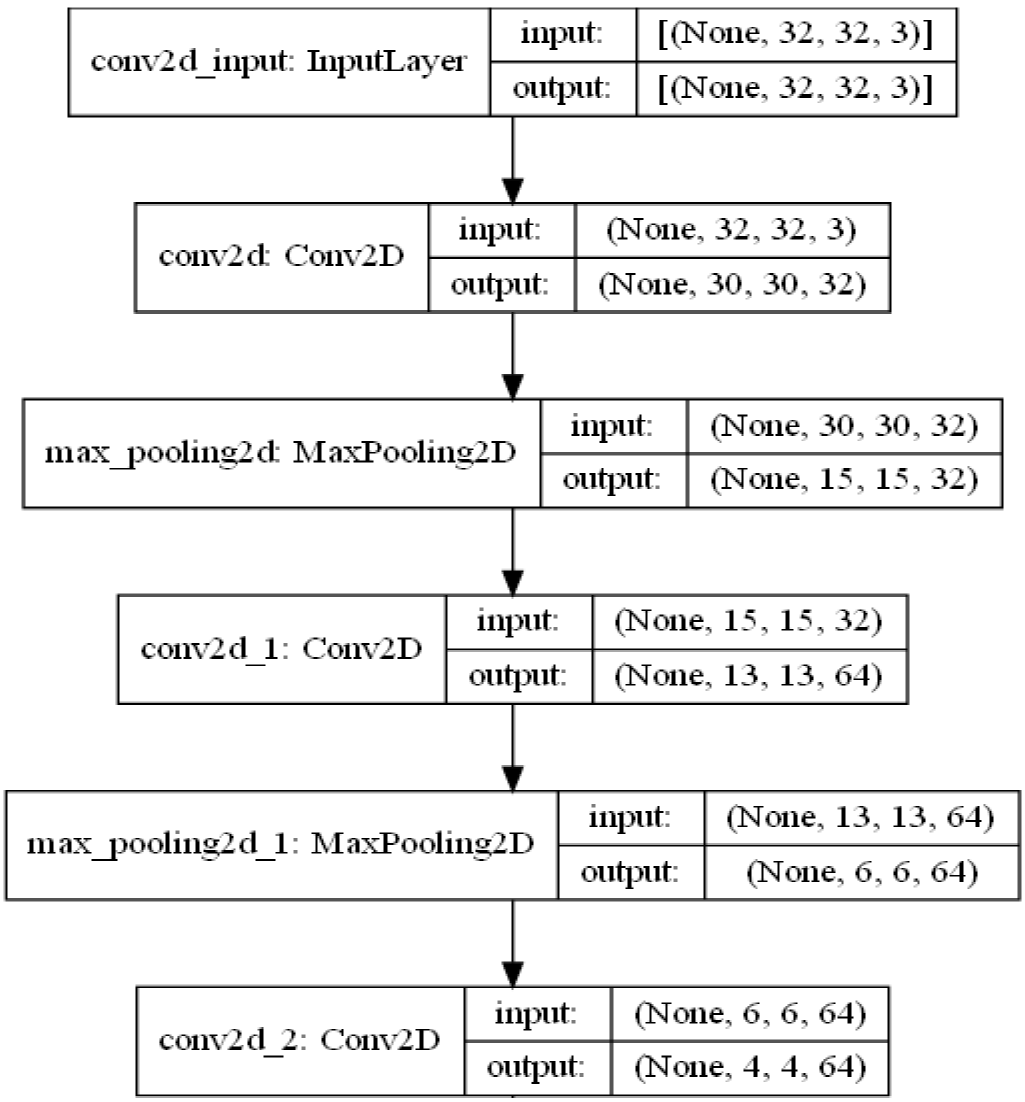


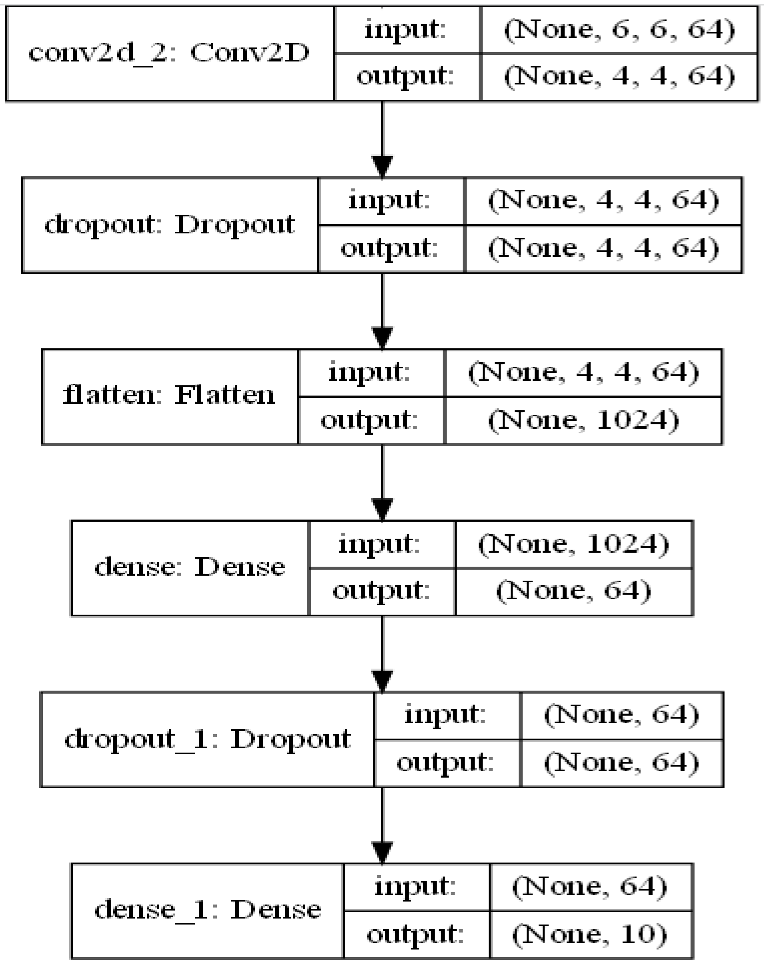
4.3 ADDITIONAL LAYERS ADDED

1> DENSITY LAYER- Used to create relation between input and output data.

2> FLATTEN LAYER- To reduce the output of the layer to 1-D.

[10]:

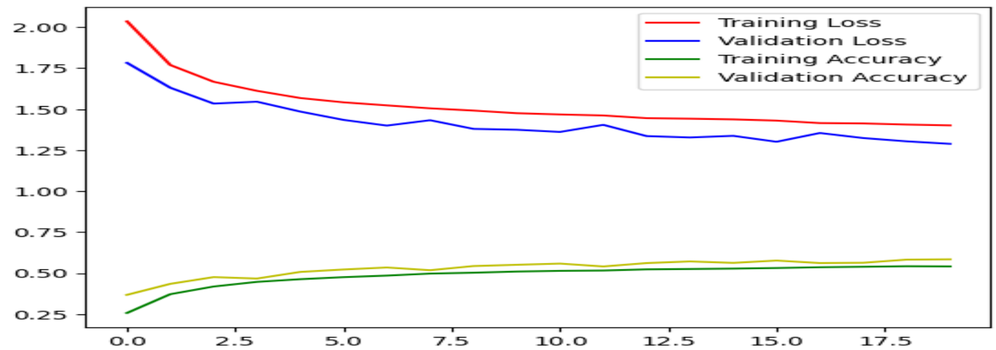


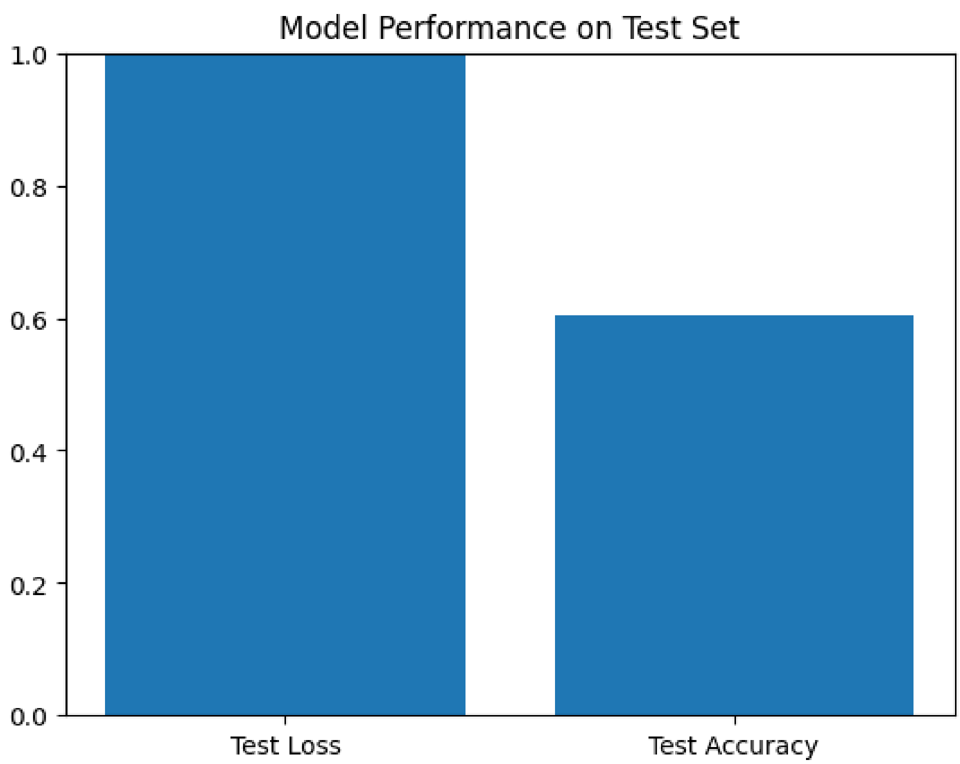

conv2d_2: Conv2D
input:
(None, 6, 6, 64)
output:
(None, 4, 4, 64)
dropout: Dropout
input:
(None, 4, 4, 64)
output:
(None, 4, 4, 64)
flatten: Flatten
input:
(None, 4, 4, 64)
output:
(None, 1024)
dense: Dense
input:
(None, 1024)
output:
(None, 64)
dropout_1: Dropout
input:
(None, 64)
output:
(None, 64)
dense_1: Dense
input:
(None, 64)
output:
(None, 10)

4.4 BEST FIT

Model is trained for 20 epochs and is best fit in comparison to other value for epochs.

EPOCH =20



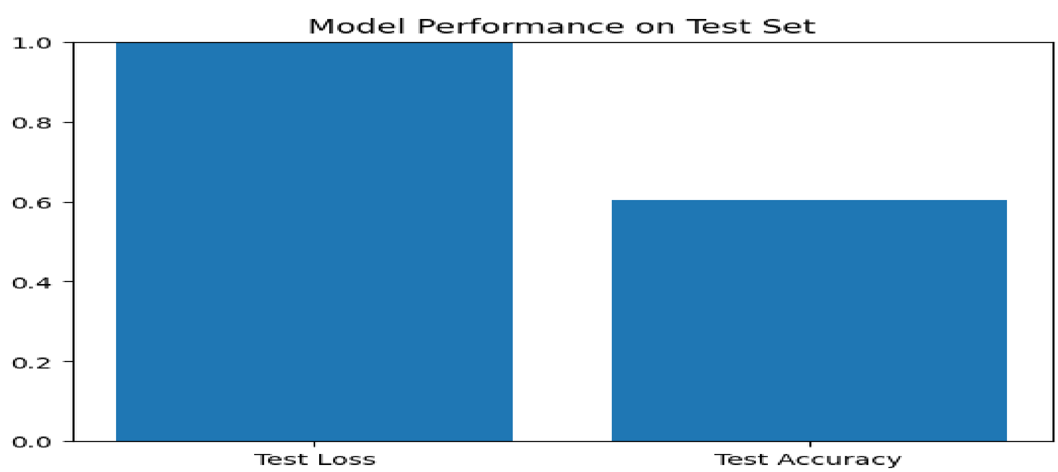


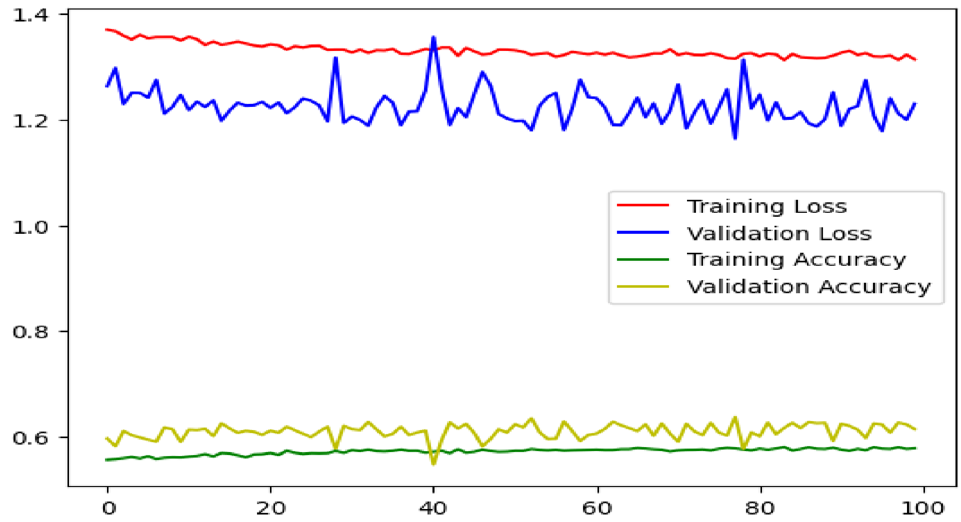
TRAINING LOSS VS VALIDATION LOSS

```
50000/50000 [==============================] - 49s 975us/sample - loss: 1.3242 - acc: 0.5790 - val_loss: 1.2407
```

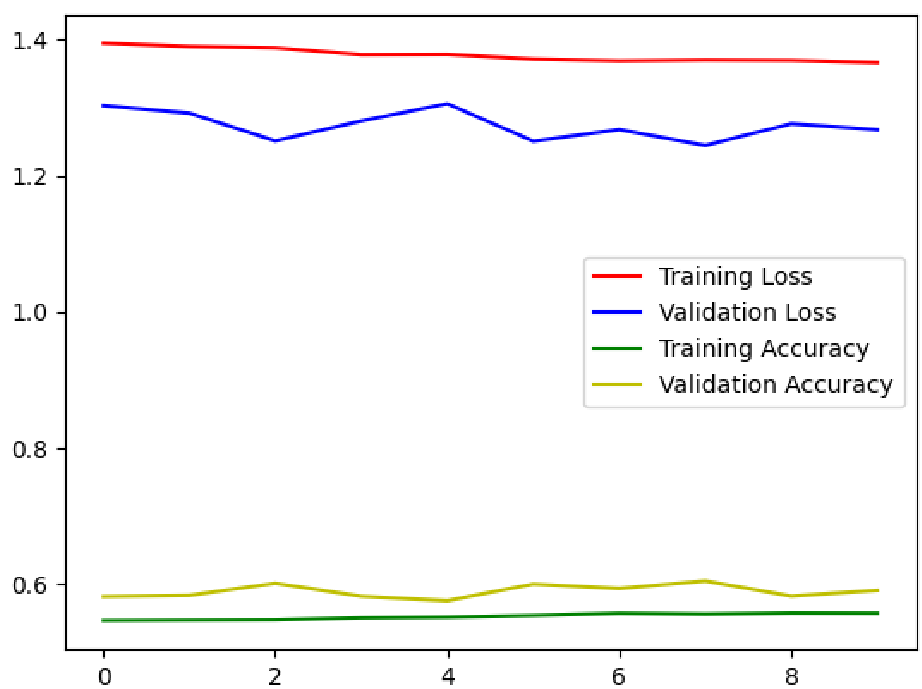
In [ ]:

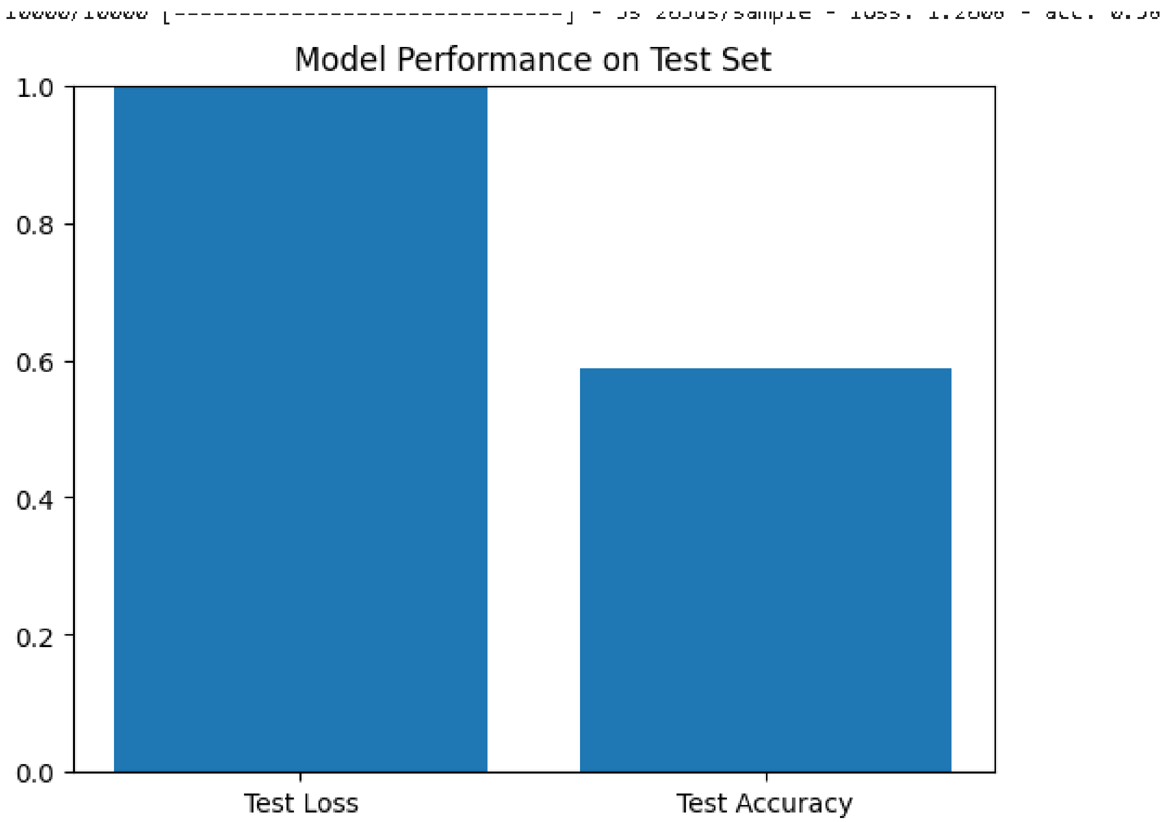
EPOCHS = 100 OVERFITTING



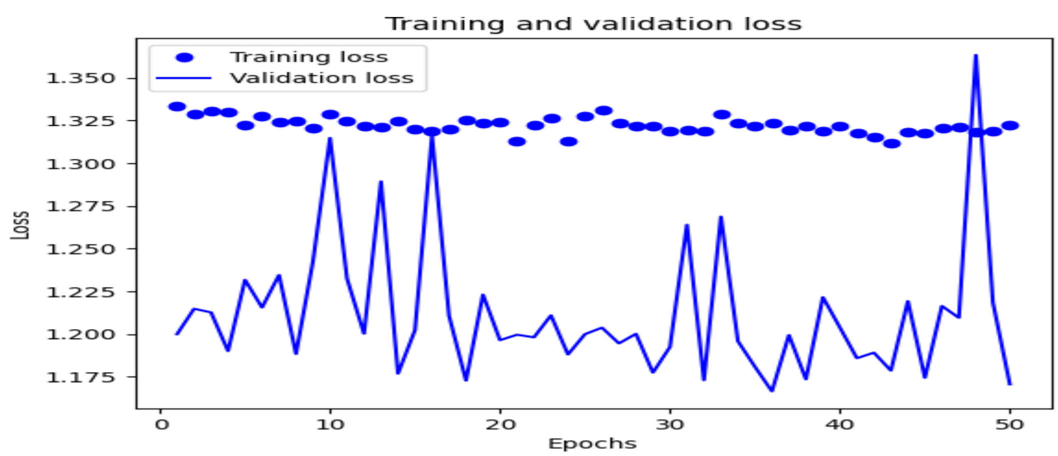


EPOCHS =10 UNDERFITTING

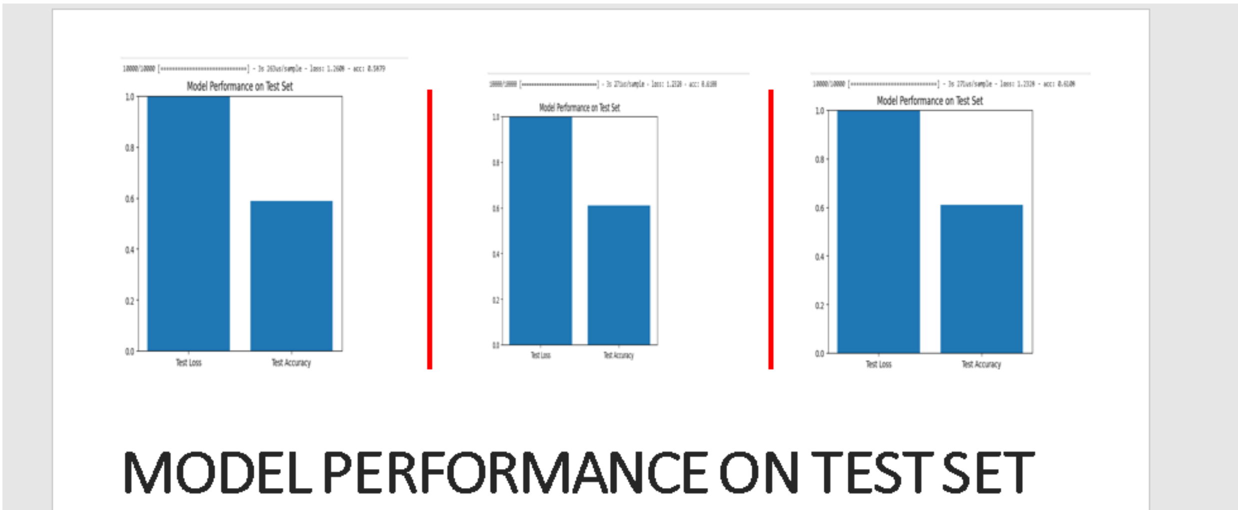




TRAINING AND VALIDATION LOSS FOR 50 EPOCHS



4.5 COMPARATIVE RESULTS



CONCLUSION

To sum up, image classification using CNN is a potent technique for categorising and identifying the images and requires in-depth familiarity with deep learning, neural networks, data sets, and different classifying applications. I want to thank my professor Changcheng Huang since he provided me with knowledge that allowed me to fully understand the ideas and chose my project while considering the numerous algorithms we learned in class.

Further two major conclusions are as follows:

- By varying the number of epochs while training the model behaviour of the model can be visualized to understand approximate number of epochs to use for training the data to prevent underfitting and overfitting of the data.
- Also, as evident increasing the number of epochs doesnot make the model better but makes it more prone to overfitting hence by choosing the right number of epochs tradeoff between overfitting and accuracy can be obtained.

At the end by using the power and potential of CNN it can be used in wide variety of domains to classify the objects or images accurately and can be used in a wide variety of domains.

REFRENCES

1> (Divya, Adepu, & Kamakshi, 2022)classification and enhancement of CIFAR 10 using Convolutional Neural Network," published by IEEE

2> https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd4

4cf9

3> **https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/**

4> **https://www.investopedia.com/terms/n/neuralnetwork.asp**

5> **https://machinelearningmastery.com/what-is-deep-learning/**

EKTA0104 (1).html