**Mini Project Report on**



**IMAGE PROCESSING**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Bhavya Gupta**  **2018764**

***Under the Mentorship of***

**Mr. Ashwini Kumar**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Image Processing”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Mr. Ashwini Kumar, Designation**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Bhavya Gupta 2018764

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | 1 |
| Chapter 2 | Literature Survey | 4 |
| Chapter 3 | Methodology | 6 |
| Chapter 4 | Result and Discussion | 9 |
| Chapter 5 | Conclusion and Future Work | 12 |
|  | References | 14 |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

**Problem Statement:**

**IMAGE PROCESSING**

* 1. **Introduction**
     1. **What Is an Image?**

Before we jump into image processing, we need to first understand what exactly constitutes an image. An image is represented by its dimensions (height and width) based on the number of pixels. For example, if the dimensions of an image are 500 x 400 (width x height), the total number of pixels in the image is 200000.

This pixel is a point on the image that takes on a specific shade it is an integer value ranging between 0 and 255(0 is completely black and 255 is completely white and all other colors range in between these values), opacity or color. It is usually represented in one of the following:

* Grayscale
* RGB
* RGBA



* + 1. **What Is Image Processing?**

Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying certain predetermined signal processing methods.Image processing requires fixed sequences of operations that are performed at each pixel of an image. The image processor performs the first sequence of operations on the image, pixel by pixel. Once this is fully done, it will begin to perform the second operation, and so on. The output value of these operations can be computed at any pixel of the image.

* + 1. **Types of Image Processing**

There are five main types of image processing:

* Visualization - Find objects that are not visible in the image
* Recognition - Distinguish or detect objects in the image
* Sharpening and restoration - Create an enhanced image from the original image
* Pattern recognition - Measure the various patterns around the objects in the image
* Retrieval - Browse and search images from a large database of digital images that are similar to the original image

***This project is a type of Recognition Image Processing.***

* 1. **4 Applications of Image Processing**
* Medical Image Retrieval
* Traffic Sensing Technologies
* Image Reconstruction
* Face Detection
  1. **CIFAR-10**

The CIFAR-10 dataset (Canadian Institute for Advanced Research, 10 classes) is a subset of the Tiny Images dataset and consists of 60000 32x32 color images. The images are labelled with one of 10 mutually exclusive classes: airplane, automobile (but not truck or pickup truck), bird, cat, deer, dog, frog, horse, ship, and truck (but not pickup truck). There are 6000 images per class with 5000 training and 1000 testing images per class.

The criteria for deciding whether an image belongs to a class were as follows:

* The class name should be high on the list of likely answers to the question “What is in this picture?”
* The image should be photo-realistic. Labelers were instructed to reject line drawings.
* The image should contain only one prominent instance of the object to which the class refers. The object may be partially occluded or seen from an unusual viewpoint as long as its identity is still clear to the labeler.
  1. **Pre-requisites of the project**
* Python
* Tkinter
* Keras
* Tensor Flow
* Convolution Neural Netwoks (CNN)

**Chapter 2**

**Literature Survey**

In this chapter some of the major existing work in these areas has been reviewed.

1. Image Filtering and Enhancement:

* Paper: "A Survey of Image Denoising Techniques" by S. S. Bhat and M. N. Shanmukha Swamy.
* Paper: "Image Enhancement Techniques: A Comprehensive Review" by R. D. Hiremath and R. C. Joshi.
* Book: "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods.

1. Image Segmentation:

* Paper: "A Survey of Image Segmentation Techniques" by A. Mohan and D. Chandrasekaran.
* Paper: "Image Segmentation Techniques: A Survey" by S. Agarwal and D. K. Srivastava.
* Book: "Machine Vision" by Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck.

1. Object Detection and Recognition:

* Paper: "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" by Shaoqing Ren et al.
* Paper: "You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon et al.
* Paper: "Deep Residual Learning for Image Recognition" by Kaiming He et al.

1. Image Registration and Stitching:

* Paper: "A Survey of Image Registration Techniques" by M. Maas and A. Kanitsar.
* Paper: "Image Stitching Techniques: A Review" by M. S. Sun and Y. H. Tai.
* Book: "Digital Image Warping" by George Wolberg.

1. Image Compression:

* Paper: "A Survey of Image Compression Techniques" by M. A. Al-Rawashdeh and I. A. Al-Bahadly.
* Paper: "Image Compression Techniques: A Comprehensive Review" by S. Sharma and A. Kaur.
* Book: "JPEG: Still Image Data Compression Standard" by William B. Pennebaker and Joan L. Mitchell.

1. Deep Learning for Image Processing:

* Paper: "ImageNet Classification with Deep Convolutional Neural Networks" by Alex Krizhevsky et al.
* Paper: "Deep Residual Learning for Image Recognition" by Kaiming He et al.
* Book: "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

**Chapter 3**

**Methodology**

A Simple Keras CNN trained on CIFAR-10 dataset with around 90% accuracy (Without Data Augmentation)

The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. It is one of the most widely used datasets for machine learning research and computer vision.

An Image cassification model trained on the CIFAR-10 dataset. It uses the following layers/functions:

1. For building the Model - CNN, Maxpooling and Dense Layers.
2. For Activation Function - ReLU (in CNN layers for handling image pixels) and Softmax (for final classification).
3. For handling Overfitting (Regularizing) - DropOut Layer.
4. For normalizing/standardizing the inputs between the layers (within the network) and hence accelerating the training, providing regularization and reducing the generalization error - Batch Normalization Layer.

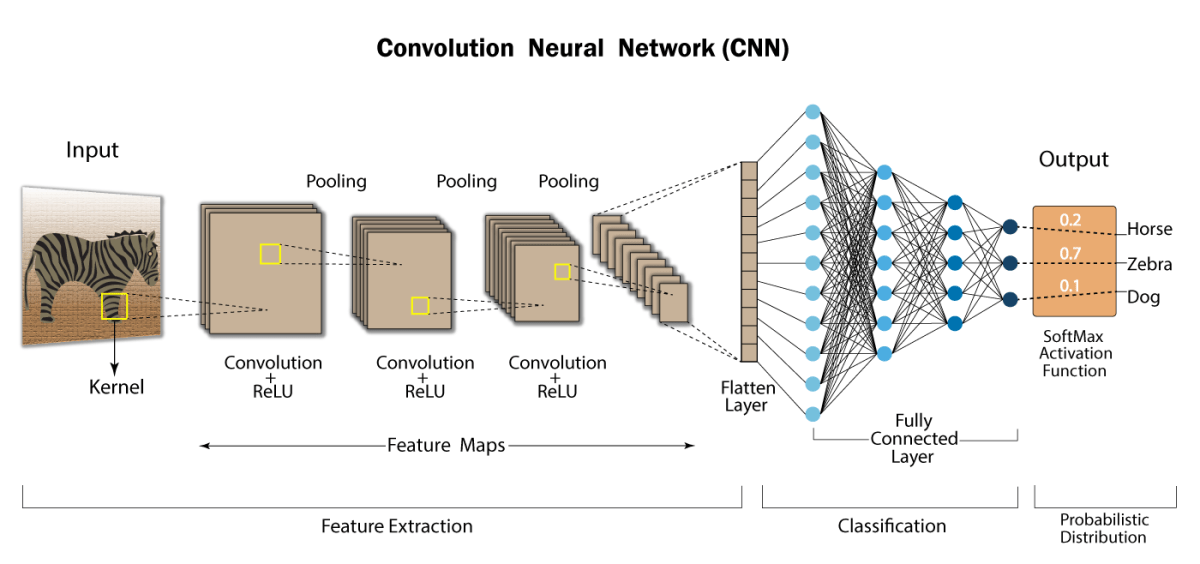
**Why Convolutional Neural Networks?**

Convolution is needed to reduce the amount of computation we need to do when doing image recognition and classification.

A simple neural network may get around 3 billion weigths and biases for out RGB channel which is too heavy to handle, hence we use convolution neural networks to reduce the dimensions hence less calculations and computations.

**How the Model Works**

Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.



The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

Based on the activation map of the final convolution layer, the classification layer outputs a set of confidence scores (values between 0 and 1) that specify how likely the image is to belong to a “class.” For instance, if you have a ConvNet that detects cats, dogs, and horses, the output of the final layer is the possibility that the input image contains any of those animals.

Program-wise approach:

1. Importing Dependencies
   * Python libraries-
     + pandas
     + numpy
     + tkinter
     + Matplotlib
     + Keras
       - Datasets
       - Models
       - Utils
       - Layers
2. Reading the cifar10 dataset from Keras datasets & setting train and test data
3. Some EDA(Exploratory Data Analysis)
4. Data Preprocessing
5. Building the CNN Model using Keras

5.1 Setting up Layers

5.2 Compiling the Model

5.3 Fitting the Model

1. Visualizing the Evaluation
2. Predicting the Results
3. Deployement

The Model is trained on a total of 551,722 Trainable Parameters and provides an accuracy of 91.44 % and test accuracy of 87.44%.

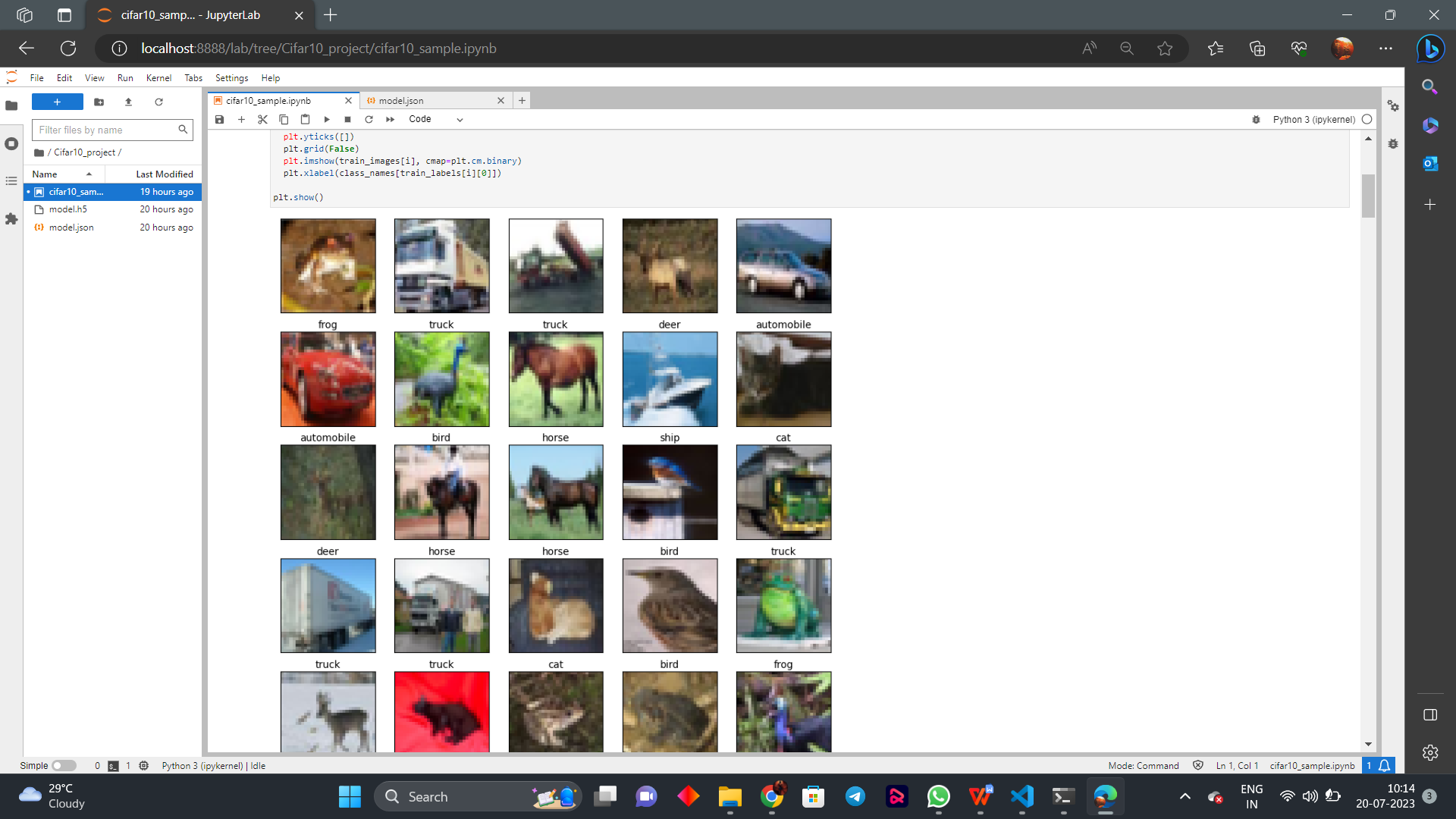
The code for the building the model is done on JupyterNotebook and the GUI is developed on Visual Studio Code using the json file of the trained model.

**Chapter 4**

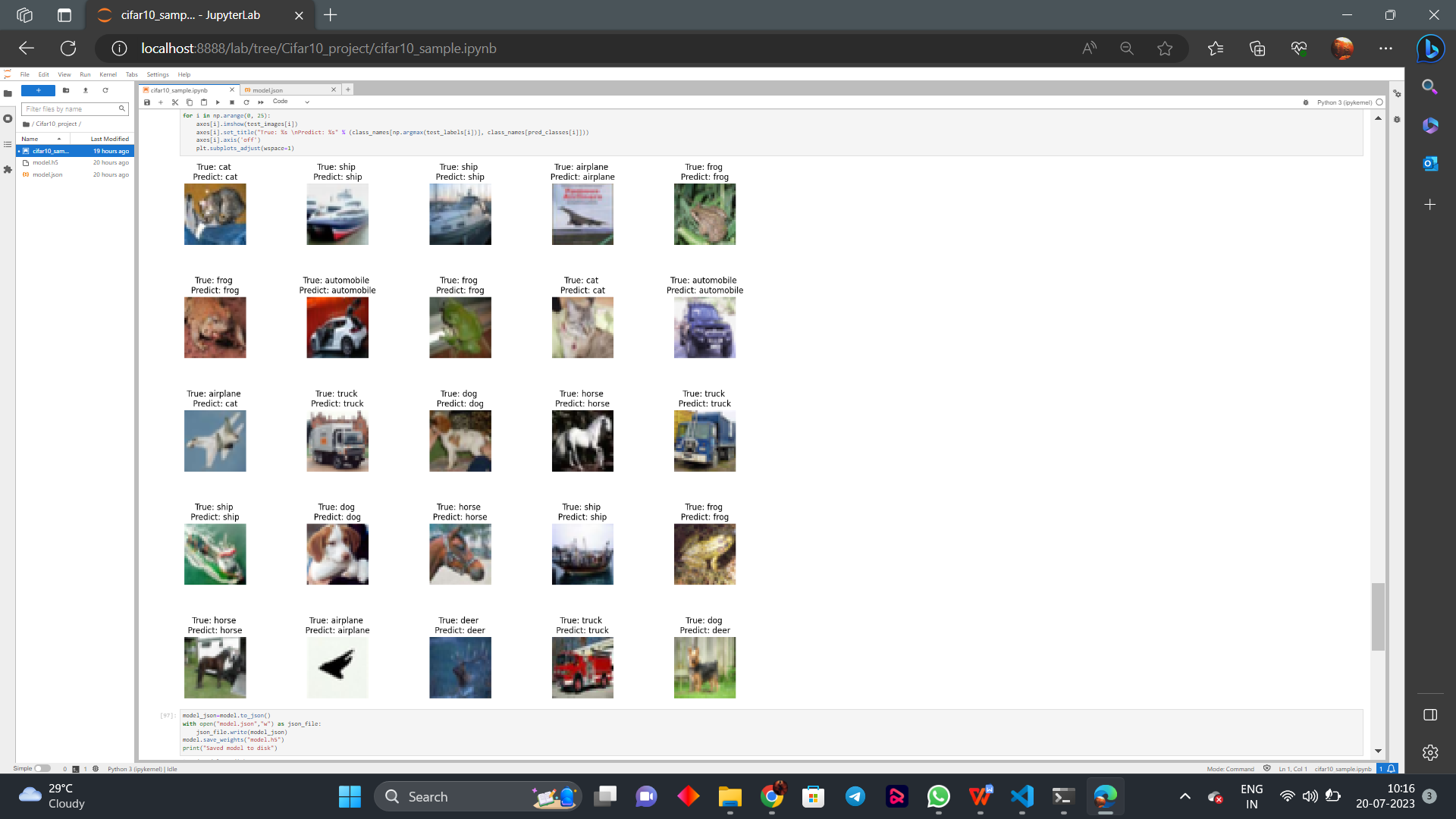
**Result and Discussion**

**4.1 Building the Model**

**Training Data**

****

**Data used for Testing**

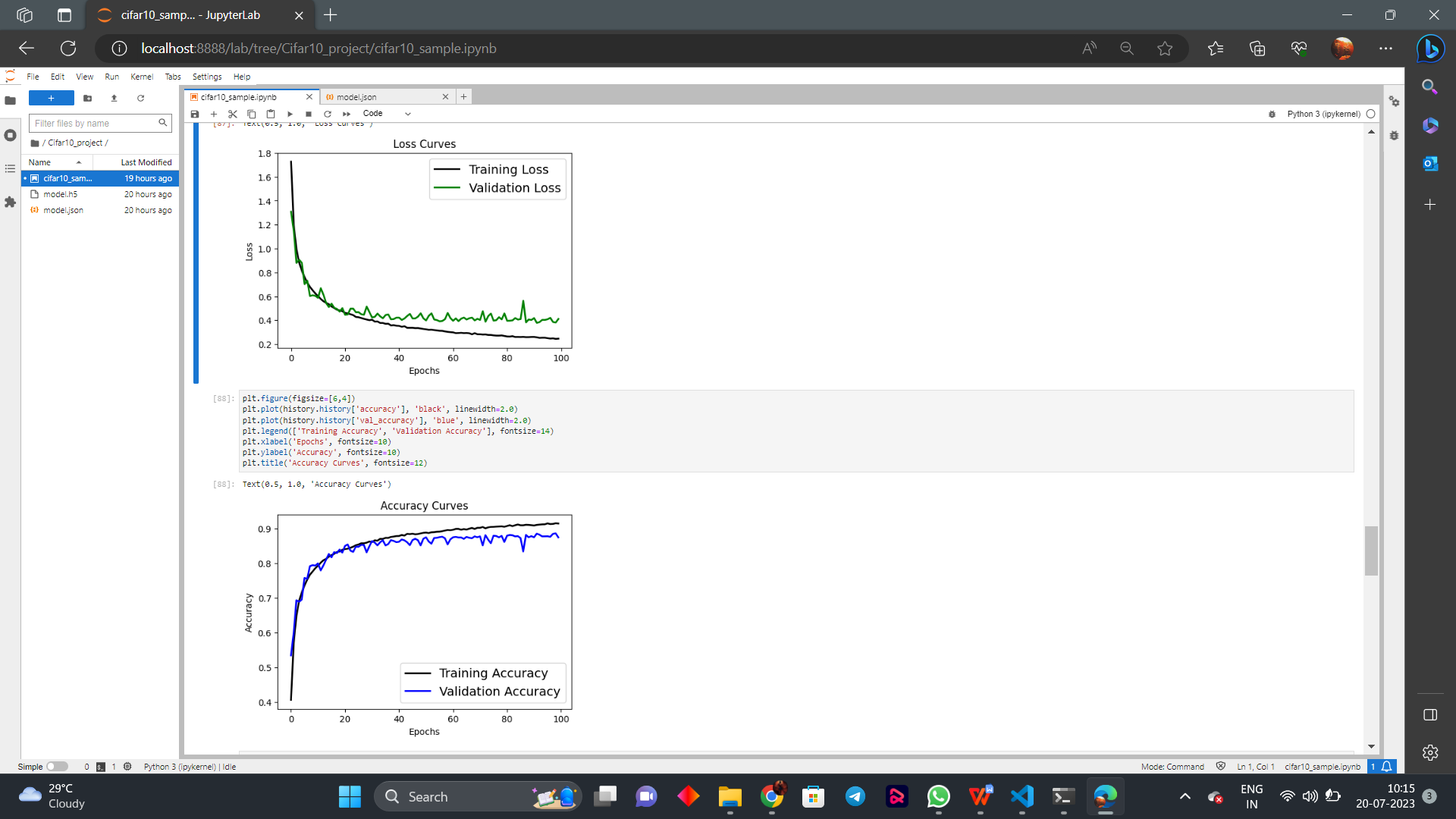
****

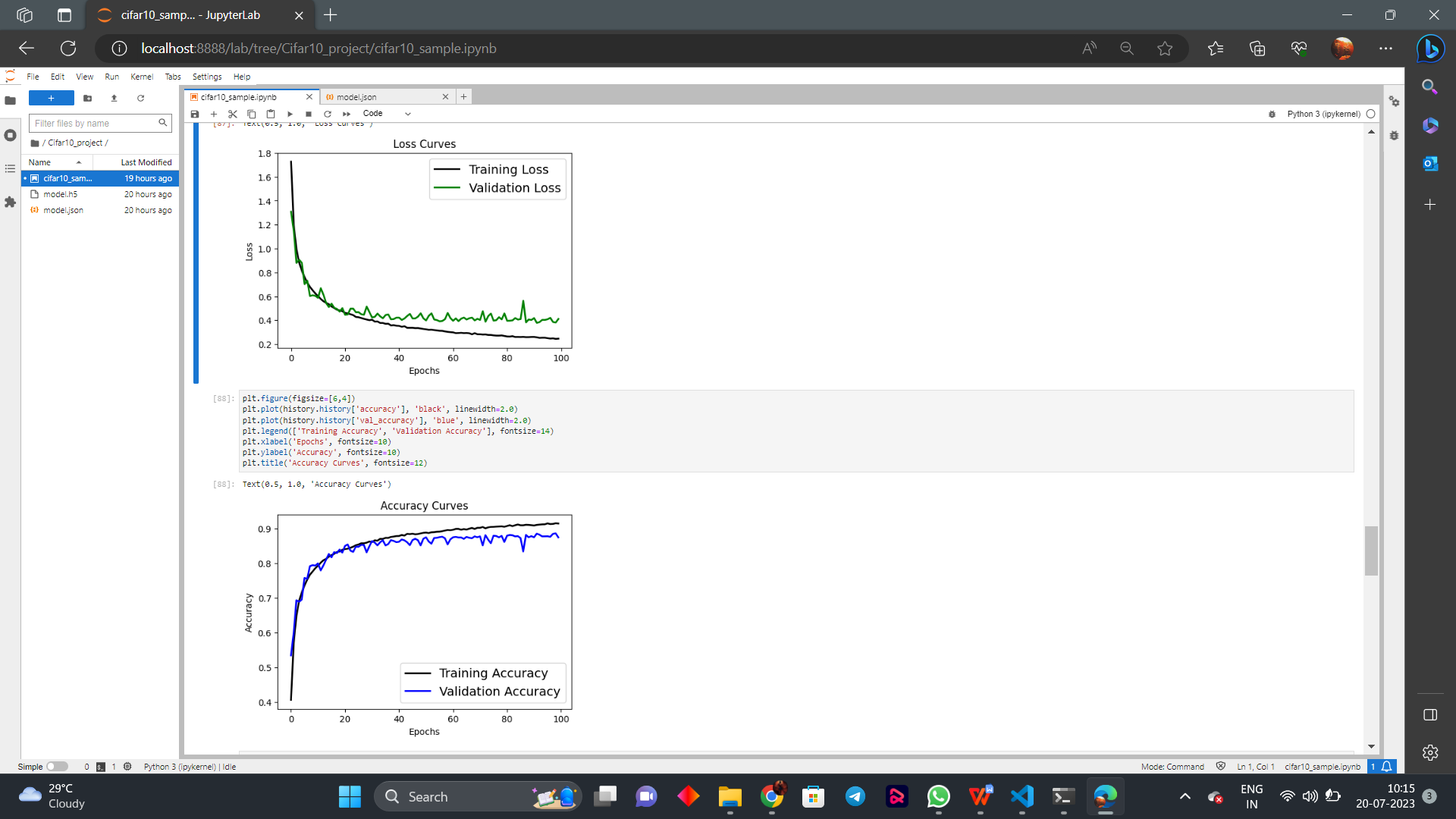
**4.2 Visualizing the Evaluation**

Loss Curve - Comparing the Training Loss with the Testing Loss over increasing Epochs.

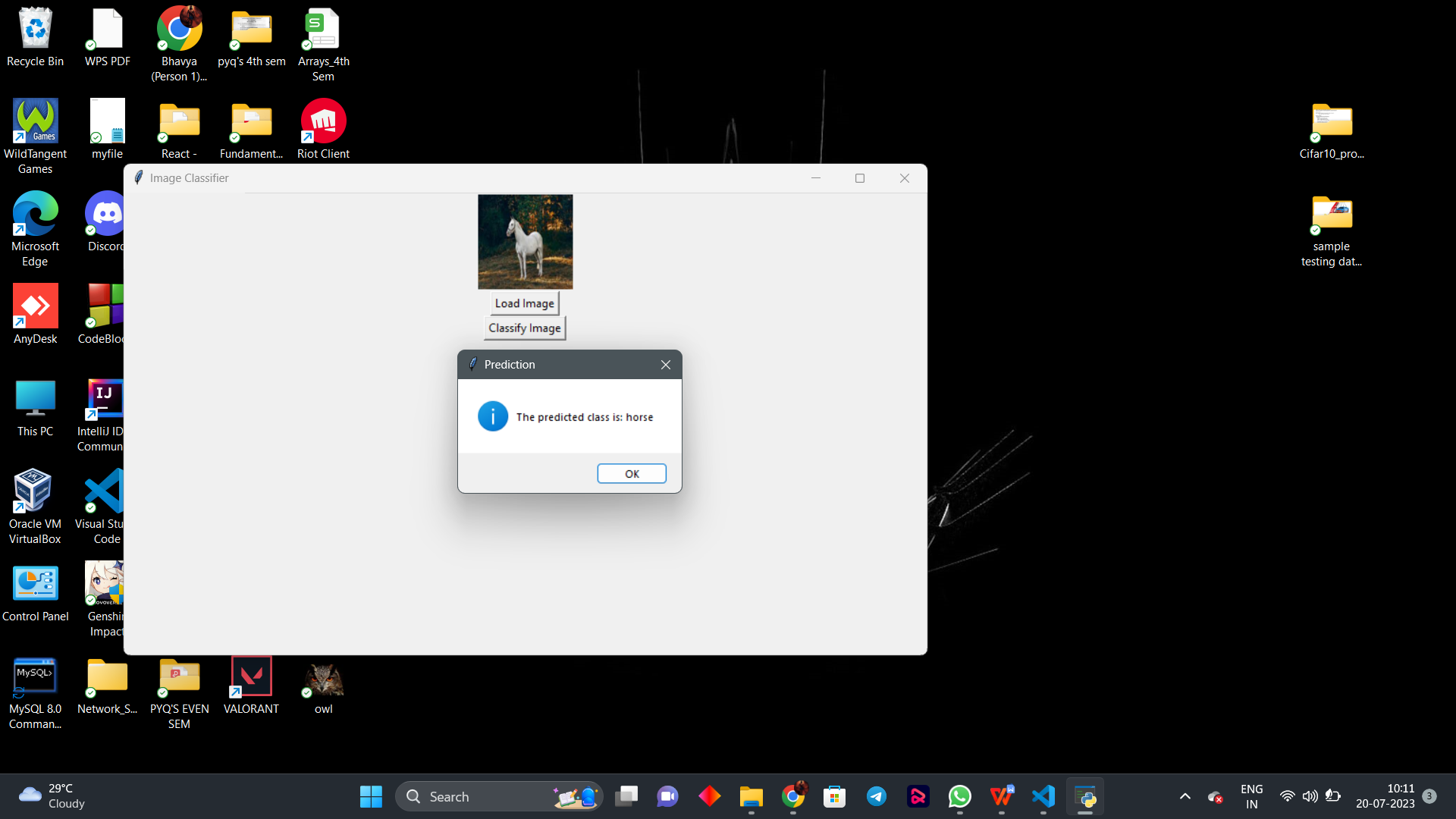
Accuracy Curve - Comparing the Training Accuracy with the Testing Accuracy over

increasing epochs

****

****

**4.3 Representing the Model**

****

1. **Load the Trained Model:**

The trained model is loaded from the saved files model.json (containing the model architecture) and model.h5 (containing the model weights). We use the model\_from\_json function to load the model architecture and load\_weights to load the model weights.

1. **Load Image Button:**

We create a "Load Image" button (load\_button) that, when clicked, opens a file dialog to allow the user to select an image file. The selected image is then displayed in the image\_label.

1. **Classify Image Button:**

The Classify Image Button executes a “classify\_image” function. This function is responsible for preprocessing the loaded image and making a prediction using the trained model. The function does the following steps:

a. Resizes the loaded image to the required input size for the model (32x32).

b. Normalizes the pixel values of the image to be between 0 and 1.

c. Expands the image dimensions to match the expected input shape for the model.

d. Makes a prediction using the model (model.predict).

e. Retrieves the predicted class index with the highest probability using np.argmax.

f. Displays a message box with the predicted class name.

**Chapter 5**

**Conclusion and Future Work**

**5.1 Benefits of Image Processing**

The implementation of image processing techniques has had a massive impact on many tech organizations. Here are some of the most useful benefits of image processing, regardless of the field of operation:

* The digital image can be made available in any desired format (improved image, X-Ray, photo negative, etc)
* It helps to improve images for human interpretation
* Information can be processed and extracted from images for machine interpretation
* The pixels in the image can be manipulated to any desired density and contrast
* Images can be stored and retrieved easily
* It allows for easy electronic transmission of images to third-party providers

**5.2 Limitations**

* Limited dataset size
* Low resolution
* Limited class diversity
* Lack of context
* Overfitting
* Fine-grained tasks
* Transfer learning challenges
* Real-world variations

**5.3 Future Work**

A CIFAR-10 trained model can have various future uses across different applications and domains. Here are some potential future uses of a CIFAR-10 trained model:

1. Image Classification:

The CIFAR-10 trained model can be used to classify new images into one of the ten classes. This can be applied in various image recognition tasks, such as identifying objects in images, detecting specific objects in real-time, or assisting in autonomous vehicles for object recognition.

1. Transfer Learning:

The pre-trained CIFAR-10 model can serve as a starting point for transfer learning. Transfer learning allows you to use knowledge learned from one task (CIFAR-10 classification) and apply it to a related but different task. By fine-tuning the pre-trained model with a smaller dataset specific to a target domain, better performance can be achieved and require less training data.

1. Feature Extraction:

CIFAR-10 trained model to extract relevant features from images for other downstream tasks.

1. Data Augmentation:

The CIFAR-10 trained model can be used in data augmentation pipelines. Data augmentation involves applying various transformations to the input data to increase the size and diversity of the training set, thereby improving the model's generalization ability.

1. Understanding Model Performance:

CIFAR-10 trained model can be used as a benchmark dataset to assess the performance of your models against a well-known baseline.

1. Educational Purposes:

CIFAR-10 is a popular dataset used in many machine learning courses and tutorials. The trained model can be used for educational purposes to demonstrate image classification techniques and the concepts of convolutional neural networks (CNNs).

1. Research and Development:

The CIFAR-10 trained model can be used as a building block for more complex models or be part of a larger system, such as an ensemble of models, in research and development projects.

1. Model Interpretability:

The CIFAR-10 trained model can be used to explore and interpret the learned representations and filters in the convolutional layers. This can provide insights into what the model is learning and how it makes its decisions.

Overall, the future use of a CIFAR-10 trained model is not limited to a single application but can be leveraged in various ways to solve a wide range of image-related problems and research tasks.

**References**

1. <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/#be44>

For introductory knowledge about Convolution Neural networks

1. <https://www.simplilearn.com/image-processing-article> for brief inro to Image processing

[3] <https://scholar.google.com/> <https://ieeexplore.ieee.org/Xplore/home.jsp> <https://arxiv.org/> for research papers and journals

[4]<https://www.coursera.org/learn/cifar-10-image-classification/home> for project guidance.