A

Project Report

On

“IMAGE RECOGNITION USING CIFAR 10”

Submitted in partial fulfillment of the requirements of the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS(BCA)

By

NAME: U .VAMSHI

ROLLNO:1009-22-861-022

Under the guidance of

Dr.M. PURNACHARY

MCA, M.Tech(phd)

Assistant professor

DEPARTMENT OF INFORMATICS,

NIZAM COLLEGE(Autonomous)

(A Constituent College, O. U)

BASHEERBAGH, HYDERABAD

2024-2025

DEPARTMENT OF INFORMATICS

NIZAM COLLEGE (AUTONOMOUS)

(A Constituent College, O.U)

CERTIFICATE

This is to certify that the project entitled "**IMAGE RECOGNITION USING CIFAR 10**" has been submitted by "**U.VAMSHI**" bearing Roll No: "**1009-22-861-022**" in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Computer Applications (BCA)**, Dept. of Informatics, **Nizam College**, **OSMANIA UNIVERSITY**, Hyderabad.

PROJECT SUPERVISOR

Dr.M. PURNACHARY

MCA,M.tech (Phd)

Assistant Professor

EXTERNAL EXAMINER

M.PURNACHARY

HEAD OF THE DEPARTMENT

DEPT. OF INFORMATICS

ACKNOWLEDGMENT

The successful completion of this project would not have been possible without the considerable guidance and assistance from many individuals. I am profoundly grateful for the support I received throughout the course of this work. Any achievements I have made are a direct result of their invaluable contributions, and I would like to extend my sincere thanks to everyone involved.

I am grateful to the **ALMIGHTY** for granting me the strength and health to complete this project without any interruptions. I would also like to express my deepest appreciation to **my parents**, whose unwavering support and encouragement were invaluable throughout the course of this work

I am profoundly grateful to my project guide, **Dr.M. PURNACHARY** Assistant Professor, for her unwavering support and expert guidance throughout the course of this project. I would also like to extend my appreciation to the other faculty members who contributed their time and knowledge to ensure the successful completion of this project. Their collective input and commitment were invaluable in providing the necessary resources and direction. I am truly thankful for their encouragement and expertise.

I wish to extend my heartfelt gratitude to **Mr. M. PURNACHARY**, Head of the Department of Informatics at Nizam College, for his invaluable moral support and guidance. His leadership and counsel played a significant role in the successful completion of our project. I deeply appreciate his commitment to fostering a supportive environment and his readiness to offer his insights whenever needed.

I would like to express my heartfelt gratitude to **Prof. AV.RAJASHEKHAR ,** Principal of Nizam College, and to the management team of Nizam College for providing a conducive environment that allowed us to effectively carry out our academic schedules and complete our project. Their commitment to fostering an atmosphere of learning and support made it possible for us to focus on our work and achieve our goals.

I am deeply grateful for the constant encouragement, support, and guidance from the entire teaching staff of the Department of Informatics, which played a crucial role in the successful completion of our project work. Their expertise and dedication provided us with the resources and assistance we needed throughout this journey.

Additionally, I would like to extend my sincere appreciation to all the non-teaching staff of the Department of Informatics for their timely support. Their contributions, often behind the scenes, were vital to the smooth progress of our project.

Finally, I would like to thank all my friends for their encouragement and support. Their camaraderie and positive feedback were invaluable to me throughout this project.

NAME: U.VAMSHI

ROLLNO:1009-22-861-022

ABSTRACT

Image classification is a vital part of digital image analysis, primarily applied in facial/image recognition, autonomous driving, and image stitching. This project examined the accuracy of different classification models by using the CIFAR-10 dataset, which consists of 60,000 images classified exclusively into ten classes. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. Image recognition refers to the task of inputting an image into a neural network and having it output some kind of label for that image. The label that the network outputs will correspond to a pre-defined class. There can be multiple classes that the image can be labeled as, or just one. If there is a single class, the term "recognition" is often applied, whereas a multi-class recognition task is often called "classification". A subset of image classification is object detection, where specific instances of objects are identified as belonging to a certain class like animals, cars, or people.

CONTENTS

CHAPTER-1
INTRODUCTION

1. INTRODUCTION

1.1 Introduction to Project

Image Processing and Machine Learning, the two trending technologies world. Did you know that we are the most documented generation in history of humanity? Every minute a whopping 1.78 million GB data gets produced online. That's a lot of data and a big chunk that of data is images and videos. This is where automated image processing and machine learning comes in. Innovative minds are exploring those with skills in Machine learning in general and image processing in particular. You would be able to pass an input image to our program and our program should be able to count the number of peoples appearing in that image. Additionally, we would also be creating a bounding box around each of the detected person. An image recognition algorithm (a.k.a an image classifier) takes an image (or a patch of an image) as input and outputs what the image contains. In other words, the output is a class label (e.g. "cat", "dog", "table" etc.). How does an image recognition algorithm know the contents of an image? Well, you have to train the algorithm to learn the differences between different classes. If you want to find cats in images, you need to train an image recognition algorithm with thousands of images of cats and thousands of images of backgrounds that do not contain cats. Needless to say, this algorithm can only understand objects / classes it has learned.

SCOPE

Image recognition technology, driven by machine learning, has broad applications across industries. It aids healthcare in disease diagnosis, enhances automotive safety, and optimizes retail operations. Security systems benefit from its use in surveillance and access control. In agriculture, it aids crop monitoring and yield prediction. Entertainment applications leverage it for content recommendation and augmented reality experiences. Manufacturers utilize it for quality control, and environmental monitoring benefits from its use in wildlife conservation and pollution control.

1.2 EXISTING SYSTEM:

The image recognition applications in industry include fingerprint or retina recognition, processing records of security or traffic cameras. The applications in medicine include ultrasound imaging, magnetic resonance. Stereography is the art of using two almost identical photographs to create a three-dimensional (3D) image. The viewer requires special glasses or a stereoscope to see the 3D image. With modern technology, it has applications in motion picture and television industry. Stereography is a complicated process. Modern stereography uses specialized computer software and camera hardware. Volumetric displays do not require special goggles. The three-dimensional graphics created by this type of display can be viewed from any angle. Each viewer can observe the picture from a different perspective. To create volumetric graphics, a technique called as swept surface volumetric display, which is based on persistence of vision is adopted. Here use of fast-moving lit surfaces creates the illusion of a solid shape. To display volumetric 3D images there is another option which is called as static volume. No moving parts are used in the visible area of the display. However, mirrors and lenses are used to direct a beam of laser light. Very fast pulses of laser light are directed at different points in the air. Persistence of vision gives the illusion of a single solid object. This method is useful for medical diagnosis. A 3D display can show a realistic image of a heart Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies. Early face-detection algorithms focused on the detection of frontal human faces, whereas newer algorithms attempt to solve the more general and difficult problem of multi-view face detection. It is also used in video surveillance. Some recent digital cameras use face detection for autofocus.

1.3 PROPOSED SYSTEM:

As of recent, self-driving vehicles have been gaining traction despite their development since the 1920s. These automated vehicles are able to navigate using a form of visual image recognition developed with deep neural networks. Image recognition is important so that these systems can understand what to avoid, such as trees, animals, and other vehicles, and where the road is. With the continuous development of these systems, we expect image classification to be better, faster, and more accurate. To further understand and explore the concept of image recognition, our team has taken a basic approach to classify a set of images. Much simpler to visual image recognition, our team has selected the CIFAR-10 dataset that has pictures consisting of only one class, that is, each image will only belong to one category. There are a total of 10 categories. Further description of the dataset will be in the "Data Collection & Description" section below. We have found other approaches to image classification of this particular dataset, but we wanted to explore what we learned from the course. Specifically, we used logistic regression, random forest, and convolutional neural network models. Having only been briefly introduced to convolutional neural networks, we found that this was a great chance to delve into this model further through application. For analysis of the models, we observed various outputs, such as the area under receiver operating characteristic curve (AUROC) and the confusion matrix. From selection of the dataset, our team understood the implications that came with it, primarily, the large size of the dataset. This was something our team had to consider when training the models. This will be in the "Data Collection & Description" section below.

ADVANTAGES OF PROPOSED SYSTEM (CIFAR-10):

Deep Neural Network Architecture: So, the idea here is to build a deep neural architecture as opposed to shallow architecture which was not able to learn features of objects accurately. The advantage of multiple layers is that they can learn features at various levels of abstraction. Multiple layers are much better at generalizing because they learn all the intermediate features between the raw input and the high-level classification. At the same time, there are few important aspects which need to be taken care off to prevent over-fitting. Deep CNN are harder to train because:

a) Data requirement increases as the network becomes deeper.

b) Regularization becomes important as number of parameters (weights) increases in order to do learning of weights from memorization of features towards generalization of features. • Data Augmentation: In Keras, we have a ImageDataGenerator class that is used to generate batches of tensor image data with real-time data augmentation. The data will be looped over (in batches) indefinitely. The image data is generated by transforming the actual training images by rotation, crop, shifts, shear, zoom, flip, reflection, normalization etc. • Regularization: Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Given below are few techniques which were proposed recently and has become a general norm these days in convolutional neural networks.

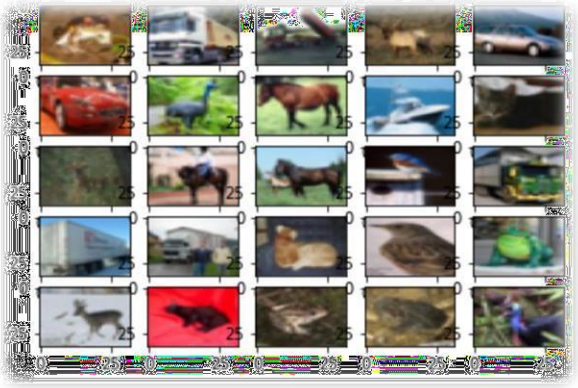


Fig 1: Representation of Images in the Form of Rows

CHAPTER-2
SYSTEM ANALYSIS

2.1 MODULES

A module is a Python object with arbitrarily named attributes that you can bind and reference. Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code

TensorFlow/Keras:

TensorFlow is an open source library created for Python by the Google Brain team. TensorFlow compiles many different algorithms and models together, enabling the user to implement deep neural networks for use in tasks like image recognition/classification and Natural language processing. TensorFlow is a powerful framework that functions by implementing a series of processing nodes, each node representing a mathematical operation, with the entire series of nodes being called a "graph". In terms of Keras, it is a high-level API (application programming interface) that can use TensorFlow's functions underneath (as well as other ML libraries like Theano). Keras was designed with user-friendliness and modularity as its guiding principles. In practical terms, Keras makes implementing the many powerful but often complex functions of TensorFlow as simple as possible, and it's configured to work with Python without any major modifications or configuration. The Packages preferred under TensorFlow module are:

- **Sequential**

The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs

- **Dropout**

The term "dropout" is used for a technique which drops out some nodes of the network. Dropping out can be seen as temporarily deactivating or ignoring neurons of the network. This technique is applied in the training phase to reduce overfitting effects.

- **Dense**

A Dense layer feeds all outputs from the previous layer to all its neurons, each neuron providing one output to the next layer.

- **Flatten**

This Python package provide a function flatten () for flattening dict-like objects. It also provides some key joining methods (reducer), and you can choose the reducer you want or even implement your own reducer.

- **Conv2D**

Keras Conv2D is a 2D Convolution Layer, this layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.

- **MaxPooling2D**

Max pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.)

- **SGD**

Accelerating Deep Learning Using Distributed SGD — An Overview. Stochastic Gradient Descent (SGD) and its multiple variants (such as RMSProp or Adam) are the most popular training algorithms for Deep Learning.

Data Collection and Description

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. We also considered the CIFAR-100 dataset that has an identical number of 60000 total images with 100 classes, but this reduced the number of images per class down to 600 from 6,000. We felt that this fractured the dataset too much to create an effective model.

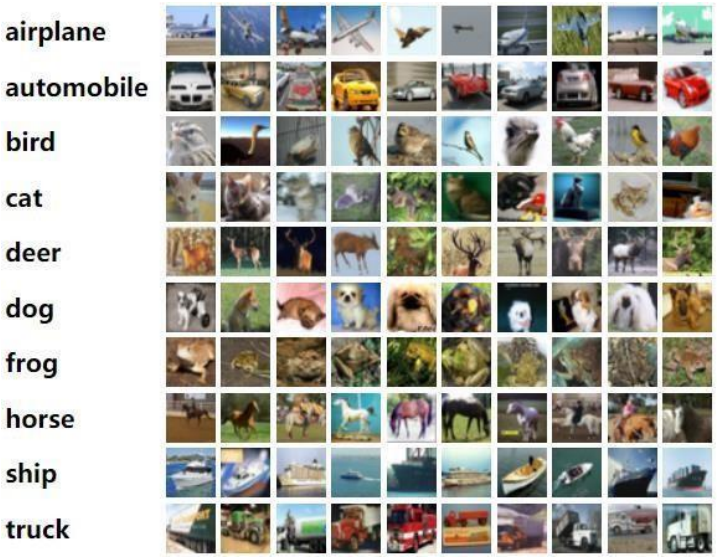


Fig 2 : CIFAR-10 dataset

The chosen CIFAR-10 dataset is divided into five training batches and one test batch, each with 10,000 images. The test batch contains exactly 1,000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5,000 images from each

CHAPTER-3

FEASIBILITY REPORT

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are:

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

3.1 Economical Feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2 Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client.

3.3 SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

CHAPTER-4
SYSTEM REQUIREMENT SPECIFICATION

4.1 SOFTWARE REQUIREMENTS:

1. Python 3.7 version or above.

2. Windows 7,8 or 10 versions.

3. Pycharm Application.

4. I3 and above generation computers.

5. Minimum of 2GB RAM to avoid interruptions.

6. GPU is recommended.

Install pip on your system in python, Download get-pip.py installer script and then open the folder in which it has downloaded and run the following command in the command prompt to install pip successfully: python get-pip.py We need to install all these python libraries in the system terminal. Also install TensorFlow which is very important package for making machine learning models. Install it using the command pip install tensorflow in cmd. You also need keras package for python which is installed using the command pip install keras.

4.2 HARDWARE REQUIREMENTS:

1. Processor above Pentium

2. Operating system: windows 7 or higher version

3. RAM: more than 2 GB 4. GPU: min 512 MB

Chapter-5
SYSTEM DESIGN

5.1 DESIGN:

The system is a specialization of machine language where TensorFlow is the package being considered under Python programming language on the concept of Convolutional Neural Networks, the type of neural network most commonly used in image classification/recognition.

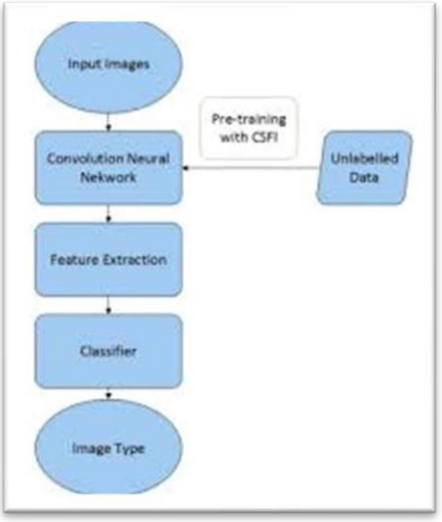


Fig 3 : Flowchart diagram the proposed module

CONVOLUTIONAL NEURAL NETWORK

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image) and height (as image generally have red, green, and blue channels). Before diving into the Convolution Neural Network, let us first revisit some concepts of Neural Network. In a regular Neural Network there are three types of layers

Input Layers: It's the layer in which we give input to our model. The number of neurons in this layer is equal to total number of features in our data (number of pixels in case of an image).

Hidden Layer: The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layers can have different numbers of neurons which are generally greater number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by addition of learnable biases followed by activation function

Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into probability score of each class. The data is then fed into the model and output from each layer is obtained this step is called feedforward, we then calculate the error using an error function, some common error functions are cross entropy, square loss error etc. After that, we back propagate into the model by calculating the derivatives. This step is called Backpropagation which basically is used to minimize the loss.

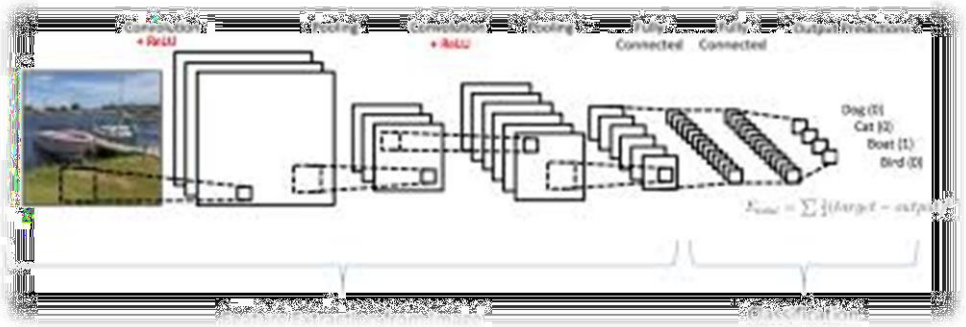


Fig 4 : CNN concept of Image recognition

Feature Extraction

In order to carry out image recognition/classification, the neural network must carry out feature extraction. Features are the elements of the data that you care about which will be fed through the network. In the specific case of image recognition, the features are the groups of pixels, like edges and points, of an object that the network will analyze for patterns. Feature recognition (or feature extraction) is the process of pulling the relevant features out from an input image so that these features can be analyzed. Many images contain annotations or metadata about the image that helps the network find the relevant features.

Classification

Image Recognition refers to the task of inputting an image into a neural network and having it output some kind of label for that image. The label that the network outputs will correspond to a pre-defined class. There can be multiple classes that the image can be labeled as, or just one. If there is a single class, the term “recognition” is often applied, whereas a multi-class recognition task is often called “classification”. A subset of image classification is object detection, where specific instances of objects are identified as belonging to a certain class like animals, cars, or people.

HOW NEURAL NETWORKS LEARN TO RECOGNIZE IMAGES

Getting an intuition of how a neural network recognizes images will help you when you are implementing a neural network model, so let's briefly explore the image recognition process in the next few sections.

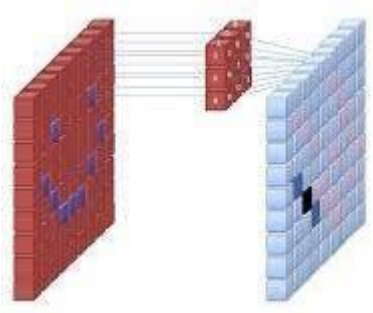


Fig 5 : Feature Extraction with Filters

size The first layer of a neural network takes in all the pixels within an image. After all the data has been fed into the network, different filters are applied to the image, which forms representations of different parts of the image. This is feature extraction and it creates "feature maps". This process of extracting features from an image is accomplished with a "convolutional layer", and convolution is simply forming a representation of part of an image. It is from this convolution concept that we get the term Convolutional Neural Network (CNN), the type of neural network most commonly used in image classification/recognition. If you want to visualize how creating feature maps works, think about shining a flashlight over a picture in a dark room. As you slide the beam over the picture you are learning

about features of the image. A filter is what the network uses to form a representation of the image, and in this metaphor, the light from the flashlight is the filter. The width of your flashlight's beam controls how much of the image you examine at one time, and neural networks have a similar parameter, the filter size. Filter size affects how much of the image, how many pixels, are being examined at one time. A common filter used in CNNs is 3, and this covers both height and width, so the filter examines a 3 x 3 area of pixels

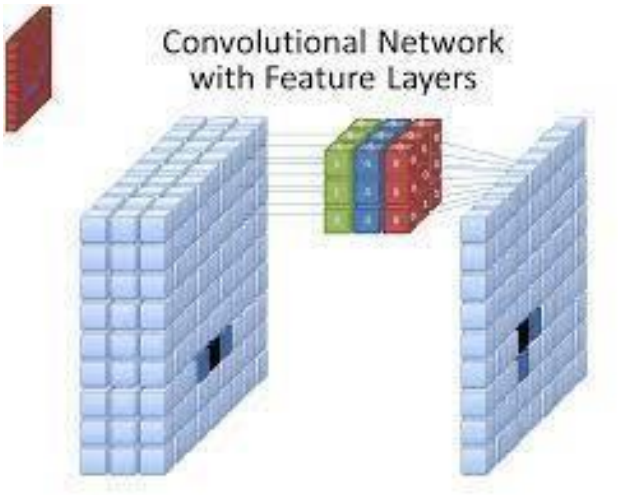


Fig 6 : Convolutional Network with feature layers

While the filter size covers the height and width of the filter, the filter's depth must also be specified.

HOW DOES A 2D IMAGE HAVE DEPTH?

Digital images are rendered as height, width, and some RGB value that defines the pixel's colors, so the "depth" that is being tracked is the number of color channels the image has. Grayscale (non-color) images only have 1 color channel while color images have 3 depth channels. All of this means that for a filter of size 3 applied to a full-color image, the dimensions of that filter will be 3 x 3 x 3. For every pixel covered by that filter, the network multiplies the filter values with the values in the pixels themselves to get a numerical representation of that pixel. This process is then done for the entire image to achieve a complete representation. The filter is moved across the rest of the image according to a parameter called "stride", which defines how many pixels the filter is to be moved by after it calculates the value in its current position. A conventional stride size for a CNN is 2.The end result of all this calculation is a feature map. This process is typically done with more than one filter, which helps preserve the complexity of the image.

5.2 BLOCK DIAGRAM

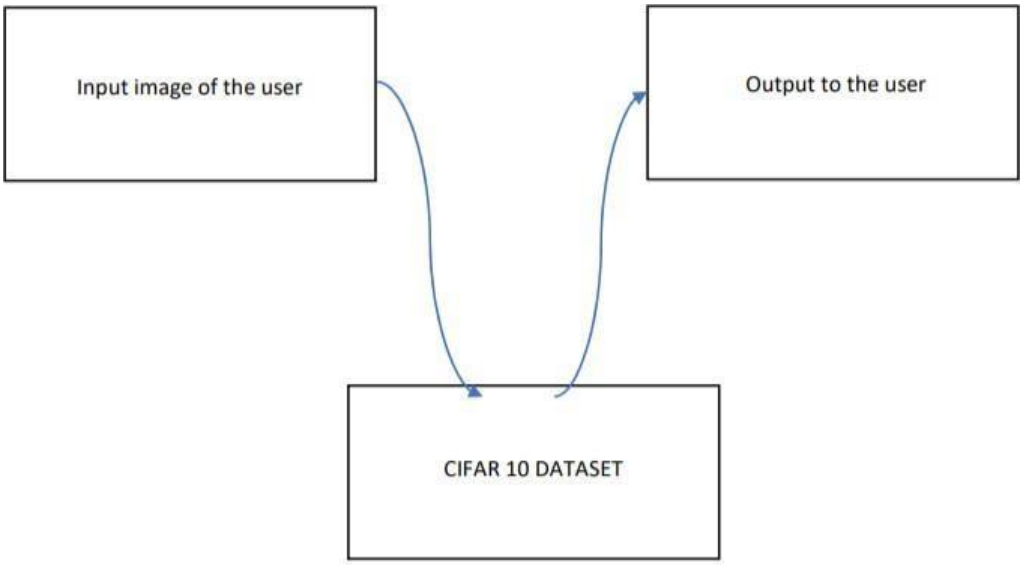
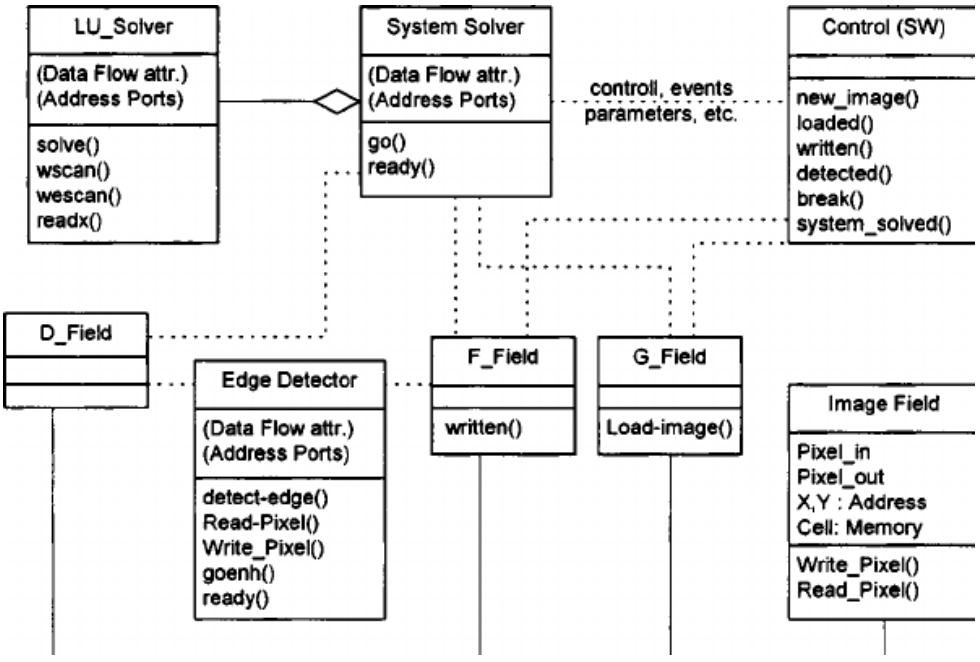


Fig 7 : System Block Diagram

5.3 CLASS DIAGRAM



5.3 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software system

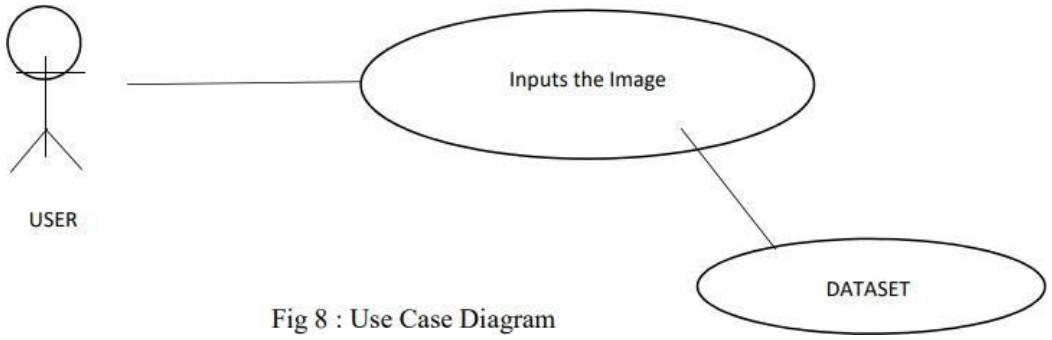


Fig 8 : Use Case Diagram

5.4 SEQUENCE DIAGRAM :

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

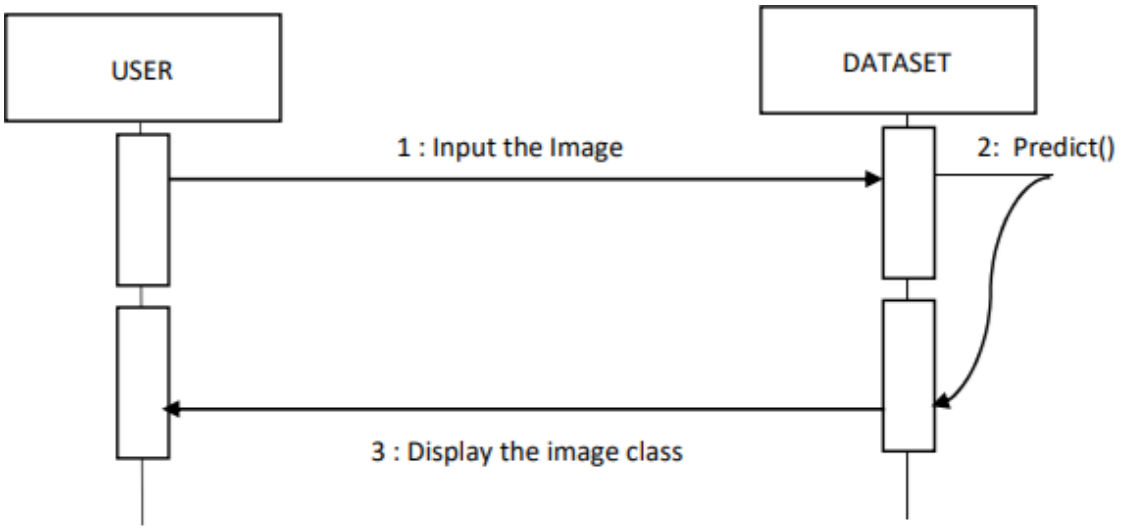


Fig 9 : Sequence Diagram

Chapter-6
IMPLEMENTATION

6.1 Python Imaging Library (PIL)

Python Imaging Library (PIL) (in newer versions known as Pillow) is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

Capabilities

Pillow offers several standard procedures for image manipulation. These include:

- per-pixel manipulations,
- masking and transparency handling,
- image filtering, such as blurring, contouring, smoothing, or edge finding,
- image enhancing, such as sharpening, adjusting brightness, contrast or color,
- adding text to images and much more.

Keras

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or TensorFlow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a Google engineer.

Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Backend. So Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano.

Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

6.2 CIFAR 10

The CIFAR-10 dataset (Canadian Institute for Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms. It is one of the most widely used datasets for machine learning research. 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. There are 6,000 images of each class. Computer algorithms for recognizing objects in photos often learn by example. CIFAR10 is a set of images that can be used to teach a computer how to recognize objects. Since the images in CIFAR-10 are low-resolution (32x32), this dataset can allow researchers to quickly try different algorithms to see what works. Various kinds of

convolutional neural networks tend to be the best at recognizing the images in CIFAR-10. CIFAR-10 is a labelled subset of the 80 million tiny images dataset. When the dataset was created, students were paid to label all of the images.

Note: As CIFAR 10 consists of 32*32 size images. We need to feed the images of 32*32 size only to detect the name of the image.

NumPy

NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc. NumPy array can also be used as an efficient multi-dimensional container for generic data. Now, let me tell you what exactly is a numpy array.

NumPy Array:

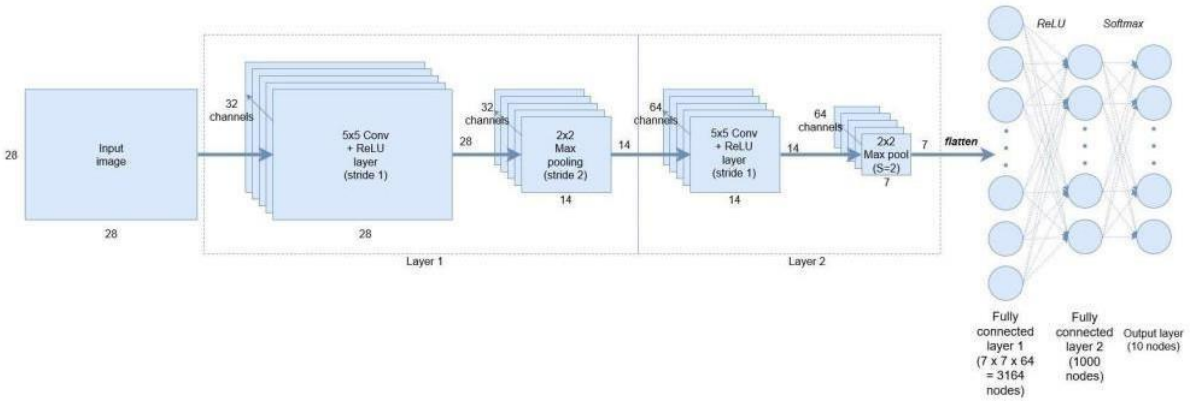
Numpy array is a powerful N-dimensional array object which is in the form of rows and columns. We can initialize numpy arrays from nested Python lists and access it

H5Py

The h5py package is a Pythonic interface to the HDF5 binary data format.
It lets you store huge amounts of numerical data, and easily manipulate that data from NumPy. For example, you can slice into multi-terabyte datasets stored on disk, as if they were real NumPy arrays. Thousands of datasets can be stored in a single file, categorized and tagged however you want. H5py uses straightforward NumPy and Python metaphors, like dictionary and NumPy array syntax. For example, you can iterate over datasets in a file, or check out the .shape or .dtype attributes of datasets.

Convolutional Neural Network (CNN)

After further research, we found that by using a Convolutional Neural Network can automatically extract features from the images by using adjacent pixel information with a window matrix and creating a convolution layer to learn patterns in the images to better classify the dataset. ¬ 4-Layer CNN Our images were 32x32x3: 32 pixels long, 32 pixels wide, and 3 pixels for red, green, and blue values. This model used the training set as the input layer, 4 convolutional layers, two that have 32 output channels and two thathave 64 output channels. For each



After further research, we found that by using a Convolutional Neural Network can automatically extract features from the images by using adjacent pixel information with a window matrix and creating a convolution layer to learn patterns in the images to better classify the dataset.

➢ 4-Layer CNN

Our images were 32x32x3: 32 pixels long, 32 pixels wide, and 3 pixels for red, green, and blue values. This model used the training set as the input layer, 4 convolutional layers, two that have 32 output channels and two that have 64 output channels. For each convolutional layer, we have decided on a kernel size of 3x3, a common option for image classification. It will scan a 3x3 window of the image at a time.

```
model.add(Conv2D(32, (3, 3), input_shape=(32, 32, 3), activation='relu',
padding='same', kernel_constraint=maxnorm(3)))
```

Next, we have selected the ReLu function for activation. The ReLu function takes max(0, x) to add non-linearity to the network. Another activation function that could be deployed would be the sigmoid function, but ReLu was chosen due to its efficiency.

Next, we have selected max pooling layers that are 2x2. These pooling layers will take 4 of the max values out of the convolutional layer for downsampling. This is done to prevent overfitting. model.add(MaxPooling2D(pool_size= (2,2)))

Lastly, the fully-connected output layer uses a softmax activation to calculate the loss. To test how the model trains with respect to epochs, we trained the model on 20, 50, and 100 epochs.

- **6-Layer CNN**

In the 6-layer CNN, the only difference is the addition of two layers with 128 output channels each

```
model.add(Conv2D(128,(3,3),padding='same',activation='relu'))
model.add(Dropout(0.2))model.add(Conv2D(128,(3,3),padding='same',activation='relu') )
model.add(MaxPooling2D(pool_size=(2,2)))
```

6.3 End-User Instructions:

1. First the user need to run the image_recognition_trainer.py file to train the model to recognise the image we upload.

2. After successfully training our model now the user need to run image_recognition_tester.py file to run the model for recognising the images and to upload images.

3. When we run the model the program prompts the user to specify the path of the image in your PC. After specifying the path of the image you want to recognise. Hit enter.

4. After hitting enter the image is processed by our model and the image is opened with the default image reader of your computer. And the recognised image name is displayed on the run console of the program.

Note: The images which we upload into the model must be of size 32*32 because the CIFAR 10 dataset consists of 32*32 images only (i.e. collect the images of 32*32 from any sources and mention that path of the image when prompted and then test the model to recognise your image).

Chapter-7
SYSTEM TESTING

7.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of Testing's:

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and

preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Maintenance

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

Results obtained after comparing all models with logistic regression model:

Taking all constructed models into account, we can see that the Convolutional Neural Network models perform the best. Much of the weights are centralized on the diagonal of the confusion matrices (shown below), representing correct predictions. The other two models, Logistic Regression and Random Forests, have confusion matrices that have elements on the matrices that are shaded darker on the off-diagonal compared to the Convolutional Neural Network models. These two models have higher misclassification rates. Specifically, these models both classified ‘airplanes’ and ‘ships’ relatively well. Similarly,

One major drawback that we had was the size of the dataset. CIFAR-10 had 50,000 training images and 10,000 testing images for each model to go through. As our team was training the models, we realized that we lacked computational power, despite using a GPU. As a result, we had to use a fraction of the training images in order to get results from our first models, it did the worst. Possibly due to the shortage of data, Logistic Regression did not perform as well as it could have. In our models, we chose a small set of parameters to perform grid search on to obtain the best parameters to train a given model. To get better results, our team could have considered using more parameters.

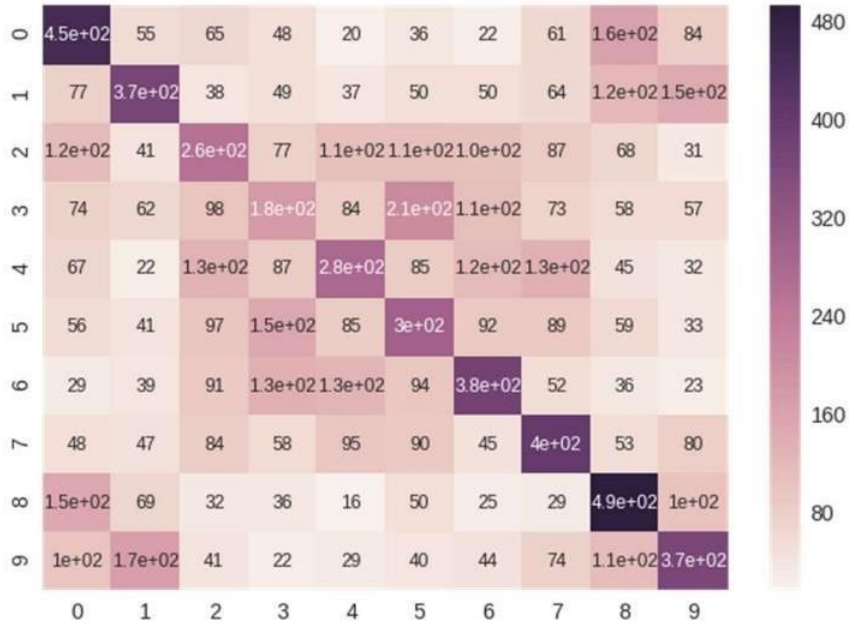


Figure I. Confusion Matrix for Multiclass Logistic Regression

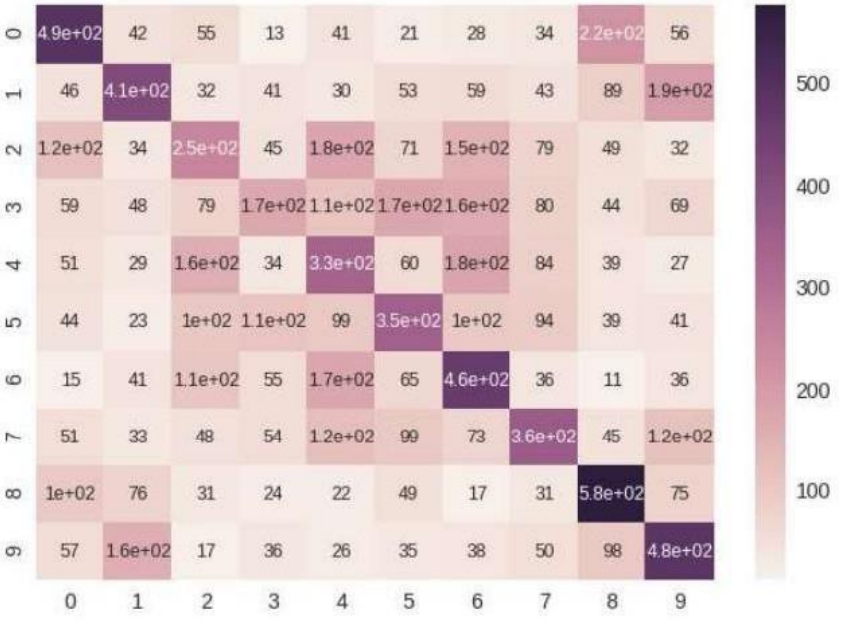


Figure II. Confusion Matrix for Random Forest Classifier

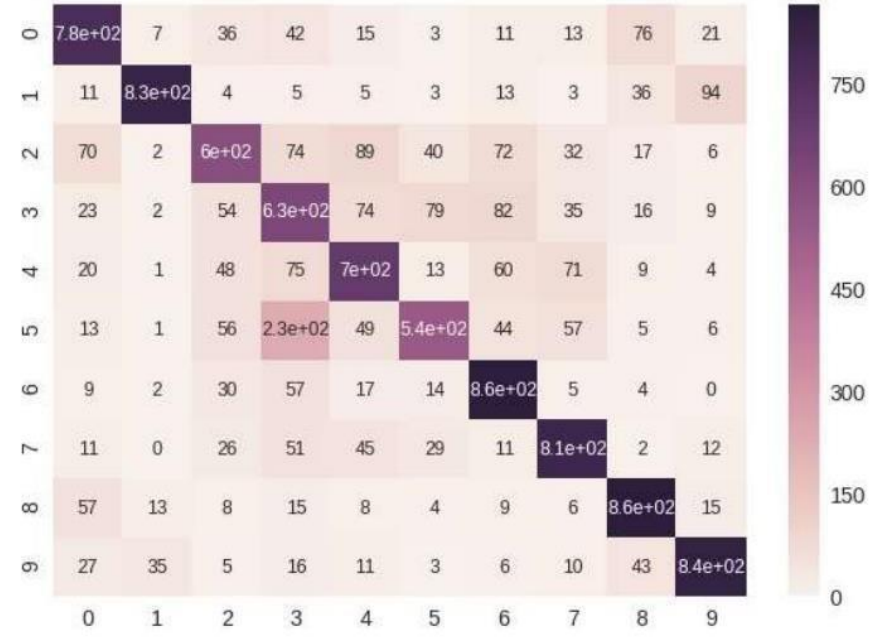
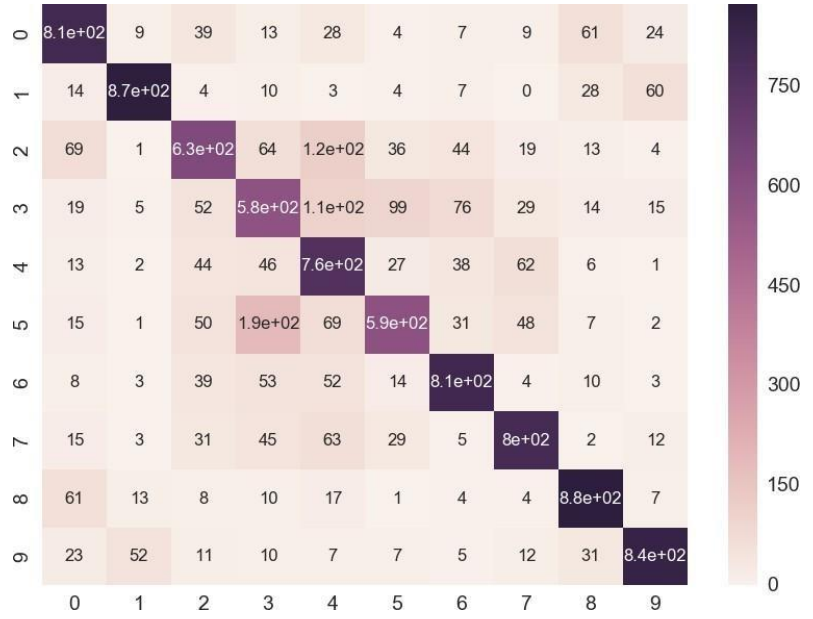


Figure III. Confusion Matrix for 4-Layered Convolutional Neural Network, 20 epochs



IV. Confusion Matrix for 4-Layered Convolutional Neural Network, 50 epochs

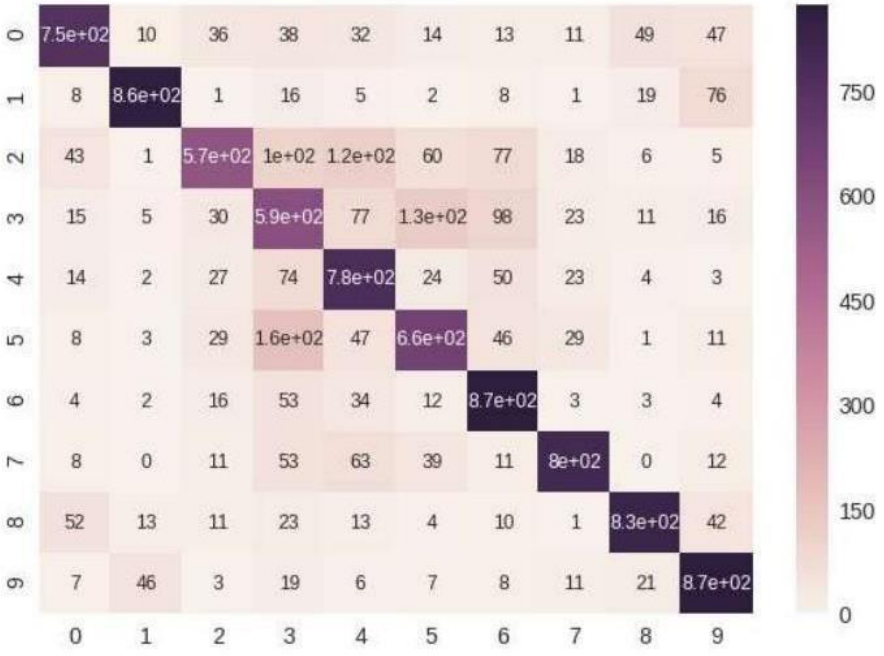


Figure V. Confusion Matrix for 4-Layered Convolutional Neural Network, 100 epochs

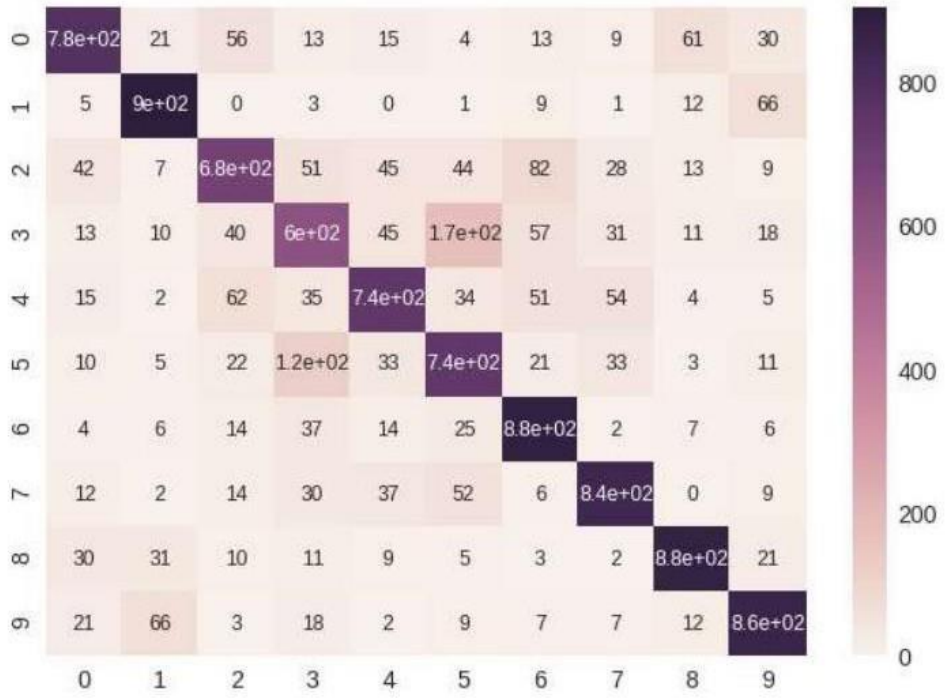


Figure VI. Confusion Matrix for 6-Layered Convolutional Neural Network, 20 epochs

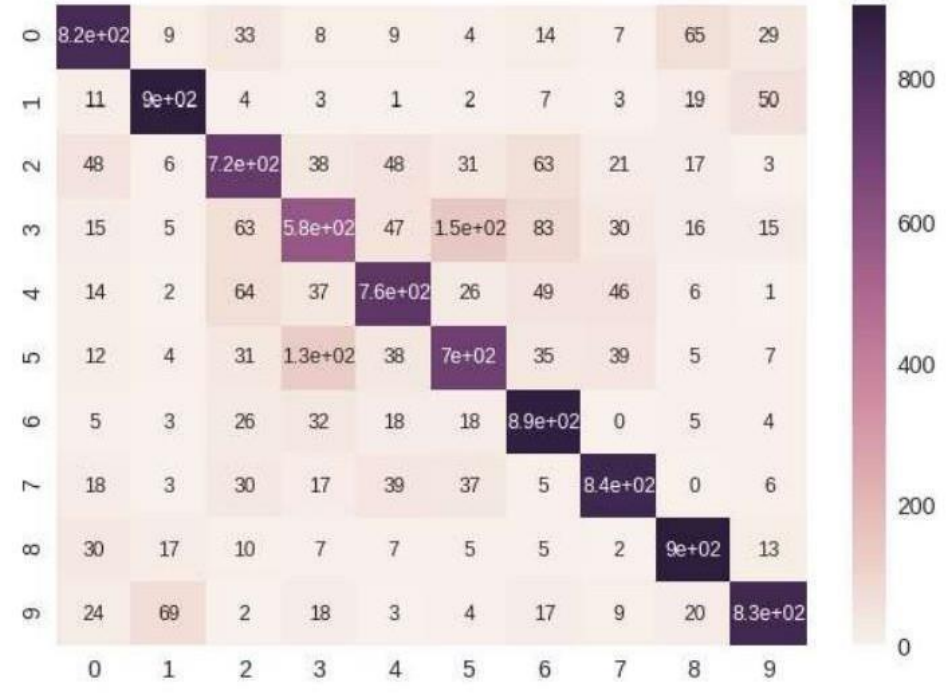


Figure VII. Confusion Matrix for 6-Layered Convolutional Neural Network, 50 epochs

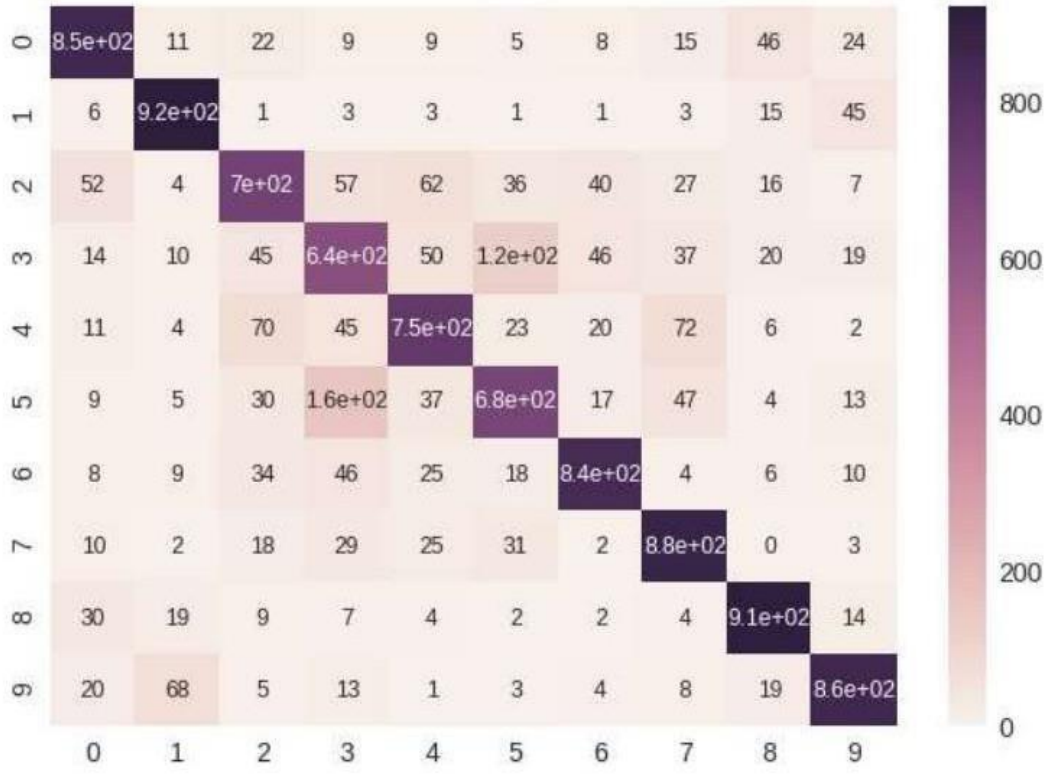


Figure VIII. Confusion Matrix for 6-Layered Convolutional Neural Network, 100 epochs

After referring to different protocols under machine learning like Logistic regression, Random Forest and Convolutional Neural Networks, CNN has been proved to be the best to choose among all the algorithms as this includes the concepts of Machine Learning. It also involves Deep Learning, as CNN is one of the concepts which arranges the images in the form of rows and columns in different records with similar elements in each class. So, CNN is the perfect mechanism to choose among the all algorithms as the customizable images can also be added and trained to allocate them to the respective classes

Chapter-8
SCREENS

ImageClassifier > Image)Recognition_Tester.py

Image_Reconition_Trainer.py | Image)Recognition_Tester.py

```
from PIL import Image
import numpy as np
from keras.models import load_model

labels = ['airplane', 'automobile', 'bird','cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']

model = load_model('Trained_model.h5')

input_path = input('Enter image file pathname: ')
input_image = Image.open(input_path)
input_image = input_image.resize((32, 32), resample=Image.LANCZOS)
image_array = np.array(input_image)
image_array = image_array.astype('float32')
image_array /=255.0
image_array = image_array.reshape(1, 32, 32, 3)
answer = model.predict(image_array)
input_image.show()
print(labels[np.argmax(answer)])
```

Fig: Screenshot of Image_recognition_Tester.py code

ImageClassifier > Image_Reconition_Trainer.py

Image_Reconition_Trainer.py | Image)Recognition_Tester.py

```
from PIL import Image
from keras.datasets import cifar10
(X_train, y_train), (X_test,y_test) = cifar10.load_data()

from keras.utils import np_utils
new_X_train = X_train.astype('float32')
new_X_test = X_test.astype('float32')
new_X_train /= 255
new_X_test /= 255
new_Y_train = np_utils.to_categorical(y_train)
new_Y_test = np_utils.to_categorical(y_test)

from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.optimizers import SGD
from keras.constraints import maxnorm

model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(32, 32, 3), activation='relu', padding='same', kernel_constraint=maxnorm(3)))
model.add(MaxPooling2D(pool_size= (2,2)))
model.add(Flatten())
model.add(Dense(512, activation='relu', kernel_constraint=maxnorm(3)))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer=SGD(lr=0.01), metrics=['accuracy'])
model.fit(new_X_train, new_Y_train, epochs=10, batch_size=32)

import h5py
model.save('trained_model.h5')

```

Fig: Screenshot of Image_Recognition_Trainer.py code

Fig: Screenshot of output of the model.

Chapter-9
CONCLUSION

CONCLUSION

Through these four models, we can see how each image classification model fares against each other. Simpler models like Logistic Regression and Random Forest showed that they were able to classify some classes relatively well, such as ‘airplanes’ and ‘ships’, but strongly misclassified ‘cats’. Our best model was the Convolutional Neural Network, which had a highly distinguishable diagonal on the confusion matrix, indicating high classification rates.

Given higher computational power, our team believes that we can achieve better results, especially for the models that did poorly. We would be able to add more parameters to our grid search to find the optimal model for Logistic Regression and Random Forest. It would also make it possible to utilize 100% of the dataset for all models.

In the future, we would like to explore more of image classification. We would like to improve on our models, through the methods stated above, and try to create models for the dataset CIFAR-100, with more 100 classes instead of 10.

Overall, we’ve come to a better understanding of how image classification works and what parameters and elements are crucial to each.

Hope it was easy to go through tutorial as I have tried to keep it short and simple. Beginners who are interested in Convolutional Neural Networks can start with this application. In short, you have learnt how to implement following concepts with python and Keras which are given below:

- Plotting images with matplotlib.
- Z-score (mean-std normalization) of images.
- Building a deep Convolutional Neural Network.
- Applying batch normalization.
- Regularization: Dropout & Kernel regularizers.
- Data Augmentation: ImageDataGenerator in Keras.
- Saving & Loading CNN models. The full python implementation of Image recognition

Chapter-10
REFERENCES

REFERENCES

1) Software Engineering - ESEC '93: 4th European Software Engineering Conference, Garmisch-Partenkirchen, Germany, September 13-17, 1993. Proceedings.

2) The Art of Software Testing, 3rd Edition.

3) Software Project Management by Bob Hughes and Mike Cotterell

4) Machine Learning in udemy.com 5) Github.com