A Major Project Phase II Report

on

CLASSIFICATION AND SEERATION OF WHATSAPP IMAGES USING MACHINE LEARNING

Submitted

In partial fulfillment of the requirements for award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

P.BALU MAHENDER **21W91A0499**

Under the Esteemed Guidance of

Mr. OWK. SRINIVASULU **M.Tech,(Ph.D).**

Associate Professor, ECE

Jawaharlal Nehru Technological University, Hyderabad

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY

(UGC AUTONOMOUS)

(Sponsored by Malla Reddy Educational society)

(Affiliated to JNTU, Hyderabad)

Maisammaguda, Dhulapally post, Secunderabad – 500014

2024-2025

MALLA REDDY
INSTITUTE OF ENGINEERING & TECHNOLOGY
(Autonomous Institution - UGC, Govt. of India)
(Sponsored by Malla Reddy Educational Society)
Approved by AICTE, New Delhi, Recognized Under 2(f) & 12(B)
Affiliated to JNTU, Hyderabad, Accredited by NBA & NAAC with 'A' Grade
Estd : 2007

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the major project phase II entitled "CLASSIFICATION AND SEPERATION OF WHATSAP IMAGES USING MACHINE LEARNING" that is being submitted by "P.BALU MAHENDER (21W91A0499)" under the guidance of Mr. OWK.SRINIVASULU M.Tech,(Ph.D) for the award of B.Tech Degree in ELECTRONICS AND COMMUNICATION ENGINEERING from the MALLA REDDY INSTITUTE OF ENGINEERING & TECHNOLOGY, Maisammaguda (Affiliated to JNTU Hyderabad) is a record of Bonafide work carried out by them under our guidance and supervision. The results embodied in this major project have not been submitted to any other university or institute for the award of any degree.

Project Guide

Mr. OWK. SRINIVASULU M.Tech,(Ph.D)

Department of ECE

HOD

External Examiner

Principal

Dr. P SRINIVAS

MALLA REDDY
INSTITUTE OF ENGINEERING & TECHNOLOGY
(Autonomous Institution - UGC, Govt. of India)
(Sponsored by Malla Reddy Educational Society)
Approved by AICTE, New Delhi, Recognized Under 2(f) & 12(B)
Affiliated to JNTU, Hyderabad, Accredited by NBA & NAAC with 'A' Grade

DECLARATION

I, P. BALU MAHENDER (21W91A0499) hereby declare that the major project phase II entitled "CLASSIFICATION AND SEPERATION OF WHATSAP IMAGES USING MACHINE LEARNING" is Bonafide work done and submitted under the guidance of Mr. OWK. SRINIVASULU in partial fulfillment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGINEERING.

DEPARTMENT OF ECE

P.BALU MAHENDER 21W91A0499

ACKNOWLEDGEMENT

I am very much thankful to Director, **Shri. P.PRAVEEN REDDY** for giving me this opportunity to do this major project. I express my deep sense of gratitude to him for his constant guidance and inspiring words.

I express my profound thanks to my Principal, **Dr. P. SRINIVAS**, for extending all the college facilities for the completion of the major project.

I would like to thank **Mr. OWK. SRINIVASULU M.Tech,(Ph.D)**, Associate Professor and Head of the Department of Electronics and Communication Engineering for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library, at anytime.

I feel highly obliged to my Major Project coordinator **Mr. S. RAMESH BABU**, Associate Professor and Project guide **Mr. OWK. SRINIVASULU(Ph.D)**, Associate Professor, Department of Electronics and Communication Engineering for their constant encouragement and moral support. They have been a source of valuable guidance, suggestions and kindness during the course of the project work. I find no words to express my gratitude and thanks to them.

I sincerely thank all the staff of the Department of Electronics and Communication Engineering, for their timely suggestions, healthy criticism and motivation during the course of our study. I would also like to thank my friends for always being there to provide required help or support. With great respect and affection, I thank my parents who were the backbone behind my deeds.

Finally, I express my immense gratitude with pleasure to one and all who have either directly or indirectly contributed to my need at right time for the development and execution of project work.

DEPARTMENT OF ECE

P. BALU MAHENDER 21W91A0499

CONTENTS

LIST OF FIGURES

ABSTRACT

In this digital era, Internet has become an integral part of human lives. Internet and social networks have become very popular, allowing anyone to easily share pictures, text, audio and video files. Among all the applications, WhatsApp has become quite famous due to its ease of use and it has replaced almost all the other messaging apps. Apart from sending messages, images and videos over it, one more reason for the heavy usage of WhatsApp is the exchange of study notes and materials by the students during the time of the examination and end up with a lot of images to be deleted at the end of each semester.

As the WhatsApp folder may have many other images, selecting the study material images, brochures, etc., one by one from the other images and then deleting them is a tedious process. Further, the proposed model classifies the study notes images into printed and handwritten notes. Notices and brochures received on WhatsApp are separated into a new folder. And also, screenshots and photos are grouped into separate folders. Traditional image classification methods are part of the field of artificial intelligence (AI), formally known as machine learning. Identification of the source social network based on the downloaded images is an important multimedia forensic task with significant cybersecurity implications in light of the sheer volume of images and videos shared across various social media platforms.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.

CHAPTER 1 - INTRODUCTION

1.1 INTRODUCTION

Classifications are systematically divided into groups and categories based on their characteristics. Image classification has emerged to narrow the gap between computer vision and human vision by training computers with data. Image classification is achieved by classifying images into predetermined categories based on the content of the vision. Motivated by [1], this article describes the study of image classification using deep learning. Traditional image classification methods are part of the field of artificial intelligence (AI), formally known as machine learning. Identification of the source social network based on the downloaded images is an important multimedia forensic task with significant cybersecurity implications in light of the sheer volume of images and videos shared across various social media platforms. Such a task has been proved possible by exploiting distinctive traces embedded in image content by Social Networks (SNs). To further advance the development of this area, we propose a novel framework, called Fusion NET, that integrates two established Convolutional Neural Networks, with the former (named 1D-CNN) learning discriminative features from the histogram of DCT coefficients and the latter (named 2D-CNN) inferring unique attributes from the sensor-related noise residual of the images in question. The separately learned features are then fused by Fusion NET to inform the ensuing source identification or source-oriented image classification component.

In this project we are using CNN (convolution neural networks) algorithm to classify Whatsapp images to different categories such as Question Paper, Mark sheets, Printed papers, hand written papers and circular. CNN algorithm will get trained on above mention categories to build a classification model. This model can be applied on test images to predict image type.

To train CNN we have used same dataset given by and below showing that dataset images

Identification of the source social network based on the downloaded images is an important multimedia forensic task with significant cybersecurity implications in light of the sheer volume of images and videos shared across various social media platforms. Such a task has been proved possible by exploiting distinctive traces embedded in image content by Social Networks (SNs). To further advance the development of this area, we propose a novel framework, called FusionNET, that integrates two established Convolutional Neural Networks, with the former (named 1D-CNN) learning discriminative features from the histogram of DCT coefficients and the latter (named 2D-CNN) inferring unique attributes from the sensor-related noise residual of the images in question. The separately learned features are then fused by FusionNET to inform the ensuing source identification or source-oriented image classification component. A series of experiments were conducted on a number of image datasets across various social networks and instant messaging apps (IMAs) to validate the feasibility of FusionNET also in comparison with the performance of the 1D-CNN and 2D-CNN. Encouraging results were observed.

Today, a simple search for an image on the Web can return thousands of related images. Some results are exact copies, some are variants (or near-duplicates) of the same digital image, and others are unrelated. Although we can recognize some of these images as being semantically similar, it is not as straightforward to find which image is the original. It is not easy either to find the chain of transformations used to create each modified version. There are several approaches in the literature to identify near-duplicate images, as well as to reconstruct their relational structure. For the latter, a common representation uses the parent-child relationship, allowing us to visualize the evolution of modifications as a phylogeny tree. However, most of the approaches are restricted to the case of finding the tree of evolution of the near-duplicates, with few works dealing with sets of trees. Since one set of near-duplicates can contain n independent subsets, it is necessary to reconstruct not only one phylogeny tree, but several trees that will compose a phylogeny forest. In this paper, through the analysis of the state-of-the-art image phylogeny algorithms, we introduce a novel approach to deal with phylogeny forests, based on different combinations of these algorithms, aiming at improving their reconstruction accuracy. We analyze the effectiveness of each combination and evaluate our method with more than 40 000 testing cases, using quantitative metrics.

1.2 MACHINE LEARNING

What is Machine Learning: -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Learning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.*Supervised learning* involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn?

The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

How to Start Learning Machine Learning?

Arthur Samuel coined the term "Machine Learning" in 1959 and defined it as a "Field of study that gives computers the capability to learn without being explicitly programmed".And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!

Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as Fork Python available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

* Terminologies of Machine Learning

- Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.

- Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

- Target (Label) – A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

- Training – The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

- Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are –

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

1.2.1 TYPES OF MACHINE LEARNING

Supervised Learning – This involves learning from a training dataset with labeled data using classification and regression models.

This learning process continues until the required level of performance is achieved.

- **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

1.3 ADVANTAGES OF MACHINE LEARNING

- **Easily identifies trends and patterns -**
  Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them.
- **No human intervention needed (automation)**
  With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.
- **Continuous Improvement**
  As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model.
- **Handling multi-dimensional and multi-variety data**
  Machine Learning algorithms are good at handling data that are multi- dimensional and multi-variety, and they can do this in dynamic or uncertain environments.
- **Wide Applications**
  You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customer.

1.4 DISADVANTAGES OF MACHINE LEARNING

- **Data Acquisition**
  Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality.
- **Time and Resources**
  ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.
- **Interpretation of Results**
  Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose. And when they do get noticed, it takes quite some time to recognize the source of the issue, even longer to correct it.
- **High error-susceptibility**
  Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. This leads to irrelevant advertisements being displayed to customers.

1.5 APPLICATIONS OF MACHINE LEARNING

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML

The separately learned features are then fused by FusionNET to inform the ensuing source identification or source-oriented image classification component. Identification of the source social network based on the downloaded images is an important multimedia forensic task with significant cybersecurity implications in light of the sheer volume of images and videos shared across various social media platforms. Such a task has been proved possible by exploiting distinctive traces embedded in image content by Social Networks (SNs). Encouraging results were observed. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition

CHAPTER 2 - LITERATURE SURVEY

2.1 CLASSIFICATION AND SEPERATION OF WHATSAPP IMAGES USING MACHINE LEARNING

In this digital era, Internet has become an integral part of human lives. Internet and social networks have become very popular, allowing anyone to easily share pictures, text, audio and video files. Among all the applications, WhatsApp has become quite famous due to its ease of use and it has replaced almost all the other messaging apps. Apart from sending messages, images and videos over it, one more reason for the heavy usage of WhatsApp is the exchange of study notes and materials by the students during the time of the examination and end up with a lot of images to be deleted at the end of each semester. And also the notices and brochures in every semester gets mixed up with other images and these needs to be separated for easy reference. As the WhatsApp folder may have many other images, selecting the study material images, brochures, etc., one by one from the other images and then deleting them is a tedious process. Henceforth, this research work has utilized machine learning to build a model for detecting and extracting the images from the WhatsApp images folder. Further, the proposed model classifies the study notes images into printed and handwritten notes. Notices and brochures received on WhatsApp are separated into a new folder. And also, screenshots and photos are grouped into separate folders. The proposed model has been built by using a deep learning concept called the Convolutional Neural Network [CNN] and by using Python's Keras library. It takes an image and decides its category and then the action is taken accordingly.

2.2 PYTHON

Python is an interpreted high-level programming language for general- purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels.

What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC.

I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

2.3 MODULES USED IN PROJECT

Tensor flow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy

Numpy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multidimensional container of generic data. Arbitrary data- types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python is an interpreted high-level programming language for general- purpose programming.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. • Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors.

2.4 INSTALL PYTHON STEP-BY-STEP IN WINDOWS AND MAC

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct provided by Python enables programmers to write both clear code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices. Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version.

2.5 RELATED WORK

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link:

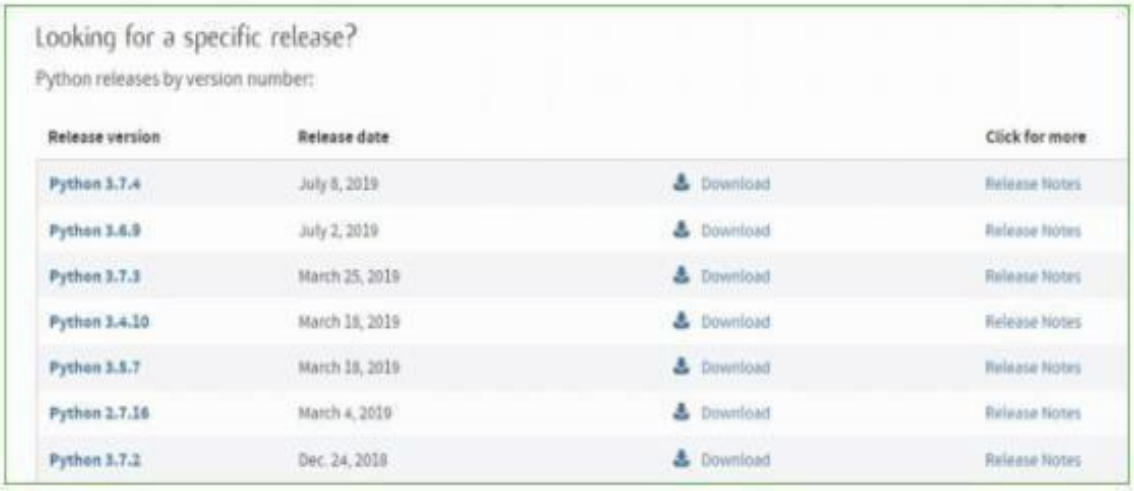
https://www.python.org

Now, check for the latest and the correct version for your operating system.

Step2: Click on the Download.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with

respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options:

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		[illegible]	[illegible]	SIG
XZ compressed source tarball	Source release		[illegible]	[illegible]	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	[illegible]	[illegible]	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	[illegible]	[illegible]	SIG
Windows help file	Windows		[illegible]	[illegible]	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	[illegible]	[illegible]	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	[illegible]	[illegible]	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	[illegible]	[illegible]	SIG
Windows x86 embeddable zip file	Windows		[illegible]	[illegible]	SIG
Windows x86 executable installer	Windows		[illegible]	[illegible]	SIG
Windows x86 web-based installer	Windows		[illegible]	[illegible]	SIG

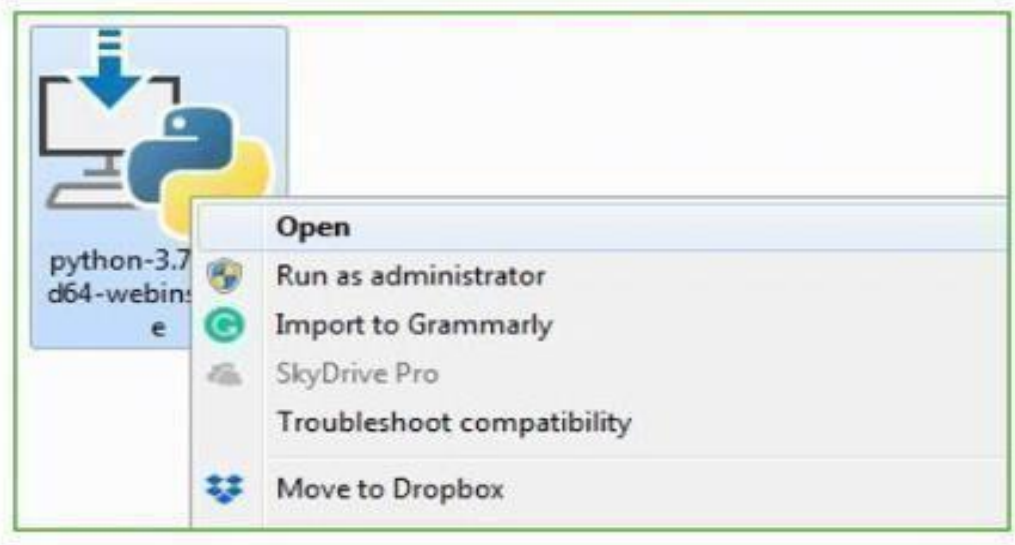
Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



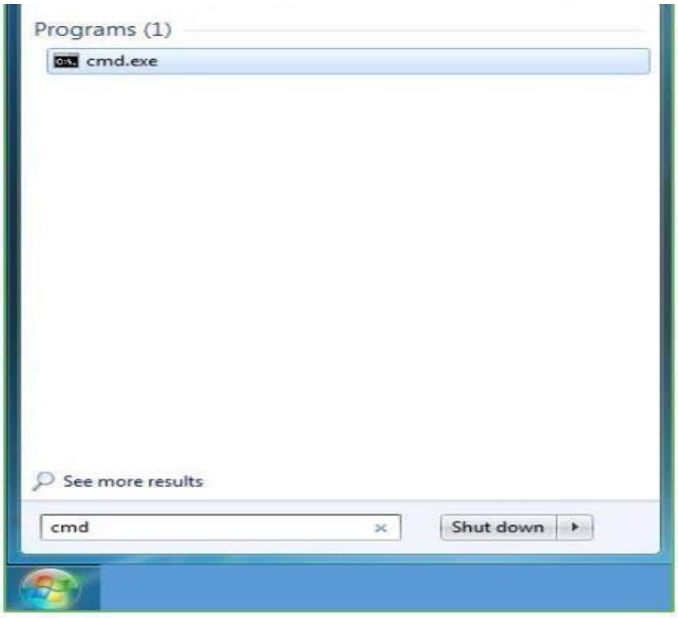
Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

Step 3: Click on Install NOW After the installation is successful. Click on Close. With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

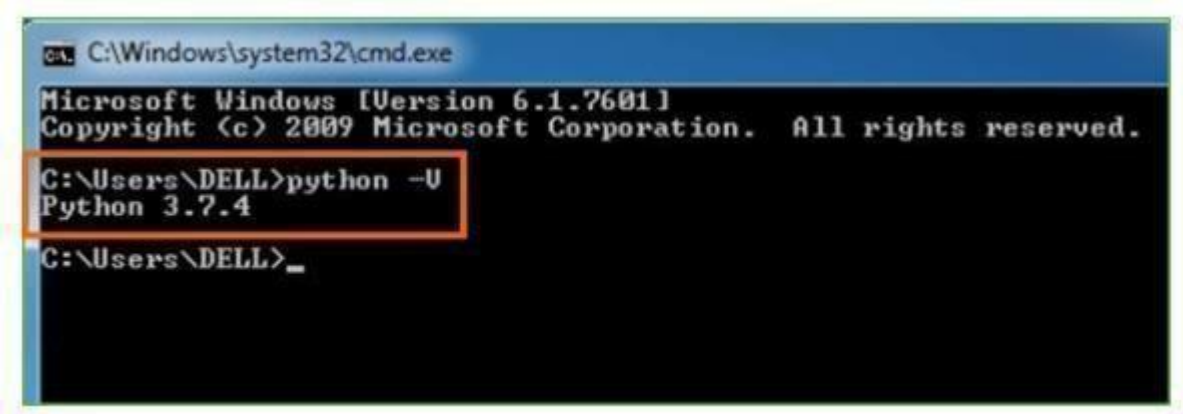
Step 1: Click on Start.



Step 2: In the Windows Run Command, type "cmd".

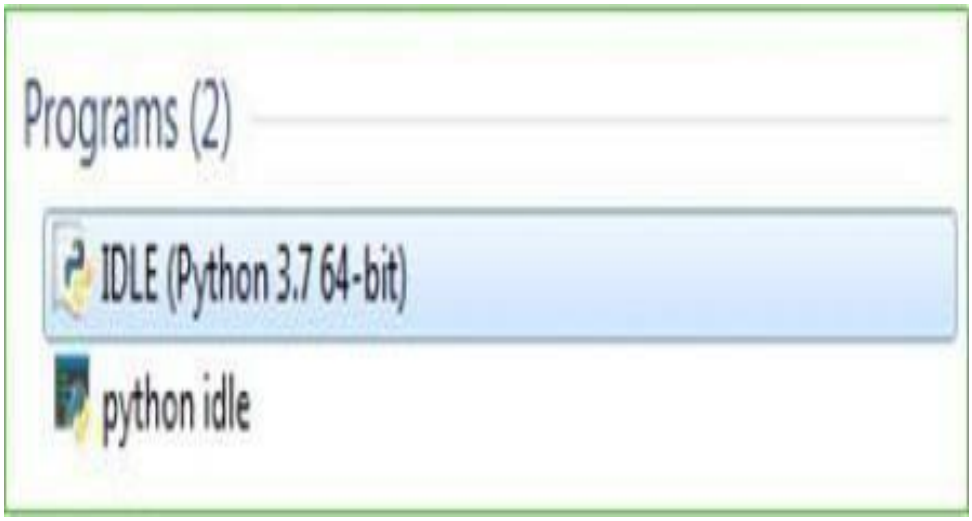
Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.
Step 5: You will get the answer as 3.7.4



Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

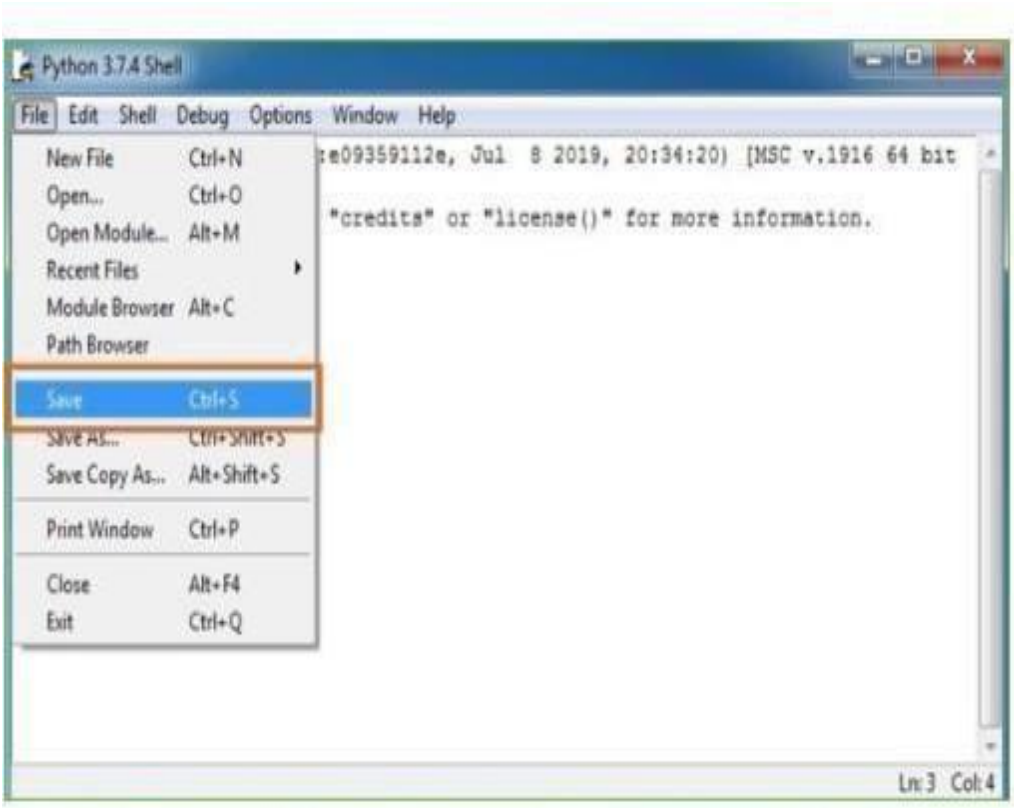
Check how the Python IDLE works



Step 1: Click on Start
Step 2: In the Windows Run command, type "python idle".
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program **Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**



The main aim of our work is to understand the performance of the networks for static as well as live video feeds. the primary step for the subsequent is to perform transfer learning on the networks with image datasets. this is often followed by checking the prediction rate of the identical object on static images and real-time video feeds. the various accuracy rates are observed and noted and presented within the tables given in further sections. The third essential criterion for assessing performance was to see if prediction accuracy differed among all CNNs used in the study. It must be noted that videos aren't used as a training dataset, they're used as testing datasets. Hence we are trying to find best image classifier where the object is the main attribute for classification of scene category.

Different layers of the convolutional neural network used are: Input Layer: The primary layer of each CNN used is ‘input layer’ which takes images, resize them for passing onto further layers for feature extraction. Convolution Layer: The next few layers are 'Convolution layers,' which operate as image filters, allowing you to extract features from images and calculate match feature points during testing.

Pooling Layer: The extracted feature sets are then passed to 'pooling layer'. This layer takes large images and shrink them down while preserving the foremost important information in them. It keeps the utmost value from each window, it preserves the simplest fits of each feature within the window. Rectified Linear Measured Layer: The next 'Rectified Linear Unit' or ReLU layer swaps every negative number of the pooling layer with 0. This keeps learnt values from being stuck near 0 or berating towards infinity, allowing the CNN to remain mathematically stable. Fully Connected Layer.

SYSTEM STUDY FEASIBILITY STUDY

At this factor, the goal's viability is evaluated, and a enterprise idea is presented in conjunction with a broadly used paintings plan and some cost estimates. The feasibility evaluation of the recommended system is to be finished throughout gadget analysis. This is to make sure that the agency is not continuously involved approximately the required device. A few specifics of the vital requirements for the writer are essential for usefulness appraisal.

The feasibility evaluate places a robust emphasis on three key problems: affordability, expediency, and usefulness of technology social usefulness cheap usefullness. This research is finished to evaluate the capacity financial effects of the maker at the commercial enterprise. The business enterprise has a restrained sum of money that may be allocated for gadget upkeep and studies. The costs have to be justified. Due to the truth that most of the technology used is quite simply available, the resourceful device in addition to the monetary resources have been consequently applied. Only the individualized merchandise had to be bought.

TECHNICAL FEASIBILITIES

This analysis is completed to determine the device's technical viability, or necessities. Any stepped forward device shouldn't require too a number of the available technological assets. The available technological sources will absolutely face heavy demands as a result. This will bring about the consumer having high expectancy. The built device ought to meet an affordable set of requirements due to the fact it could handiest be used with most efficient minimal or no changes.

Social software

The reason of the assessment component is to determine the diploma of consumer acceptability of the device. This involves the technique of educating and training the character on how to efficaciously function the equipment. The device should not make the person experience intimidated; as a substitute, it needs to be provided to them as a need. The techniques employed to train the character approximately the device and familiarize him with it have an immediate effect on the extent of recognition with the clients. Since he is the very last consumer of the gadget, his degree of self-warranty desires to be expanded so as for him to be powerful in submitting a few welcoming court cases.

CHAPTER 3 - EXISTING SYSTEM

3.1 EXISTING SYTEM

Recognizing information about the origin of a digital image has been individuated as a crucial task to be tackled by the image forensic scientific community. Understanding something on the previous history of an image could be strategic to address any successive assessment to be made on it: knowing the kind of device used for acquisition or, better, the model of the camera could focus investigations in a specific direction. Sometimes just revealing that a determined post-processing, such as an interpolation or a filtering, has been performed on an image could be of fundamental importance to go back to its provenance. This paper locates in such a context and proposes an innovative method to inquire if an image derives from a social network and, in particular, try to distinguish from, which one has been downloaded.

The technique is based on the assumption that each social network applies a peculiar and mostly unknown manipulation that, however, leaves some distinctive traces on the image; such traces can be extracted to feature every platform. By resorting at trained classifiers, the presented methodology is satisfactorily able to discern different social network origins. Experimental results carried out on diverse image datasets and in various operative conditions witness that such a distinction is possible. In addition, the proposed method is also able to go back to the original JPEG quality factor the image had before being uploaded on a social network. Our system will operate according to the system architecture depicted in the diagram below, capturing images either through a digital camera or through a database. For the next step, each image will be normalised to a predetermined size. We employ feature extraction approaches as M-BTC (Block Transition Coding), Histogram Equlization, and others to reduncate dimentianality. Feature vectors are formed by extracting features from a picture using various approaches such as MBTC (Block Transition Coding), Histogram Equlization, and so on.

The NN will be given this processed image to use in the classification process.There are several existing systems and tools that can help classify and organize images from WhatsApp or similar image-based applications using machine learning and image processing techniques. These systems typically rely on advanced machine learning models, specifically deep learning (like CNNs), to automate the process of categorizing and organizing images. For a fully managed, cloud-based system that can classify and separate WhatsApp images automatically, services like Google Photos, Amazon Rekognition, Clarifai, Microsoft Azure, and IBM Watson are great options. They all offer pre-trained models and image categorization features. Several systems and solutions have been developed to classify and organize images, including those from WhatsApp. These systems use machine learning models, such as deep learning-based convolutional neural networks (CNNs), to identify patterns, objects, and scenes in images. Below are some of the existing systems that can be adapted for WhatsApp image classification:

❖ **Google Photos**

- **Overview**: Google Photos offers a highly effective automatic image classification system, leveraging **deep learning** and **image recognition** to automatically organize images into categories like "Selfies," "Nature," "Food," "Documents," etc.
- **Features**:

o **Object and Scene Recognition**: Google Photos can identify objects and scenes in images, such as people, landscapes, food, and more.

o **Face Recognition**: It categorizes images by identifying faces and grouping them by individuals. o **Search and Organization**: Google Photos allows users to search for images based on keywords (e.g., "beach," "dog," "wedding").

o **Cloud Storage**: All images are uploaded to the cloud and automatically categorized.

How it relates to WhatsApp: WhatsApp images, once backed up to Google Photos, can be automatically classified using Google's machine learning models, making it easy to separate images into different categories.

- **Limitations**:

o **Lack of customization**: Google Photos' classifications are generalized, meaning that users cannot define custom categories like "WhatsApp-specific images" or organize them in a personalized way. o **Privacy Concerns**: Requires cloud storage, and users may not want their images uploaded to Google's servers.

❖ **Amazon Rekognition**

- **Overview**: Amazon Rekognition is a cloud-based machine learning service that can recognize and classify objects and scenes in images. It also offers features like facial recognition and text detection (OCR).
- **Features**:

 Custom Labels: Rekognition allows you to train the model to recognize custom categories tailored to your needs.

- **Facial and Object Detection**: It can detect faces, scenes (e.g., beach, city), and objects (e.g., car, tree).
- **Text Recognition**: It can detect text in images (useful for documents from WhatsApp).

How it relates to WhatsApp: WhatsApp images can be uploaded to Amazon S3 or directly analyzed through the Rekognition API. The system can automatically classify WhatsApp images based on predefined labels like "document," "selfie," or "food."

Limitations:

- **Costs**: Amazon Rekognition charges based on the number of images processed, which could become expensive at scale.
- **Dependency on Cloud**: Requires cloud storage for processing, which may not be ideal for privacy-conscious users.

- **Overview**: Clarifai is an AI-powered image recognition platform that uses machine learning models to classify and organize images.
- **Features**:

Pre-trained Models: Clarifai provides various pre-trained models for general image recognition, such as food detection, object classification, and people detection.

Custom Models: You can train a custom model based on your own labeled dataset.

API and SDKs: Clarifai offers APIs and SDKs for easy integration into applications.

How it relates to WhatsApp: You can integrate Clarifai into a WhatsApp image management system. By uploading WhatsApp images to the Clarifai API, you can classify them and separate them into categories like "landscape," "selfies," and "documents."

- **Limitations**:

- **Cloud Dependency**: Requires an internet connection to upload images to the cloud for classification.
- **Limited Customization**: While custom models are possible, the free tier comes with limitations on the number of API calls.

- **Microsoft Azure Computer Vision**

Overview: Microsoft Azure offers a suite of AI-powered tools, including Computer Vision, that helps analyze and classify images. **Features**:

Image Classification: Azure's Vision API can classify images into categories based on common objects and scenes.

OCR (Optical Character Recognition): Can detect and extract text from images (useful for WhatsApp documents).

Custom Vision: Azure allows users to train models using their own dataset to classify images into custom categories.

- **How it relates to WhatsApp**: You can use Azure Computer Vision to classify WhatsApp images into categories and store them in organized folders. Images can be classified based on detected content, such as "selfies," "documents," or "landscapes."
- **Limitations**:
  - **Cloud-based**: Like other solutions, it requires internet connectivity and cloud storage.
  - **Complexity**: Requires setting up an Azure account and potentially using advanced features that require configuration.

3.2 DEEP LEARNING

Deep learning models include various image classification models used in real world applications. Many methods have been developed and are emerging. Therefore, here are some basics of other models compared to advanced CNNs. Deep Neural Networks (DNN) are used to train neural networks for regression and classification. DNN performance is not good for images due to its low accuracy. Convolutional Neural Networks (CNN) have proven to be very successful in image classification and objects. Identification, recognition, etc. Here, the results are highly optimized compared to DNN. This means that already trained models will be used in large datasets to get good results in the relevant work. However, the accuracy is high here and the time is shorter than other products. Deep Learning is transforming the way machines understand, learn, and interact with complex data.

Deep learning mimics neural networks of the human brain, it enables computers to autonomously uncover patterns and make informed decisions from vast amounts of unstructured data. Transfer learning is another approach used to reuse acquired knowledge. This means that already trained models will be used in large datasets to get good results in the relevant work. However, the accuracy is high here and the time is shorter than other products.

Deep Learning is transforming the way machines understand, learn, and interact with complex data. Deep learning mimics neural networks of the human brain, it enables computers to autonomously uncover patterns and make informed decisions from vast amounts of unstructured data. It consists of layers of interconnected nodes, or neurons, that collaborate to process input data. In a fully connected deep neural network, data flows through multiple layers, where each neuron performs nonlinear transformations, allowing the model to learn intricate representations of the data.

In a deep neural network, the input layer receives data, which passes through hidden layers that transform the data using nonlinear functions. The final output layer generates the model's prediction. Deep Neural Networks (DNN) are used to train neural networks for regression and classification. DNN performance is not good for images due to its low accuracy. Convolutional Neural Networks (CNN) have proven to be very successful in image classification and objects. Identification, recognition, etc. Here, the results are highly optimized compared to DNN. By resorting at trained classifiers, the presented methodology is satisfactorily able to discern different social network origins. Experimental results carried out on diverse image datasets and in various operative conditions witness that such a distinction is possible. In addition, the proposed method is also able to go back to the original JPEG quality factor the image had before being uploaded on a social network. Our system will operate according to the system architecture depicted in the diagram below, capturing images either through a digital camera or through a database. For the next step, each image will be normalised to a predetermined size. Deep learning mimics neural networks of the human brain, it enables computers to autonomously uncover patterns and make informed decisions from vast amounts of unstructured data. Transfer learning is another approach used to reuse acquired knowledge. Deep Neural Networks (DNN) are used to train neural networks for regression and classification. DNN performance is not good for images due to its low accuracy. Convolutional Neural Networks (CNN) have proven to be very successful in image classification and objects.

3.3 Deep Learning in Machine Learning Paradigms

- Supervised Learning: Neural networks learn from labeled data to predict or classify, using algorithms like CNNs and RNNs for tasks such as image recognition and language translation.

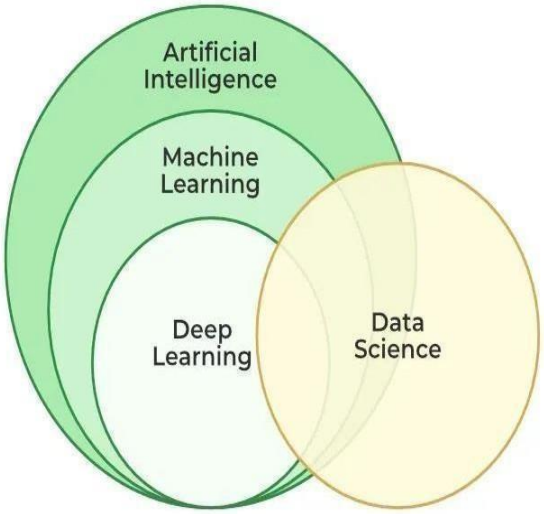


Fig 1 Reinforcement Learning

- Unsupervised Learning: Neural networks identify patterns in unlabeled data, using techniques like Autoencoders and Generative Models for tasks like clustering and anomaly detection.
- Reinforcement Learning: An agent learns to make decisions by maximizing rewards, with algorithms like DQN and DDPG applied in areas like robotics and game playing.

3.4 Challenges in Deep Learning

Deep learning has made significant advancements in various fields, but there are still some challenges that need to be addressed. Here are some of the main challenges in deep learning:

- **Data availability**: It requires large amounts of data to learn from. For using deep learning it's a big concern to gather as much data for training.
- **Computational Resources:** For training the deep learning model, it is computationally expensive because it requires specialized hardware like GPUs and TPUs.
- **Time-consuming:** While working on sequential data depending on the computational resource it can take very large even in days or months.
- **Interpretability:** Deep learning models are complex, it works like a black box. it is very difficult to interpret the result.
- **Overfitting:** when the model is trained again and again, it becomes too specialized for the training data, leading to overfitting and poor performance.

3.5 ADVANTAGES OF DEEP LEARNING

- **High accuracy:** Deep Learning algorithms can achieve state-of- the-art performance in various tasks, such as image recognition and natural language processing.
- **Automated feature engineering:** Deep Learning algorithms can automatically discover and learn relevant features from data without the need for manual feature engineering.
- **Scalability:** Deep Learning models can scale to handle large and complex datasets, and can learn from massive amounts of data.
- **Flexibility:** Deep Learning models can be applied to a wide range of tasks and can handle various types of data, such as images, text, and speech.
- **Continual improvement:** Deep Learning models can continually improve their performance as more data becomes available.

3.6 DISADVANTAGES OF DEEP LEARNING

- **High computational requirements:** Deep Learning AI models require large amounts of data and computational resources to train and optimize.
- **Requires large amounts of labeled data**: Deep Learning models often require a large amount of labeled data for training, which can be expensive and time- consuming to acquire.
- **Interpretability:** Deep Learning models can be challenging to interpret, making it difficult to understand how they make decisions.
- **Overfitting:** Deep Learning models can sometimes overfit to the training data, resulting in poor performance on new and unseen data.
- **Black-box nature**: Deep Learning models are often treated as black boxes, making it difficult to understand how they work and how they arrived at their prediction

CHAPTER-4 PROPOSED SYSTEM

4.1 CNN (CONVOLUTION NEURAL NETWORKS)

In this project we are using CNN (convolution neural networks) algorithm to classify Whatsapp images to different categories such as Question Paper, Mark sheets, Printed papers, hand written papers and circular. CNN algorithm will get trained on above mention categories to build a classification model. This model can be applied on test images to predict image type.

To train CNN we have used same dataset given by and below showing that dataset images The purpose of classification and separation of WhatsApp images using machine learning is to automate the process of organizing and categorizing images received through the WhatsApp messaging platform. This project aims to leverage machine learning techniques to analyze the content of WhatsApp images and classify them into specific categories or groups based on their visual features. The ultimate goal is to streamline the image management process and enable users to easily locate and access specific types of images within their WhatsApp media library.

4.2 OBJECTIVES

By implementing image classification and separation using machine learning, the project aims to achieve the following objectives:

- Efficient Organization
- Customizable Categories
- Visual Feature Extraction
- Automation and Time-Saving
- Scalability and Adaptability

4.3 SYSTEM ARCHITECTURE

The purpose of classification and separation of WhatsApp images using machine learning is to automate the process of organizing and categorizing images received through the WhatsApp messaging platform. This project aims to leverage machine learning techniques to analyze the content of WhatsApp images and classify them into specific categories or groups based on their visual features.

This project aims to leverage machine learning techniques to analyze the content of WhatsApp images and classify them into specific categories or groups based on their visual features.

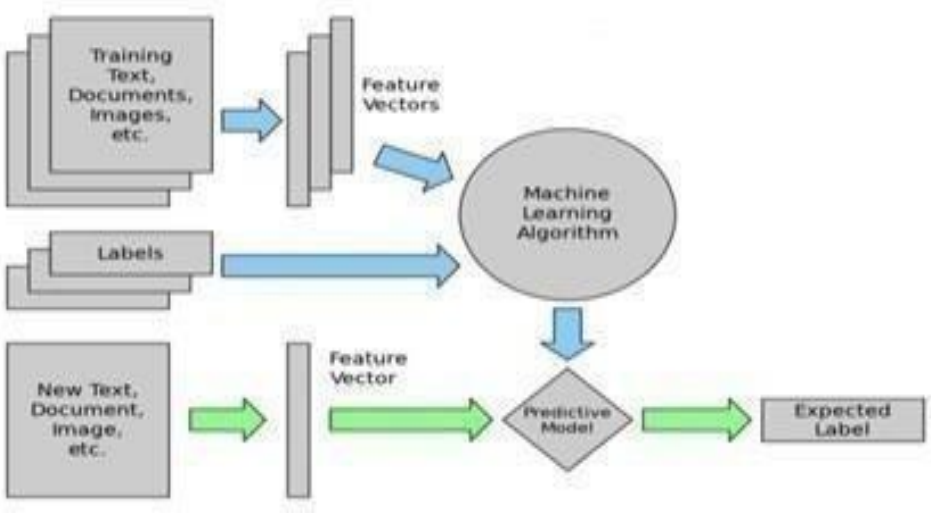


Fig 2 System Architecture

The ultimate goal is to streamline the image management process and enable users to easily locate and access specific types of images within their WhatsApp media library.

Implementation:

- **Image Classification Techniques:** Numerous studies have explored image classification techniques for organizing and categorizing images in various domains. Convolutional Neural Networks (CNNs) have been widely adopted for their ability to learn and extract meaningful features from images.
- **Feature Extraction and Representation:** Image feature extraction plays a vital role in image classification and separation tasks. Researchers have explored different feature extraction techniques, such as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Speeded-Up Robust Features (SURF).
- **Deep Learning Approaches:** Deep learning techniques have shown remarkable success in various computer vision tasks, including image classification. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been applied to capture temporal dependencies and context in WhatsApp image sequences. Additionally, Generative Adversarial Networks (GANs) have been used to generate new images and enhance image quality, contributing to better image separation and organization.

- **Semantic Segmentation**: Semantic segmentation involves assigning class labels to each pixel in an image, allowing for detailed separation and categorization. Fully Convolutional Networks (FCNs) and U-Net architectures have been employed for semantic segmentation tasks on WhatsApp images. This approach enables fine grained separation backgrounds, facilitating of objects and more precise classification and organization.

- **Data Augmentation and Preprocessing:** Data augmentation techniques have been employed to address limited labeled data availability. Augmentation methods, such as rotation, flipping, and scaling, help generate additional training samples, reducing the risk of overfitting and improving classification performance. Preprocessing steps, including resizing, normalization, and denoising, have also been applied to enhance image quality and remove unwanted artifacts.

MODULES:

To implement this project we have designed following modules

1) Upload Whatsapp Image Dataset: using this module we will upload dataset to application and then read all images and categories from dataset

2) Preprocess Dataset: using this module we will resize all images to equal size and then normalize image pixel values and then shuffle the dataset. After processing all images will be split into train and test where application using 80% dataset for training and 20% for testing

3) Train CNN Algorithm: using this module we will input 80% dataset to CNN to trained a model and then 20% test images will be applied on trained model to calculate prediction accuracy

4) CNN Training Graph: using this module we will plot CNN training and loss graph

5) Whatsapp Image Classification: using this module we will upload test image and then

CNN will predict or classify type of images:

- **Evaluation Metrics:** To assess the performance of image classification and separation models, various evaluation metrics have been used. Common metrics include accuracy, precision, recall, and F1 score. Some studies have also employed Intersection over Union (IoU) or Dice coefficient to evaluate the quality of image segmentation results.
- Some studies have also employed Intersection over Union (IoU) or Dice coefficient to evaluate the quality of image segmentation results. The classification and separation of WhatsApp images using machine learning techniques have witnessed substantial research and advancements in recent years.
- Researchers have explored image classification techniques, feature extraction and representation methods, deep learning approaches, semantic segmentation, data augmentation, preprocessing, and evaluation metrics.

- These efforts contribute to the development of effective models for automating the organization and categorization of WhatsApp images. By building upon the findings and methodologies from existing related work, researchers and practitioners can further enhance the accuracy and efficiency of image classification and separation algorithms in the context of WhatsApp.

CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import cv2
import random
import numpy as np
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from keras.callbacks import ModelCheckpoint
import pickle
import os
from keras.models import load_model
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

```
main = tkinter.Tk()
main.title("Classification & Separation of Whatsapp Images Using Machine Learning")
main.geometry("1300x1200")

global filename
global classifier
global labels, X, Y, X_train, y_train, X_test, y_test, cnn_model

def readLabels(filename):
    global labels
    labels = []
    for root, dirs, directory in os.walk(filename):
        for j in range(len(directory)):
            name = os.path.basename(root)
            if name not in labels:
                labels.append(name)



def uploadDataset():
    global filename
    global labels
    labels = []
    filename = filedialog.askdirectory(initialdir=".")
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n\n");
    readLabels(filename)
    text.insert(END,"Types of Images found in Whatsapp dataset are\n\n")
    for i in range(len(labels)):
        text.insert(END,labels[i]+"\n")

def processDataset():
    text.delete('1.0', END)
```

```
    global filename, X, Y, X_train, y_train, X_test, y_test
    if os.path.exists("model/X.txt.npy"):
        X = np.load('model/X.txt.npy')
        Y = np.load('model/Y.txt.npy')
    else:
        for root, dirs, directory in os.walk(filename):
            for j in range(len(directory)):
                name = os.path.basename(root)
                if 'Thumbs.db' not in directory[j]:
                    img = cv2.imread(root+"/"+directory[j])
                    img = cv2.resize(img, (32,32))
                    im2arr = np.array(img)
                    im2arr = im2arr.reshape(32,32,3)
                    X.append(im2arr)
                    label = getID(name)
                    Y.append(label)
                    print(name+" "+str(label))
        X = np.asarray(X)
        Y = np.asarray(Y)
        np.save('model/X.txt',X)
        np.save('model/Y.txt',Y)
    X = X.astype('float32')
    X = X/255
    text.insert(END,"Dataset Preprocessing Completed\n")
    text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n\n")
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    Y = to_categorical(Y)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2) #split dataset into train
and test
    text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n")
```

```
    text.insert(END,"80% images are used to train CNN : "+str(X_train.shape[0])+"\n")
    text.insert(END,"20% images are used to train CNN : "+str(X_test.shape[0])+"\n")


def trainCNN():
    text.delete('1.0', END)
    global X_train, y_train, X_test, y_test, cnn_model, labels
    #creating and initializing CNN object
    cnn_model = Sequential()
    #creating and adding CNN layer with 32 neurons and image size is of 3 dimension such as RGB and input shape will be image dimension
    #this layer filtered images 32 times
    cnn_model.add(Convolution2D(32, (3, 3), input_shape = (X_train.shape[1], X_train.shape[2], X_train.shape[3]), activation = 'relu'))
    #max pooling layer to collect relevant features from filtered image
    cnn_model.add(MaxPooling2D(pool_size = (2, 2)))
    #creating another layer to further filter images
    cnn_model.add(Convolution2D(32, (3, 3), activation = 'relu'))
    #another maxpooling layer
    cnn_model.add(MaxPooling2D(pool_size = (2, 2)))
    cnn_model.add(Flatten())#converting multi dimension array to 2 dimension arraay
    cnn_model.add(Dense(output_dim = 256, activation = 'relu'))#defining output layer
    cnn_model.add(Dense(output_dim = y_train.shape[1], activation = 'softmax')) #defining target prediction layer with y_train
    #compiling, training and loading model
    cnn_model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
    if os.path.exists("model/model_weights.hdf5") == False:
        model_check_point = ModelCheckpoint(filepath='model/model_weights.hdf5', verbose = 1, save_best_only = True)
        hist = cnn_model.fit(X_train, y_train, batch_size = 32, epochs = 20, validation_data=(X_test, y_test), callbacks=[model_check_point], verbose=1)
        f = open('model/history.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
```

```
    else:
        cnn_model.load_weights("model/model_weights.hdf5")
    predict = cnn_model.predict(X_test)
    predict = np.argmax(predict, axis=1)
    testY = np.argmax(y_test, axis=1)
    p = precision_score(testY, predict,average='macro') * 100
    r = recall_score(testY, predict,average='macro') * 100
    f = f1_score(testY, predict,average='macro') * 100
    a = accuracy_score(testY,predict)*100
    text.insert(END,"CNN Whatsapp Image Classification Accuracy  : "+str(a)+"\n")
    text.insert(END,"CNN Whatsapp Image Classification Precision : "+str(p)+"\n")
    text.insert(END,"CNN Whatsapp Image Classification Recall    : "+str(r)+"\n")
    text.insert(END,"CNN Whatsapp Image Classification FSCORE    : "+str(f)+"\n\n")
    conf_matrix = confusion_matrix(testY, predict)
    plt.figure(figsize =(6, 6))
    ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True,
cmap="viridis" ,fmt ="g");
    ax.set_ylim([0,len(labels)])
    plt.title("CNN Whatsapp Image Classification Confusion matrix")
    plt.ylabel('True class')
    plt.xlabel('Predicted class')
    plt.show()

def graph():
    f = open('model/history.pckl', 'rb')
    graph = pickle.load(f)
    f.close()
    accuracy = graph['val_accuracy']
    error = graph['val_loss']

    plt.figure(figsize=(10,6))
    plt.grid(True)
    plt.xlabel('EPOCH')
```

```
    plt.ylabel('Accuracy/Loss')
    plt.plot(accuracy, 'ro-', color = 'green')
    plt.plot(error, 'ro-', color = 'red')
    plt.legend(['CNN Accuracy', 'CNN Loss'], loc='upper left')
    plt.title('CNN Whatsapp Image Classification Training Accuracy & Loss Graph')
    plt.show()


def classifyImages():
    global cnn_model, labels
    filename = filedialog.askopenfilename(initialdir="testImages")
    image = cv2.imread(filename)
    img = cv2.resize(image, (32, 32))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1, 32, 32, 3)
    img = np.asarray(im2arr)
    img = img.astype('float32')
    img = img/255
    preds = cnn_model.predict(img)
    predict = np.argmax(preds)

    img = cv2.imread(filename)
    img = cv2.resize(img, (700,400))
    cv2.putText(img, 'Whatsapp Image Classified as : '+labels[predict], (10, 25),
cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 0, 255), 2)
    cv2.imshow('Whatsapp Image Classified as : '+labels[predict], img)
    cv2.waitKey(0)


def close():
    main.destroy()


font = ('times', 16, 'bold')
title = Label(main, text='Classification & Separation of Whatsapp Images Using Machine
Learning',anchor=W, justify=CENTER)
```

```
title.config(bg='yellow4', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 13, 'bold')
upload = Button(main, text="Upload Whatsapp Image Dataset", command=uploadDataset)
upload.place(x=50,y=100)
upload.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='yellow4', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=50,y=150)

processButton = Button(main, text="Preprocess Dataset", command=processDataset)
processButton.place(x=50,y=200)
processButton.config(font=font1)

trainButton = Button(main, text="Train CNN Algorithm", command=trainCNN)
trainButton.place(x=50,y=250)
trainButton.config(font=font1)

graphButton = Button(main, text="CNN Training Graph", command=graph)
graphButton.place(x=50,y=300)
graphButton.config(font=font1)

classifyButton = Button(main, text="Whatsapp Image Classification",
command=classifyImages)
classifyButton.place(x=50,y=350)
classifyButton.config(font=font1)

exitButton = Button(main, text="Exit", command=close)
```

```
exitButton.place(x=50,y=400)
exitButton.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=25,width=78)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=370,y=100)
text.config(font=font1)

main.config(bg='magenta3')
main.mainloop()
```

4.4 HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS :

- System : i3 or above.
- Ram : 4 GB.
- Hard Disk : 40 GB

SOFTWARE REQUIREMENTS :

- Operating system : Windows8 or Above.
- Coding Language : Python

Non- Functional Requirements:

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Usability requirement

- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement

- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement

SYSTEM TEST

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 TESTS

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.

System Test

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

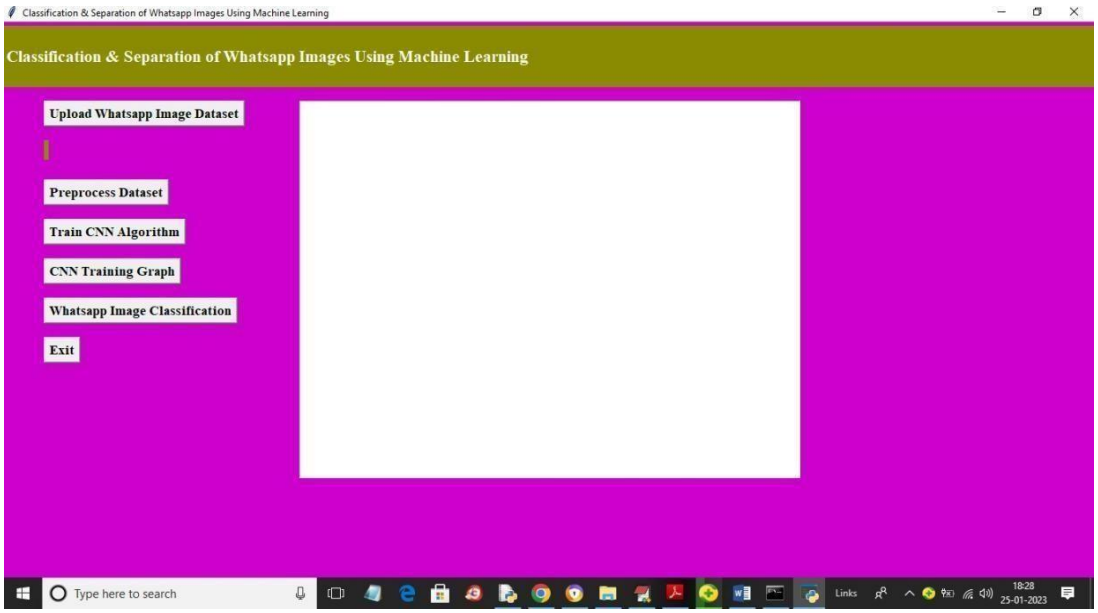
White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

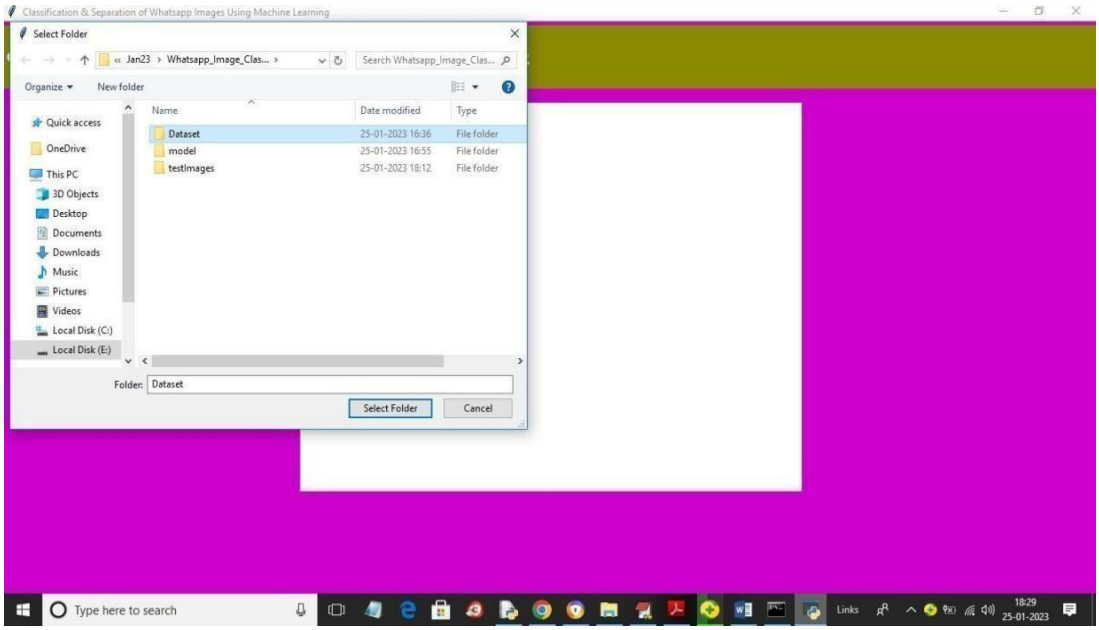
Black Box Testing

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.

CHAPTER 5 - RESULTS

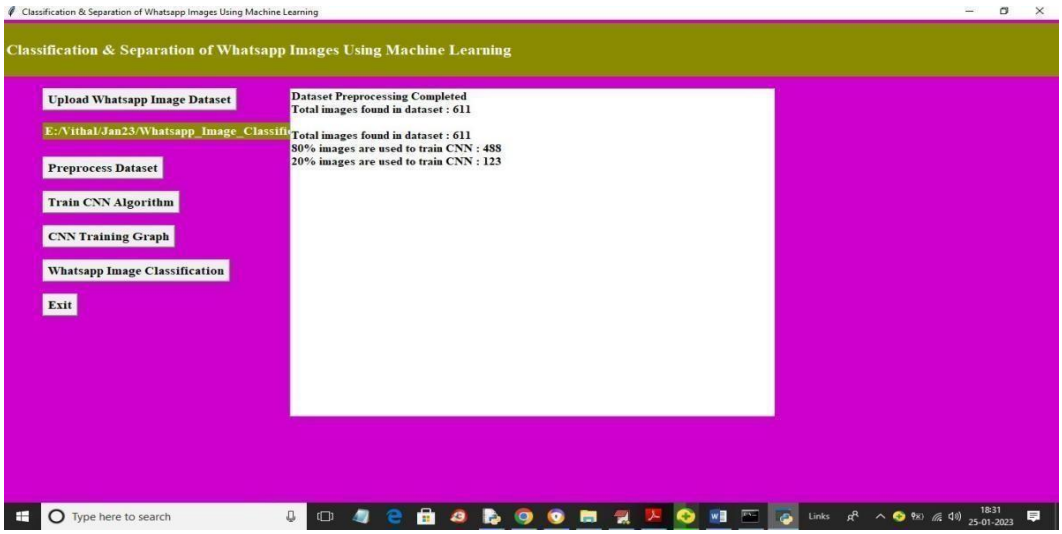
To run project double click on 'run.bat' file to get below screen. In above screen click on 'Upload Whatsapp Image Dataset' button to upload dataset and get below page

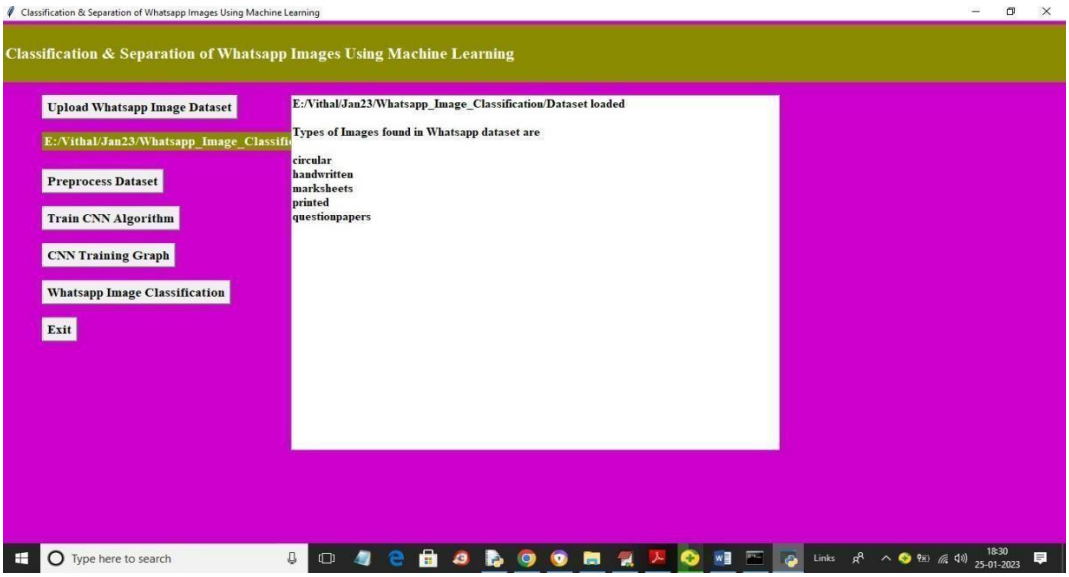




In above screen selecting and uploading 'Dataset' entire folder and then click on 'Select Folder' button to load dataset and get below page.

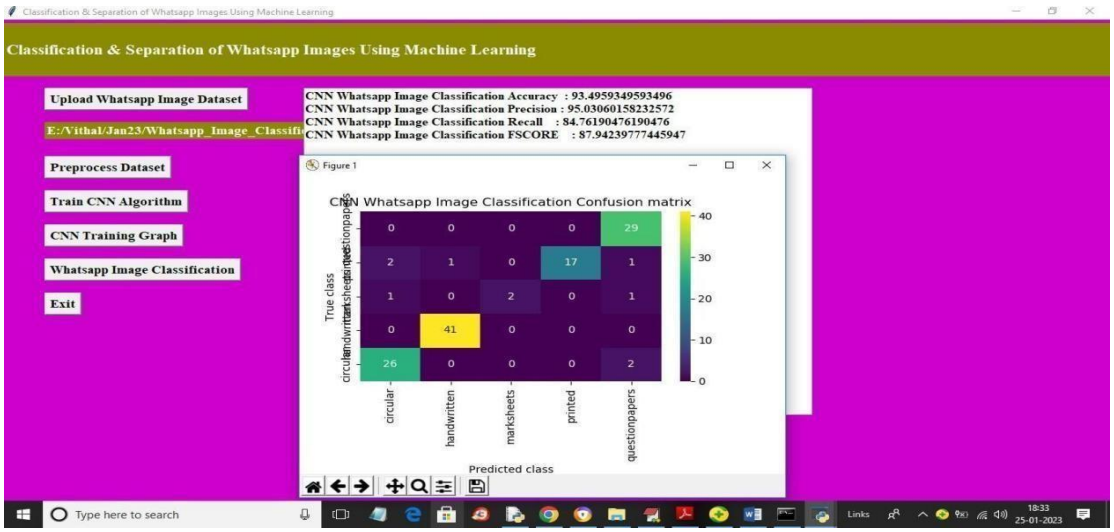
In above screen we can see dataset loaded and then we can see types of categories loaded and now click on ‘Preprocess Dataset’ button to resize, normalize, shuffle and split dataset into train and test

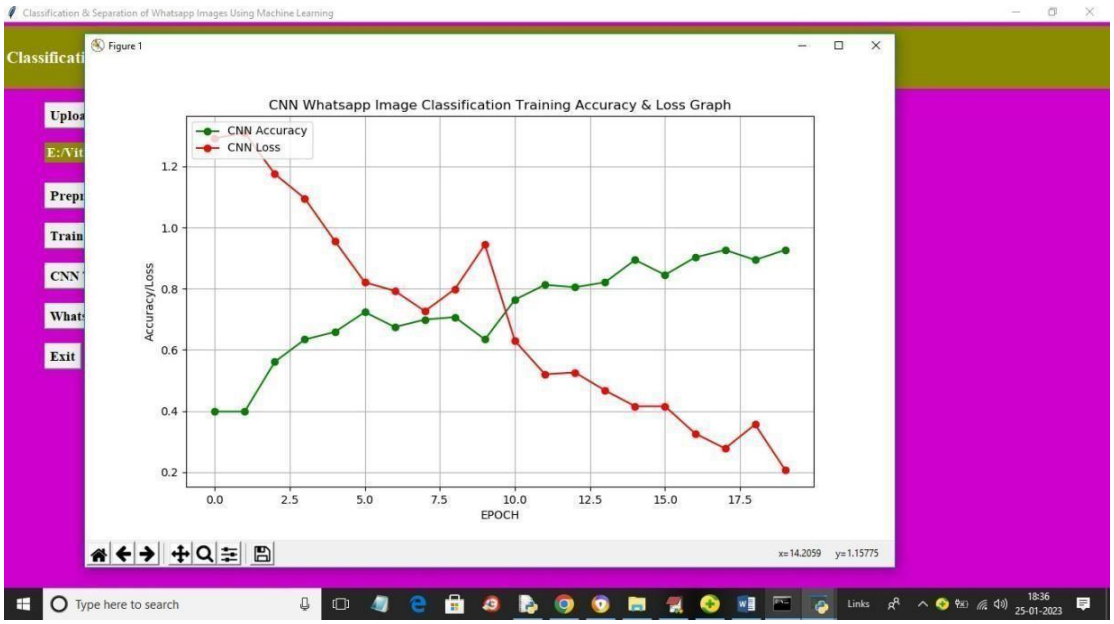




In above screen we can see application found total 611 images in the dataset and then process and then took 488 images for training and images for testing a 80 and 20%. Now click on ‘Train CNN Algorithm’ button to train CNN and get below output.

In above screen with CNN we got 93% accuracy and we can see precision, recall and FSCORE metric. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels.

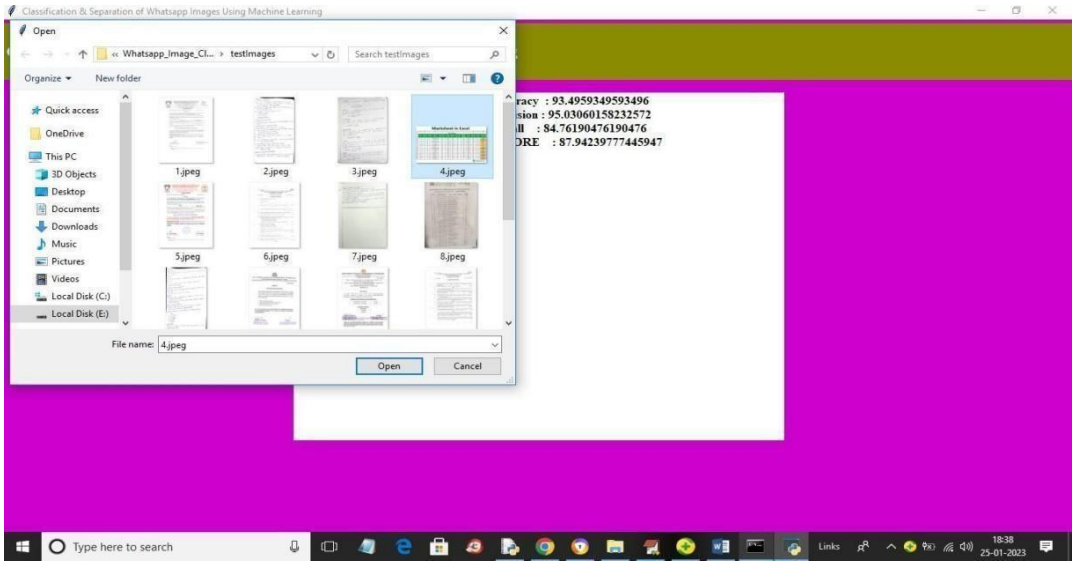


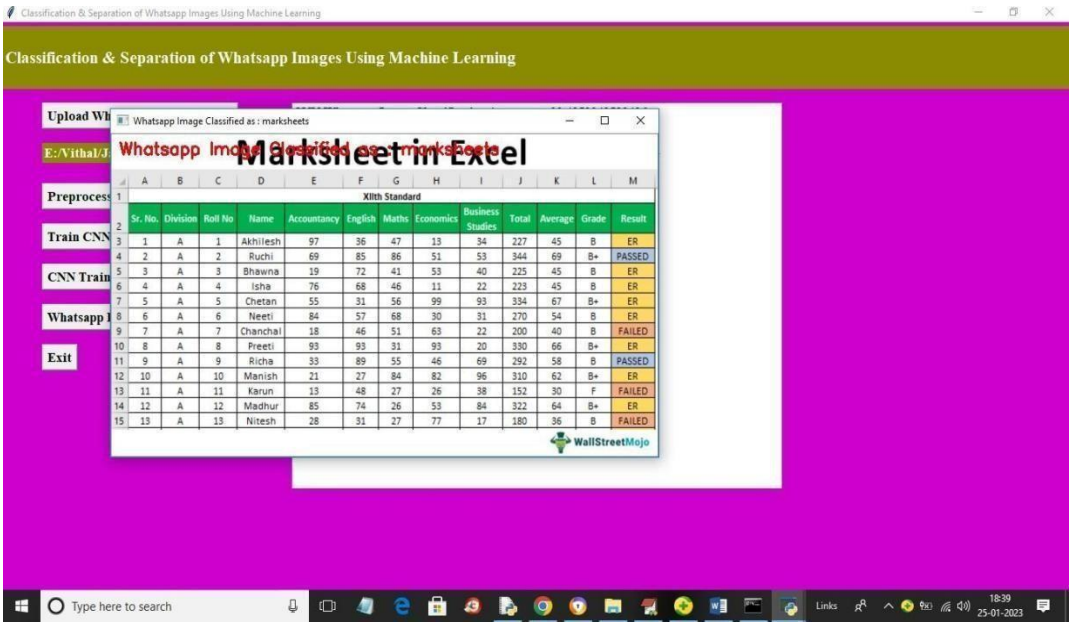


All blue colour boxes contains INCORRECT prediction count which are very few and different colour boxes contains CORRECT prediction count which are high in numbers so we got 93% accuracy. Now click on

'CNN Training Graph' button to get below page

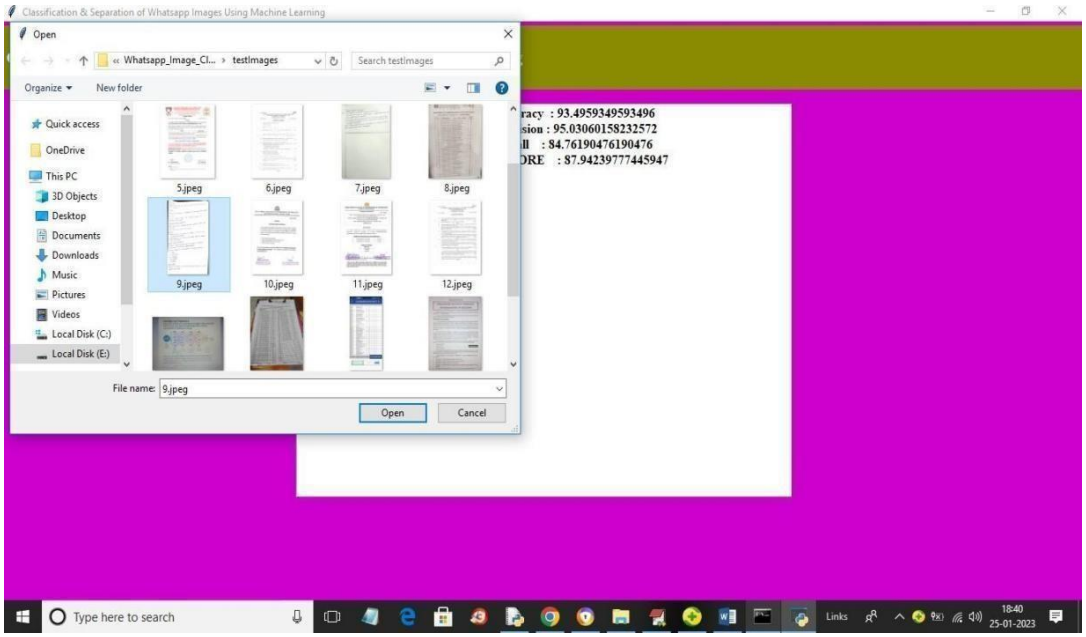
In above CNN training graph x-axis represents training epoch and y-axis represents accuracy and loss values. Green colour line represents Training Accuracy and red colour line represents Training LOSS and in above graph we can see with each increasing epoch accuracy got increase and reached closer to 1 and loss got decreased and reached closer to 1.

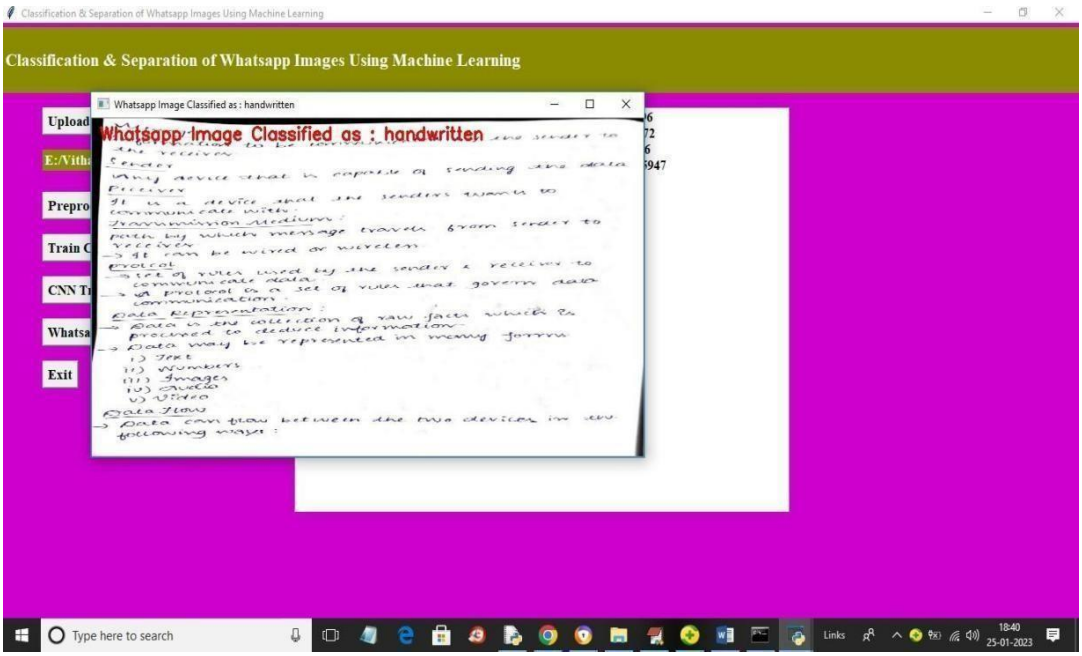




Now close above graph and then click on ‘Whatsapp Image Classification’ button to upload test image and get classification output. In above screen selecting and uploading 4.jpeg file and then click on ‘Open’ button to get below output

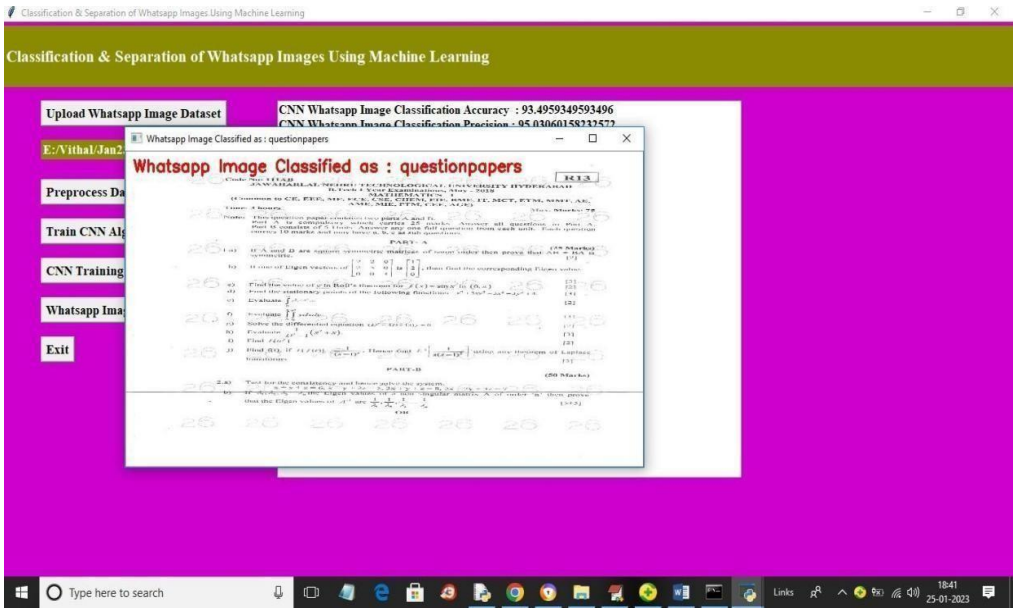
In above output image in red colour text or in image title you can see image classified as mark sheet. Similarly you can upload and test other images

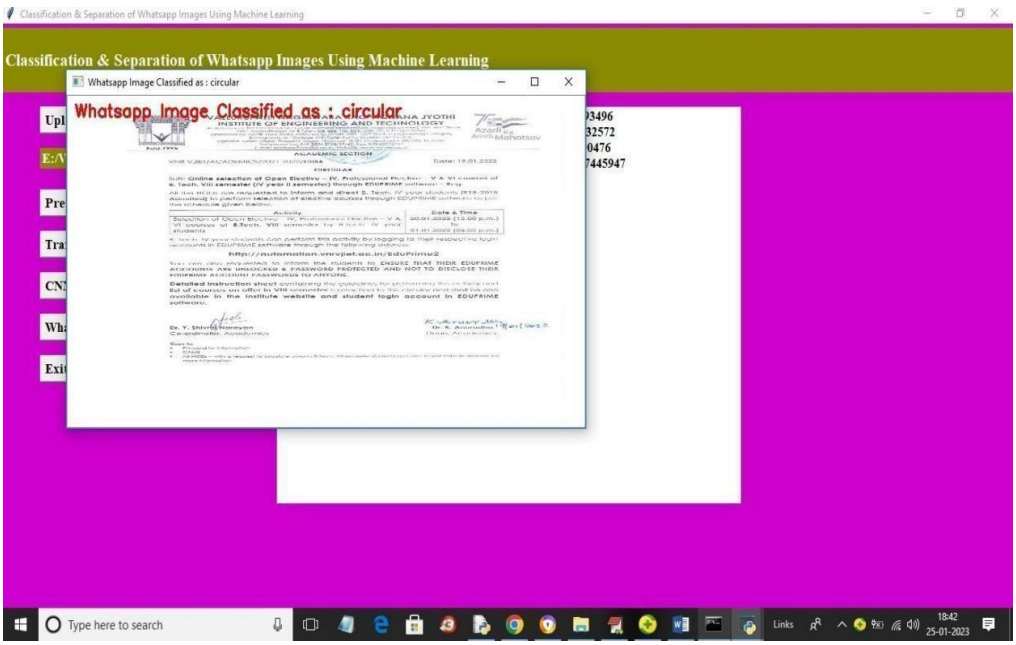




In above screen uploading another image and below is the output. In above screen image classified as ‘hand written’

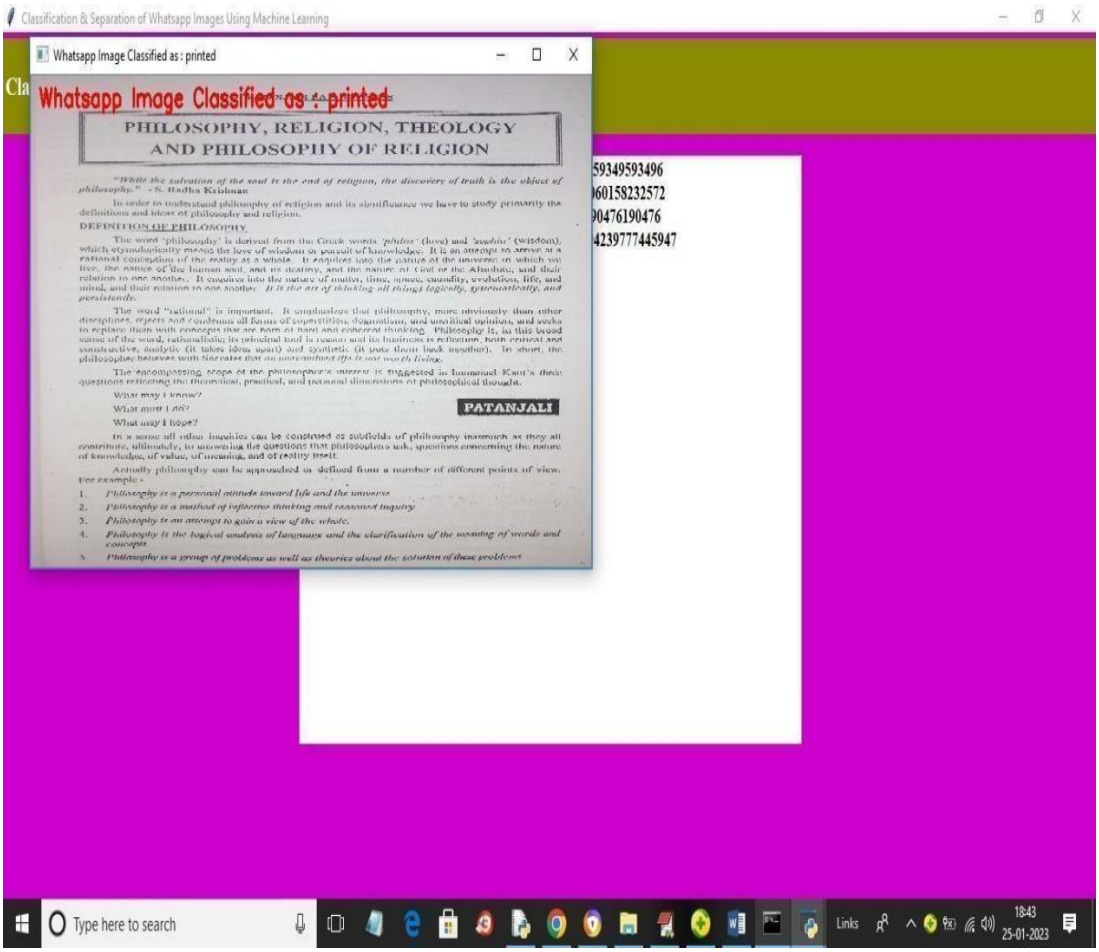
We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently.





Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data.



In this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data.

CHAPTER 6 – FUTURE SCOPE AND CONCLUSION

FUTURE SCOPE

- **Existing Systems**: Cloud-based solutions like Google Photos, Amazon Rekognition, and Microsoft Azure Computer Vision already provide image classification and organization, but they come with limitations such as lack of customization, privacy concerns, and cloud dependency.
- **Proposed System**: The proposed solution offers a customized, on-device classification system that can sort WhatsApp images into categories, ensuring privacy, control, and adaptability. It uses machine learning models trained on specific WhatsApp image data to classify and separate images into different folders locally, providing users with a more efficient and personalized way to organize their media. This proposed system addresses the gaps in existing solutions by providing more flexibility, privacy, and performance.

CONCLUSION

As a result, the proposed single sharing CNN method is very appropriate for categorizing images for simple search, and it has been successfully implemented. Using WhatsApp to disseminate false information is a significant societal issue. The accuracy of Fusion Net was superior to other conventional picture categorization techniques, as shown in the results and discussion section. In the future, this study intends to focus on combining message information (such as sender, timestamp, and group) with textual characteristics to better classify messages. A Deep Leaning method would be used to extract features from both text and media data in order to study multi-modal misinformation detection. Subsequently, the research urges to look at semi-automatic techniques forcreating constantly tagged WhatsApp datasets because misinformation changes over time.

REFERENCES

[1] Karuppusamy, P. "Building Detection using Two-Layered Novel Convolutional Neural Networks." Journal of Soft Computing Paradigm (JSCP) 3, no. 01 (2021): 29-37.

[2] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol. 11, no. Dec, pp. 3371–3408, 2010.

[3] Manoharan, J. Samuel. "Capsule Network Algorithm for Performance Optimization of Text Classification." Journal of Soft Computing Paradigm (JSCP) 3, no. 01 (2021): 1-9.

[4] G. Schaefer and M. Stich, "UCID - an uncompressed colour image database," in Proceedings of the Storage and Retrieval Methods and Applications for Multimedia, 2004, pp. 472–480. [

5] Manoharan, Samuel, and Narain Ponraj. "Analysis of Complex Non- Linear Environment Exploration in Speech Recognition by Hybrid Learning Technique." Journal of Innovative Image Processing (JIIP) 2, no. 04 (2020): 202-209.

[6] J. Yang, G. Zhu, and Y.-Q. Shi, "Analyzing the effect of jpeg compression on local variance of image intensity," Trans. Img. Proc., vol. 25, no. 6, pp. 2647–2656, Jun. 2016. [Online]. Available: http://dx.doi.org/10.1109/TIP.2016.2553521.

[7] Hamdan, Yasir Babiker. "Faultless Decision Making for False Information in Online: A Systematic Approach." Journal of Soft Computing Paradigm (JSCP) 2, no. 04 (2020): 226- 235

[8] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," J. Mach. Learn. Res., vol. 15, no. 1, pp. 3221–3245, Jan. 2014.

[9] Dhaya, R. "Hybrid Machine Learning Approach to Detect the Changes in SAR Images for Salvation of Spectral Constriction Problem." Journal of Innovative Image Processing (JIIP) 3, no. 02 (2021): 118-130.

[10] B. Tondi, "Pixel-domain adversarial examples against cnn-based manipulation detectors," Electronics Letters, vol. 54, pp. 1220–1222(2), October 2018. [Online].

Available: http://digital-library.theiet.org/content/journals/10.1049/el.2018.6469