# COLOR DETECTION USING PYTHON OPENCV

*A Project Report Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of*

##### BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

**T.JANAKAMMA(19NR1A05A9)**

Under the esteemed Guidance of

**MR.MURTHY SIR**

**Assistant Professor**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**BABA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Approved by A.I.C.T.E and Affiliated to J.N.T.U, Kakinada)**

**P.M. PALEM, MADHURWADA VISAKHAPATNAM-530048**

**(2019-2023)**



**.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

CERTIFICATE

This is to certify that the project work entitled “**COLOR DETECTION**” is Bonafide work done by, **T.JANAKAMMA(19NR1A05A9) ,**during the year 2019-2023 in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** from **BABA INSTITUTE OF TECHNOLOGY AND**

**SCIENCES**, (Affiliated to J.N.T.U. Kakinada), P.M Palem, Madhurwada, Visakhapatnam.

**Project Guide Head of the Department**

MR.MURTHY SIR MR. S.DURGA PRASAD

Assistant Professor Associate Professor



**BITS-VIZAG**

DECLARATION

I hereby declare that this project entitled “**COLOR DETECTION”** submitted to the department of **CSE,BABA INSTITUTE OF TECHNOLOGY AND SCIENCES,** Visakhapatnam for the partial fulfillment of the requirements for the award of B.Tech under the guidance of **MURTHY SIR** This work is not submitted to any University for the award of any degree or diploma.

##### T.JANAKAMMA(19NR1A05A9)

**ACKNOWLEDGMENT**

I would like to express our special thanks and gratitude to the management of our esteemed institute “**BABA INSTITUTE OF TECHNOLOGY AND SCIENCES**”, who gave us the golden opportunity to do this wonderful project and also helped us in doing a lot of research and we came to know about so many things.

I also wish to express our deep gratitude to **DR. N. VICTOR BABU**, Principal, BITS who gave his complete support for the fulfillment of our project.

I am grateful to **MR. S. DURGA PRASAD**, **Head of the Department, Department of Computer Science and Engineering** who gave his complete support for the fulfillment of our project.

I would like to express special thanks to our project guide **MR.MURTHY SIR**, **Assistant Professor, Department of Computer Science and Engineering** who gave us moral support and guided us in different matters regarding the topic. He had been very kind and patience while suggesting us the outlines of this project and correcting our doubts. We thank him for his overall support.

I am thankful to all teaching and non-teaching staff of the Department of Computer Science & Engineering for the cooperation given for the successful completion of our project.

##### T.JANAKAMMA(19NR1A05A9)

# ABSTRACT

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Color detection is the process of detecting name of the color. Here this is easy task for human to detect the color and choose one. But computer cannot detect the color easily. This is tough task for computer to detect the color easily. So that's why we choose this project. Many of the project and research papers are written on this problem. But we use different techniques for this project. Pandas and open CV libraries used in python languages. Open Source Computer Vision Library. Open CV was designed for computational efficiency and with a robust specialize in real-time platform that gives video and audio encoding infrastructure. Color detection [1] is the process of detecting name of the color. Here this is easy task for humans to’ detect the color and choose one. But computer cannot detect the color easily. This is tough task for computer to detect the color easily. So that’s why we choose this project. Many of the project and research papers are written on this problem. But we use different techniques for this project. Naive Bayes algorithm, Pandas and Open CV libraries used in python languages. Naive Bayes may be a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the category labels are drawn from some finite set. There’s not one algorithm for training such classifiers, but a family of algorithms supported a standard principle: all naive Bayes classifiers assume that the worth of a specific feature is independent of the worth of the other feature, given the category variable. Open Source Computer Vision Library. OpenCV was designed for computational efficiency [2] and with a robust specialize in real-time applications. Dedicated video encoding within the cloud. Panda may be a cloud-based platform that gives video and audio encoding infrastructure

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Pandas

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, International Journal of Scientific Research in Engineering and Management (IJSREM)

Volume: 05 Issue: 10 | Oct - 2021 ISSN: 2582-3930

© 2021, IJSREM | www.ijsrem.com | Page 1

COLOR DETECTION USING PANDAS AND OPENCV

Dr.C K GOMATHY, Ms. CHITTURI.CHAITANYA, Ms .B. JAI SAI ANVITHA, Ms. CHERAPALLI JAHNAVI

Sri Chandrasekharendra SaraswathiViswa Mahavidyalaya , Kanchipuram

ABSTRACT:-

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Panda

International Journal of Scientific Research in Engineering and Management (IJSREM)

Volume: 05 Issue: 10 | Oct - 2021 ISSN: 2582-3930

© 2021, IJSREM | www.ijsrem.com | Page 1

COLOR DETECTION USING PANDAS AND OPENCV

Dr.C K GOMATHY, Ms. CHITTURI.CHAITANYA, Ms .B. JAI SAI ANVITHA, Ms. CHERAPALLI JAHNAVI

Sri Chandrasekharendra SaraswathiViswa Mahavidyalaya , Kanchipuram

ABSTRACT:-

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Panda

P International Journal of Scientific Research in Engineering and Management (IJSREM)

Volume: 05 Issue: 10 | Oct - 2021 ISSN: 2582-3930

© 2021, IJSREM | www.ijsrem.com | Page 1

COLOR DETECTION USING PANDAS AND OPENCV

Dr.C K GOMATHY, Ms. CHITTURI.CHAITANYA, Ms .B. JAI SAI ANVITHA, Ms. CHERAPALLI JAHNAVI

Sri Chandrasekharendra SaraswathiViswa Mahavidyalaya , Kanchipuram

ABSTRACT:-

Color detection is the process of

detecting name of the color. Here this is easy task

for human to detect the color and choose one. But

computer cannot detect the color easily. This is

tough task for computer to detect the color easily.

So that's why we choose this project. Many of the

project and research papers are written on this

problem. But we use different techniques for this

project. Pandas and openCV libraries used in

python languages. Open Source Computer Vision

Library. Open CV was designed for computational

efficiency and with a robust specialise in real-time

platform that gives video and audio encoding

infrastructure.

Keywords: Color Detection, Open CV, Panda

KKKKKKandas

Keywords: Color Detection, Open CV, Pandas

# INDEX

## CONTENTS Page No

1. [INTRODUCTION 1](#_TOC_250023)
   1. FEASABILITY STUDY 1
   2. [EXISTING SYSTEM 3](#_TOC_250022)
   3. [PROPOSED SYSTEM 3](#_TOC_250021)
2. SOFTWARE REQUIREMENT SPECIFICATION 4
   1. [INTRODUCTION 4](#_TOC_250020)
   2. EXTERNAL INTERFACE REQUIREMNET 7
   3. [SYSTEM FEATURES 8](#_TOC_250019)
   4. [OTHER NON-FUNCTIONAL REQUIREMENTS 11](#_TOC_250018)
3. [ANALYSIS 12](#_TOC_250017)
   1. [USE CASE DIAGRAM 14](#_TOC_250016)
   2. [SEQUENCE DIAGRAM 15](#_TOC_250015)
   3. [COLLABORATION DIAGRAM 16](#_TOC_250014)
   4. [ACTIVITY DIAGRAM 16](#_TOC_250013)
4. [DESIGN 20](#_TOC_250012)
   1. [ARCHITECTURE DESIGN 20](#_TOC_250011)
   2. [PROJECT FLOW 22](#_TOC_250010)
   3. DATABASE DESGIN/MODEL 23
   4. [CLASS DIAGRAM 24](#_TOC_250009)
   5. [COMPONENT DIAGRAM 25](#_TOC_250008)
   6. DEPLOYEMENT DIAGRAM 26
5. [IMPLEMENTATION 27](#_TOC_250007)
   1. [INTRODUCTION TO TECHNOLOGY 27](#_TOC_250006)
   2. SAMPLE CODE 47
   3. [SCREENSHOTS 57](#_TOC_250005)
6. [TESTING 63](#_TOC_250004)
   1. [INTRODUCTION TO TESTING 63](#_TOC_250003)
   2. [SAMPLE TEST CASES 66](#_TOC_250002)
7. [CONCLUSION 67](#_TOC_250001)
8. FUTURE ENHANCMENTS 68
9. [REFERENCES 69](#_TOC_250000)

**LIST OF FIGURES**

**FIGURE TITLE Page No**

##### FIG.3.1.1 USE CASE DIAGRAM 14

**FIG.3.2.1 SEQUENCE DIAGRAM 15**

**FIG.3.3.1 COLLABORATION DIAGRAM 16**

**FIG.3.4.1 ACTIVITY DIAGRAM 17**

**FIG.4.1.1 ARCHITECTURE 21**

**FIG.4.2.1 PROJECT FLOW 22**

**FIG.4.3.1 DATABASE MODEL 23**

**FIG.4.5.1 CLASS DIAGRAM 24**

**FIG.4.5.1 COMPONENT DIAGRAM 25**

**FIG.4.6.1 DEPLOYMENT DIAGRAM 26**

**FIG.5.1.1.1 HISTORY OF PYTHON 28**

**FIG.5.1.5.1 PERCEPTRON 46**

**FIG.5.3.1 HOME PAGE 57**

**FIG.5.3.2 UPLOAD PAGE 57**

**FIG.5.3.3 DATASET 58**

**FIG.5.3.4 MODEL SELECTION 58**

**FIG.5.3.5 SVM ACCURACY 59**

**FIG.5.3.6 NEURAL NETWORKS ACCURACY 59**

**FIG.5.3.7 SVM-NN ACCURACY 60**

##### FIG.5.3.8 PREDICTION 60

##### FIG.5.3.9 OUTPUT 1 61

##### FIG.5.3.10 OUTPUT 2 61

##### FIG.5.3.11 GRAPH 62

# INTRODUCTION

Introduction to the Project:

Before going into the speculations of the

project it is important to know the definition of

color detection. It is simply the process of

identifying the name of any color. It is obvious that

humans performs this action naturally and do not

put any effort in doing so. While it is not the case

for computers. Human eyes and brain work in

coordination in order to translate light into color.

Light receptors that are present in eyes transmit the

signal to the brain which in turn recognizes the

color. There is no exaggeration in saying that

humans have mapped certain lights with their color

The process of detecting the name of any color in an image is known as color detection. This is an exceedingly simple task for humans, but it is not that simple for computers. The eyes and brains of humans work together to convert light into color. The signal is transmitted to the brain via light receptors in our eyes. The color is then recognized by our brain..

Before going into the speculations of the project it is important to know the definition of color detection. It is simply the process of identifying the name of any color. It is obvious that humans performs this action naturally and do not put any effort in doing so. While it is not the case for computers. Human eyes and brain work in coordination in order to translate light into color. Light receptors that are present in eyes transmit the signal to the brain which in turn recognizes the color. There is no exaggeration in saying that humans have mapped certain lights with their color names since childhood. The same strategy is useful in detecting color names in this project. Three different colors Red,Green and Blue are being tracked by utilising the fundamentals of computer vision. After successful compilation when we execute the code a window redirects the image displayed on it whose path is given as an argument. II. EXISTING SYSTEM In the existing system they have gone through with the opencv but while extraction of the colors they got the wrong outputs.There is no exact color representation of colors with accuracy..

**1.1 FEASIBILITY STUDY**

Determining the feasibility of an idea is, in many ways, among the most difficult of tasks, because so much information is needed and so little is available. There are also almost always time pressures such that analysis and estimates must be made quickly and, usually, with too few resources. These conditions are just reality. While, in theory, changes can later occur in scope, schedule and budget, in fact, most of these parameters are set during the feasibility study and are actually difficult to change later. That fact makes this phase especially important. The cost to perform a feasibility study is usually relatively small – perhaps 1–5% of the ultimate cost of the project – but it is money well spent, if spent wisely.

The feasibility study itself has two significant parts – first, determining and defining the scope, budget and schedule; and second, comparing the economic value of the defined project to some corporate standard or to other options for investment (Valle-Riestra, 1983). Some of the elements of scope include (Clark, 1999):

* + Process description
  + Process flow diagram
  + Raw ingredient utilization and procurement
  + Operations and production parameters
  + Packaging and filling
  + Degree of automation
  + Packaging material utilization
  + Materials handling methods
  + Finished product storage
  + Utility requirements

Site parameters (size, configuration, owner's style).

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

### 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.

### 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. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### 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.

## EXISTING SYSTEM

In the existing system they have gone through with the opencv but while extraction of the colors they got the wrong outputs.There is no exact color representation of colors with accuracy.

## 1.3 PROPOSED SYSTEM

In the proposed system, we are introducing the CV database and according to it the number of shades that can be identified using 865 color names along with their RGB and hex values. Whenever the cursor clicks the image, it automatically shows the RGB shades color values.

### ADVANTAGES

Better accuracy in segmentation under various illuminations

Less time consuming process

It is less sensitive to background noise

### ****APPLICATIONS****

People counting

Vehicle detection

Manufacturing industry applications

Tracking objects

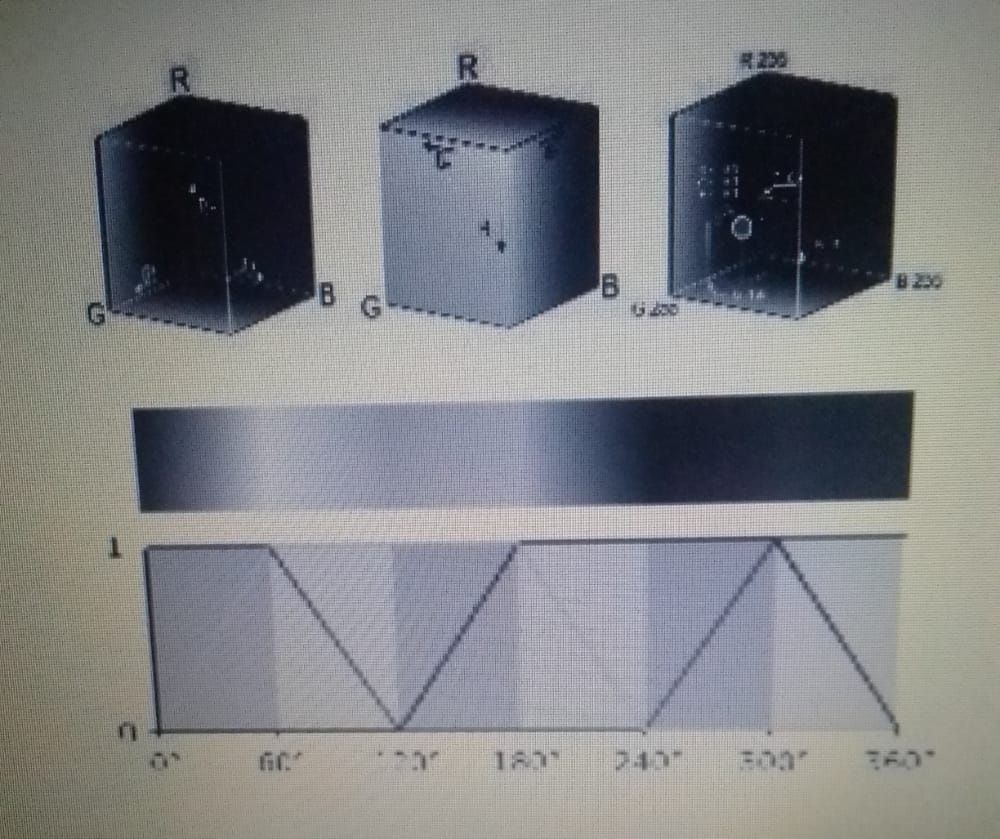
# SOFTWARE REQUIRMENTS SPECIFICATIONS

## INTRODUCTION:

Claims:1. Implementation of Color Detection system using Machine learning and  
Python consist of Fuzzy, Color Detection, 3-D Histogram, HSV, HSI etc.,  
2. Implementation of Color Detection system using Machine learning and  
Python of claim 1, where in said that Machine learning algorithms are now the  
most important concept for developers and programmers.  
3. Implementation of Color Detection system using Machine learning and  
Python of claim 1, wherein said Fuzzy logic is a good place to start learning  
about supervised machine learning.  
4. Implementation of Color Detection system using Machine learning and  
Python of claim 1, wherein said it is also recommended to utilize MATLAB,  
which has all the necessary tools and functions.  
5. Implementation of Color Detection system using Machine learning and  
Python of claim 1, wherein said that the image colors are compared to the  
lookup table, and the result is obtained if the lookup table values match the  
image colors , Description:  
In real world, human eye can detect and differentiate colors from predefined  
recognition or by experience from traditional learning, then the eyes can sense  
the color it sees and the mind interprets what that color is. In computer vision  
the color detection or color sensation mechanism is almost the same as in  
human eyes but it should be programmed using algorithms and some logic  
procedures that enables the CPU to take that piece of color and apply the  
proper mathematical and logical operations in order to take a decision of what  
color that piece includes according to its recognition about all color names.  
Color refers to a narrow band of wavelengths in terms of the subjective  
interpretation of light wavelength combinations by the human brain. There are  
no physical events associated with the term "color" to account for this. As a  
result, even the most fundamental concepts, such as "primary colors," are  
hazy. Light colors, also known as additive colors, are created by combining  
different wavelengths of red, green, and blue light. Color refers to the ability to  
notice and distinguish between various qualities of things. In any image  
processing technique, preprocessing is frequently performed on image intensity  
components rather than image color or tone components. It is impossible to  
predict the color of an object in advance because color varies depending on  
how bright the background is. As a result, it's critical to maintain a consistent  
color temperature regardless of background or lighting changes. A color  
space is useful for organizing and describing the color components of an  
image in a clear and concise manner. In practice, most machine learning  
techniques are implemented using fuzzy logic. It can detect and compare  
colors in an image file, and it can use a recognized dataset to decide between  
an image's colors and a specified color by comparing an image's colors to  
those found in a Look-up Table (LUT). The implementation also includes a  
dominant color detection and RGB to HSV, HSI color spaces conversions, and  
also 3D histogram representation of an image. The whole project is  
implemented using MATLAB programming environment by creating a (GUI)  
based application to load the image selected. The user can perform and apply  
all the previously mentioned functions to get instant results. The project’s  
results were successfully achieved as expected. Generally, in this paper we  
managed to perform three different functions that deals with images and  
colors. The major one is using the fuzzy logic to implement a color detection  
application, then implementing a dominant color detection application, then  
implementing basic conversions between RGB, HSV, HIS color models. The  
whole idea of this study is implemented and performed using MATLAB  
programming environment. The fuzzy logic color detection algorithm’s main  
function is implemented into two parts; firstly, implementing a fuzzy logic color  
learning and saving the colors’ information into a predefined dataset. Secondly,  
examining all the information of the predefined colors into the acquired image,  
compare all the known colors with image color, and extract the required color  
that matches the user demand if found in the dataset by using leaner  
interpolation method. For the dominant color detection function, we managed  
to extract the RGB value of the dominant color by measuring the Euclidian  
distance of each color to all colors. And for conversation between RGB, HSV,  
and HSI image color models we used a built-in color space function added with  
the software. Accordingly, this paper is organized into five sections. Section one  
provides the introduction and highlights the main problematic issues  
concerning image detection applications based on color detection mechanism;  
section two introduces a review of the related theoretical topics involved with  
proposed contribution study. Section three elaborates on the methodology used  
in this paper, while section four and five presents the conclusion and  
recommendations of this study, respectively.  
Conclusions:  
Fuzzy logic is one of the basic and common methods in artificial intelligence  
field and it has been implemented in too many applications. As such, in this  
project the fuzzy logic is used to implement a color detection algorithm using a  
predefined dataset that contain a lookup table with corresponding color names.  
The colors of the image are compared with the lookup table, and the result of  
the comparison is achieved if the values of the image colors are identified in the  
lookup table which matches the desired color to be detected by specifying the  
color name. In addition to three other function; most dominant color detection,  
HSV and HSI color models conversion, and 3D histogram illustration. The  
results show that the implementation was efficient and it satisfies the main  
objectives of the project.

This software requirements document specification provides complete information about the system called Be Friend which will be developed by our project team Visiondary. The system is planned as a camera and sensor integrated hardware device for blind people. In this section, we are going to give the definition of the problem, introduction of the purpose and scope of this document, definitions, acronyms and abbreviations, references and overview. In the following sections, we are going to introduce an overall description and feature of the project, present the specific requirements, use cases, data models and behavioral models and their detailed description. Finally, as our development program, we are going to state planning, team structure, team schedule and conclusion of the project respectively

### Design:

Self-diagnosing Identifying Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software are determined, software developers will design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view. RGB Color Space An image is basically a matrix of pixel values. An image can be represented using many color models like grey-scale, RGB, HSV etc. RGB model is used to detect the colors in an image [6]. The RGB color space comprises within an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. RGB commonly is being used in color detection, since image data captured by camera are normally provided and stored as RGB. For example, Yang et al. [7] constructed two codebooks using RGB features and local binary pattern features for visual tracking. 

### Implementation:

. In computers’ user interface applications some complex artificial intelligence  
should be used as machine learning algorithm to implement the sense of  
detection in the software especially in digital image processing applications. In  
this paper it’s proposed that using of fuzzy logic approaches that adopts some  
datasets of colors information and with the image processing techniques, image  
application can be more intelligent and reliable for advanced usage and various  
applications. As such, MATLAB-based software was that uses machine learning  
fuzzy logic was implemented to detect and predefine sense colors into an image.  
Furthermore, to extract the most dominant color in the image and implement  
some color model conversions for that image. The results demonstrated that  
fuzzy logic color detection mechanism is efficient and reliable. The contribution  
of this study can be of great value in improving computer vision and machine  
learning applications that based on color detection, also other basic practical  
image processing techniques are introduced which can be valued for interested  
researchers

## Testing:

After a whole lot of contemplation, the aforementioned leads the authors to believe that color detection is not restricted to just the human eyes and further has several partitions and techniques in machine interpretation of colors. Now to teach an algorithm is not hard, the implementation follows having a data file that contains the color name and its RGB values, the distance from each color is calculated to find the shortest one, it is readjusted in each iteration. So, the process goes as follows, collection of the dataset and consequently cleaning and importing it. The real magic comes here in building the shortest distance algorithm. In the given pseudo code, functions and keys are in bold and italic, mathematical calculations are highlighted in italic within quotes and variable are in bold. 1. create a function to calculate minimum distance from all colors and get the most matching color 2. set a minimum value to correspond with distance 3. while values in the file exist: a. set distance as the “sum of absolute values of difference between image and file values” b. reset minimum value c. get the name of color corresponding Because of their size, the resources dependencies needed to run the various examples and unit tests are not provided within the Pip package. They are separately available as [git Sub modules](https://git-scm.com/book/en/v2/Git-Tools-Submodules) when cloning the [repository](https://github.com/colour-science/colour-checker-detection). Once the dependencies are satisfied  **Color - Checker Detection** can be installed from the [Python Package Index](http://pypi.python.org/pypi/colour-checker-detection) by issuing this command in a shell:

to the new minimum 4. map “accuracy” based on value in the file and value in the image 5. plot distance loops 6. return the name of final closest valued color In this color detection algorithm, an application is built through which you can automatically get the name of the color by clicking on them. The benefit of the said model is that the distances are calculated by successive update of the minimum distance.

Installation:

pip install --user color-checker-detection

The overall development dependencies are installed as follows:

pip install --user 'colosr-checker-detection[development]'

### Maintenance:

Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes.

A common perception of maintenance is that it is merely fixing bugs. However, studies and surveys over the years have indicated that the majority, over 80%, of the maintenance effort is used for non-corrective actions. This perception is perpetuated by users submitting problem reports that in reality are functionality enhancements to the system.

As they evolve, they grow more complex unless some action such as code refactoring is taken to reduce the complexity.

The key software maintenance issues are both managerial and technical. Key management issues are: alignment with Patient priorities, staffing, which organization does maintenance, estimating costs. Key technical issues are: limited understanding, impact analysis, testing, and maintainability measurement.

## Purpose of this Document :

. Colour detection is necessary **to recognize objects**, it is also used as a tool in various image editing and drawing apps

## Overview:

The overview of our project is when user provides the data then the algorithm comes in action and verifies by predicting whether the account is Fake or Authentic by rectifying the parameters.

## Scope:

Colors information plays an important role in image and real time color sensor detection. Which affects the results of video segmentation and correct real time temperature value. According to the color information in RGB color space, the dominant color is determined at first. In the color image segmentation, the primary step is to settle on color space. The color model [9] we all know contains RGB, HSI, HSV, CMYK, CIE, YUV, and so on. The RGB model is that the most ordinarily used for hardware color model (Fig 5) while the HSI model is that the most ordinarily used color model for color processing. They’re often utilized in image processing technology.

Project Name : color Detection using python opencv

Team Member : T.JANAKAMMA

Purpose : To detects the colors in python

## EXTERNAL INTERFACE REQUIREMENTS :

### USER INTERFACES

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

FRONTEND : HTML,CSS,JS

BACKEND : Python

### SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and pre- requisites that need to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is works

* Operating System : Windows 7+
* Server side Script : Python, HTML, CSS & JS
* IDE : PyCharm
* Libraries Used : Pandas, NumPy,opencv,

### HARDWARE REQUIREMENTS

When deploying your gateway on the subject of-premises, you must make certain that the underlying hardware on the order of which you are deploying the gateway VM is practiced to dedicate the taking into account minimum resources:

* Processor . : I3/Intel Processor
* RAM : 4GB (min)
* Hard Disk : 128 GB
* Key Board : Standard Windows Keyboard

## SYSTEM FEATURES

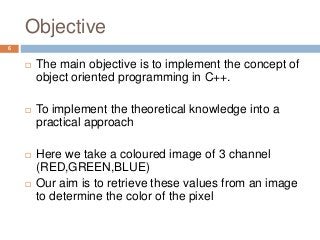
### INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input
* Methods for preparing input validations and steps to follow when error occur.

### OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. direction to the management for getting correct information from the computerized system in this Project.

1. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors
2. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that I 

### OUTPUT DESIGN

The Quality of a Insta User Details in Self Diagnosing A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In

output design it is determined how the information is to be displaced for immediate need and also the hard copy output.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action

## Functional Features:

### System:

* Download and unzip the zip file. ... **Color\_detection.py** – main source code of our project.
* **Colorpic.jpg** – sample image for experimenting.
* **Colors.csv** – a file that contains our dataset.

## Taking an image from the user:

### 

We are using argparse library to create an argument parser. We can directly give an image path from the command prompt:

import argparse

ap = argparse.ArgumentParser()

ap.add\_argument('-i', '--image', required=**True**, help="Image Path")

args = vars(ap.parse\_args())

img\_path = args['image']

#Reading image with opencv

img = cv2.imread(img\_path)

## Next, we read the CSV file with pandas:

The pandas library is very useful when we need to perform various operations on data files like CSV. **pd.read\_csv()** reads the CSV file and loads it into the pandas DataFrame. We have assigned each column with a name for easy accessing.

#Reading csv file with pandas and giving names to each column

index=["color","color\_name","hex","R","G","B"]

csv = pd.read\_csv('colors.csv', names=index, header=None

## Set a mouse callback event on a window:

First, we created a window in which the input image will display. Then, we set a callback function which will be called when a mouse event happens.

cv2.namedWindow('image')

cv2.setMouseCallback('image',draw\_function)

## Create the draw\_function:

It will calculate the rgb values of the pixel which we double click. The function parameters have the event name, (x,y) coordinates of the mouse position, etc. In the function, we check if the event is double-clicked then we calculate and set the r,g,b values along with x,y positions of the mouse.

**def** draw\_function(event, x,y,flags,param):

**if** event == cv2.EVENT\_LBUTTONDBLCLK:

global b,g,r,xpos,ypos, clicked

clicked = **True**

xpos = x

ypos = y

b,g,r = img[y,x]

b = int(b)

g = int(g)

r = int(r)

## Calculate distance to get color name:

We have the r,g and b values. Now, we need another function which will return us the color name from RGB values. To get the color name, we calculate a distance(d) which tells us how close we are to color and choose the one having minimum distance.

Our distance is calculated by this formula:

*d = abs(Red – ithRedColor) + (Green – ithGreenColor) + (Blue – ithBlueColor)*

**def** getColorName(R,G,B):

minimum = 10000

**for** i **in** range(len(csv)):

d = abs(R- int(csv.loc[i,"R"])) + abs(G- int(csv.loc[i,"G"]))+ abs(B- int(csv.loc[i,"B"]))

**if**(d<=minimum):

minimum = d

cname = csv.loc[i,"color\_name"]

**return** cname

## Display image on the window:

Whenever a double click event occurs, it will update the color name and RGB values on the window.

Using the **cv2.imshow()** function, we draw the image on the window. When the user double clicks the window, we draw a rectangle and get the color name to draw text on the window using **cv2.rectangle** and **cv2.putText()** functions.

while(1):

cv2.imshow("image",img)

if (clicked):

#cv2.rectangle(image, startpoint, endpoint, color, thickness) -1 thickness fills rectangle entirely

cv2.rectangle(img,(20,20), (750,60), (b,g,r), -1)

#Creating text string to display ( Color name and RGB values )

text = getColorName(r,g,b) + ' R='+ str(r) + ' G='+ str(g) + ' B='+ str(b)

#cv2.putText(img,text,start,font(0-7), fontScale, color, thickness, lineType, (optional bottomLeft bool) )

cv2.putText(img, text,(50,50),2,0.8,(255,255,255),2,cv2.LINE\_AA)

#For very light colours we will display text in black colour

if(r+g+b>=600):

cv2.putText(img, text,(50,50),2,0.8,(0,0,0),2,cv2.LINE\_AA)

clicked=False

#Break the loop when user hits 'esc' key

if cv2.waitKey(20) & 0xFF ==27:

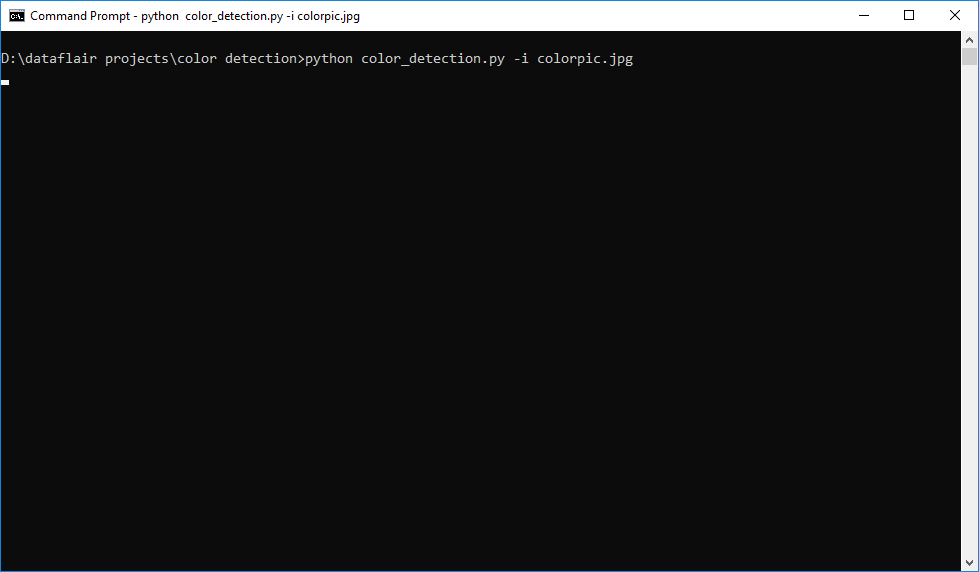
break

cv2.destroyAllWindows()

### Run Python File

The beginner Python project is now complete, you can run the Python file from the command prompt. Make sure to give an image path using ‘-i’ argument. If the image is in another directory, then you need to give full path of the image:

python color\_detection.py -i <add your image path here>

* 

### User :

* + Load Dataset : Purpose – To load the dataset he/she want to work on.

Actor - User

Input - The user choose the image and submit it.

Output - The submitted image will be displayed on screen.

* + Select model : Purpose – User can apply the model to the dataset for accuracy.

Actor - User

Input - The user select a model to see accuracy.

Output – accuracy of the selected model will be displayed.

* + Prediction : Purpose – Passing Parameters to predict the output.

Actor - User

Input - The user will enter all the required fields. The fields include

Output – displays whether the account is fake or not

* + - Graphs : Purpose - User can evaluate the model performance using the

graphs.

Actor – System

Input – The user will click the submit button

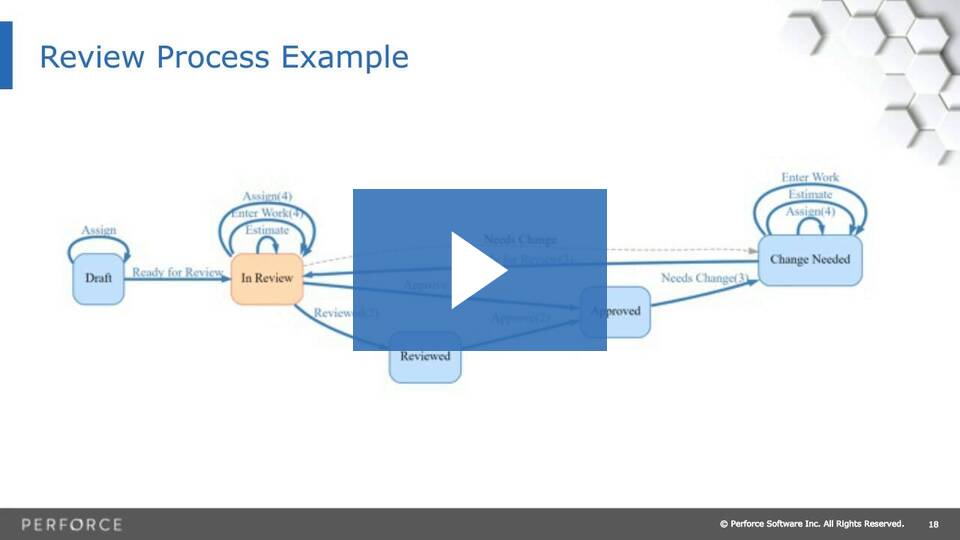
Output – The performance Comparison of Algorithms will be displayed.

## OTHER NON-FUNCTIONAL REQUIREMENTS:

While a system can still work if NFRs are not met, it may not meet user or stakeholder expectations, or the needs of the business.

NFRs also keep functional requirements in line, so to speak. Attributes that make the product affordable, easy to use, and accessible, for example, come from NFRs.

Let’s get more specific.

[](https://www.perforce.com/webinars/alm/essential-tips-modern-requirements-management?wvideo=04rel6s8df)

NFRs are often thought of as the “itys.” While the specifics will vary between products, having a list of these NFR types defined up front provides a handy checklist to make sure you’re not missing critical requirements.

This is not an exhaustive list, but here’s what we mean:

### NFR “Itys”

**Security** — Does your product store or transmit sensitive information? Does your IT department require adherence to specific standards? What security best practices are used in your industry?

**Capacity** — What are your system’s storage requirements, today and in the future? How will your system scale up for increasing volume demands?

**Compatibility** — What are the minimum hardware requirements? What operating systems and their versions must be supported?

**Reliability and Availability** — What is the critical failure time under normal usage? Does a user need access to this all hours of every day?

**Maintainability  + Manageability**—How much time does it take to fix components, and how easily can an administrator manage the system? Under this umbrella, you could also define **Recoverability**and **Serviceability.**

**Scalability** – The Black Friday test. What are the highest workloads under which the system will still perform as expected?

**Usability**— How easy is it to use the product? What defines the experience of using the product?

“Ity’s” don’t cover all types, however.

# 3.ANALYSIS

## UML Diagrams

Unified Modelling Language is the language used to visualize, specify, construct and document any component of software engineering.

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

### User Model View

1. This view represents the system from the user’s perspective.
2. The analysis representation describes a usage scenario from the end user’s perspective.

### Structural model view

1. In this model the data and functionality are arrived from inside the system.
2. This model view models the static structures.

### Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

### Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

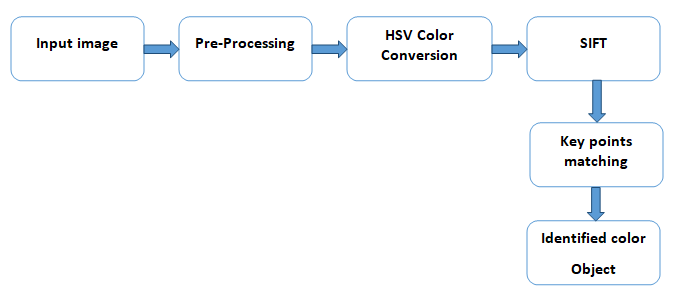
### Environmental Model View

In this structural and behavioral aspects of the environment in which the system is to be implemented are represented. UML is specifically constructed through two different domains they are:

* 1. UML analysis modelling, which focuses on the user model and structural model views of the system.
  2. UML design modelling, which focuses on the behavioral modelling,implementation modelling and environmental model views.

## USE CASE DIAGRAM:

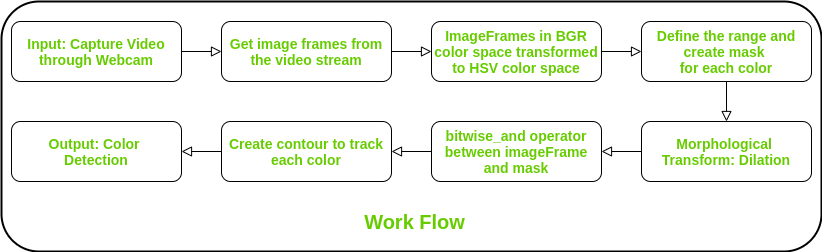
In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

* Scenarios in which your system or application interacts with people, organizations, or external systems
* Goals that your system or application helps those entities (known as actors) achieve
* The scope of your system
  + System – It deals with Taking File, Preprocessing ,Generating Accuracy, Analyze Data, Predict Results.
  + User – User deals with following usecases
    - Upload image
    - Select Model
    - Enter Data

### SEQUENCE DIAGRAM:

A sequence diagram is a type of **interaction diagram** because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

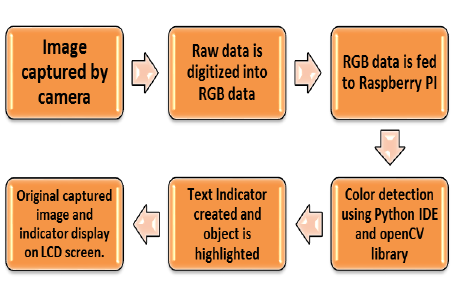
* + - Sequence diagram consists of objects like User and System.
    - These objects shows the messages passed between them in sequential order. Following are the messages passed
      * Upload image
      * Select Model

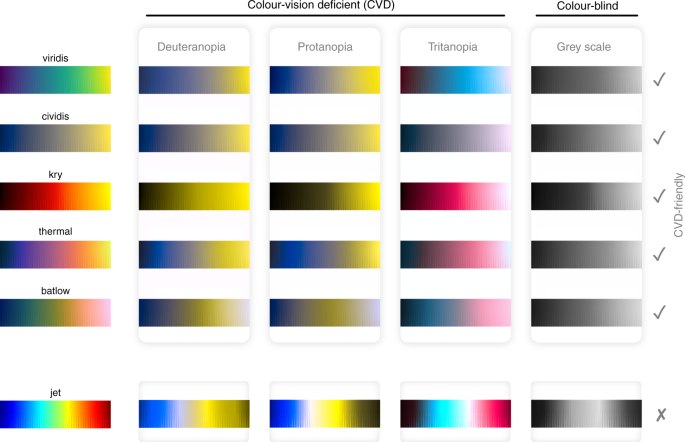


* + - * Enter Data
      * View Results
      * Preprocessing
      * Generate Accuracy
      * Analyse Data
      * Predict Results

### COLLABORATION DIAGRAM:

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.





# 3.4 state chart diagram

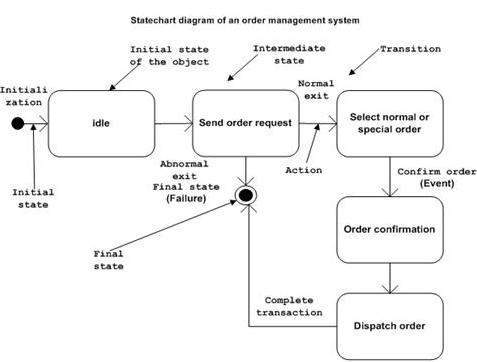
Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.

Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

To model the dynamic aspect of a system.

To model the life time of a reactive system.

To describe different states of an object during its life time. 

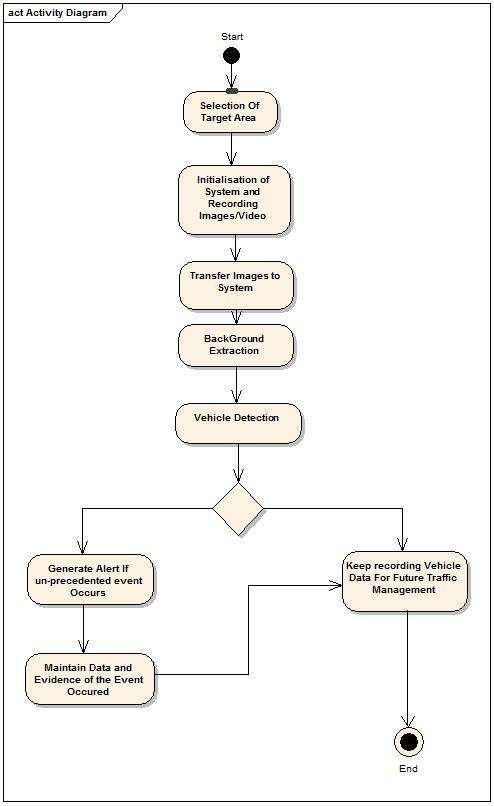
Define a state machine to model the states of an object.

* 1. **ACTIVITY DIAGRAM:**

#### An activity diagram is a behavioural diagram i.e., it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed

.They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system

* Activity Diagram portraits the control flow from a start point to end point.
* Here, start point will be initiated by System and User and the whole process will
* be continued in top to bottom flow.

z

* Activity diagram mainly used to depict the dynamic aspects of a system

# DESIGN

## ARCHITECTURE DESIGN:

### 3 Tier Architecture:

Three-tier is a client–server architecture in which the user interface, functional process logic, computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. It was developed by John J. Donovan in Open Environment Corporation.

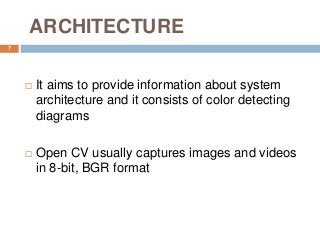
The three-tier model is software architecture and a software design pattern.Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the user interface code.

### Presentation tier:

This is the Index level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network .

### Application tier (business logic, logic tier, data access tier, or middle tier)

The logic tier is pulled out from the presentation tier and, as its own layer; it controls an application’s functionality by performing detailed processing.



### Data tier:

This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.



##### Fig.4.1.1 Architecture

Here, User gives the dataset to the system and that will further splits the dataset and model training will be done to generate graphs.

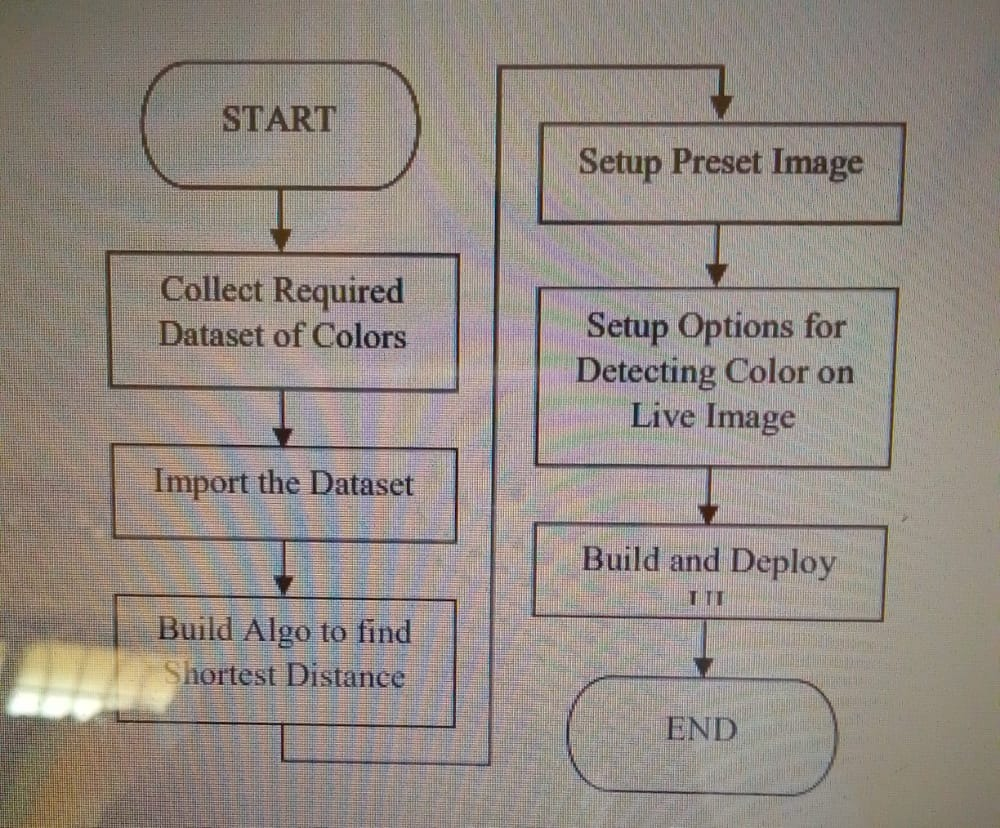
The System takes the data by the user (i.e., Parameters) and selects the highest accuracy model to give the result.

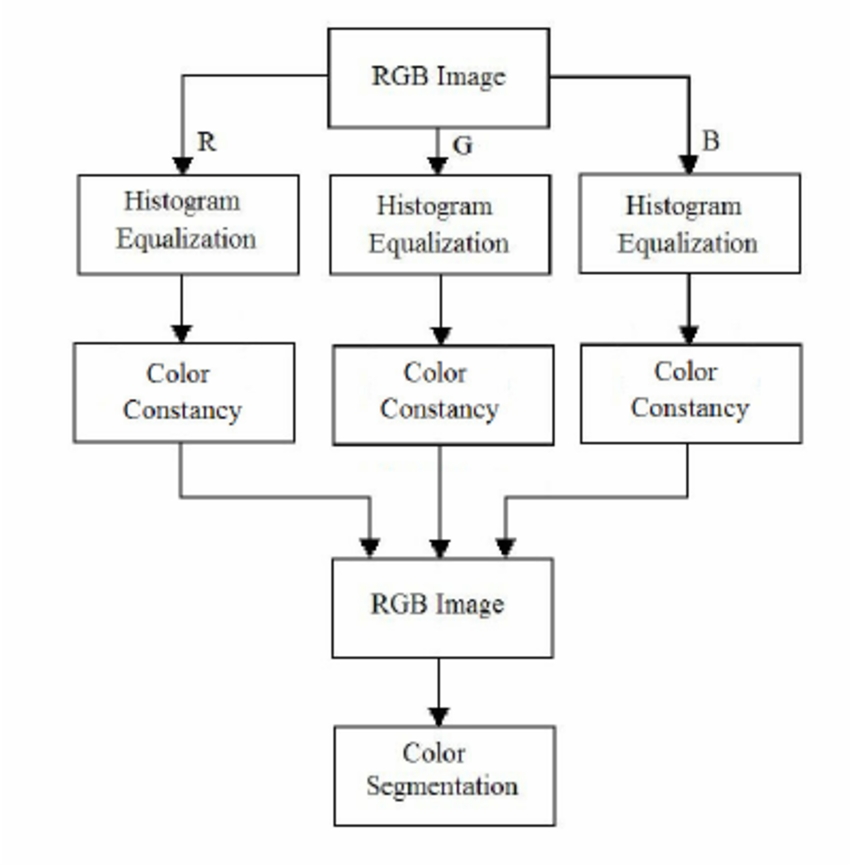
Finally the system predicts and gives the result (i.e., whether the account is fake or not ) and displays the result to the user.

## PROJECT FLOW:

Project Flow is an indicator of change planned for occurrence within a project over time. It shows the movement of project product from its conceptualization and design through delivery and deployment. Project flow describes a preset sequence of activities required to plan, produce, deliver and maintain project product, along with information, materials, and resources required by the project.

Project flow is a convenient way to define and plan projects. It helps link project budget and schedule to project activities and tasks. Traditionally it is designed in the form of a chart or diagram which is a great tool to visually represent how a project is supposed to produce and deploy its product. Simple tree-like lists or hierarchies of project activities are also used to map out and depict project flow.



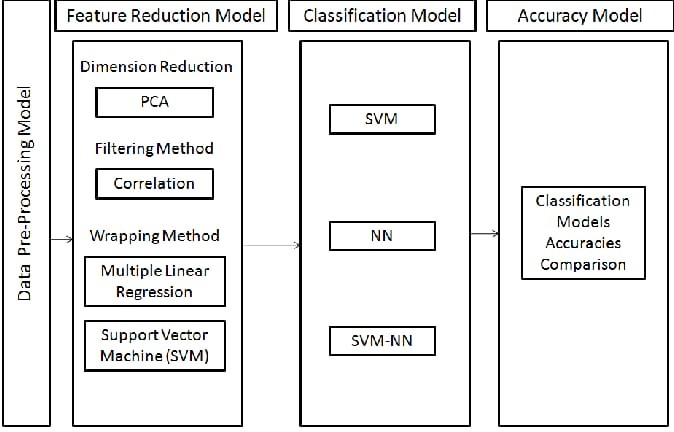


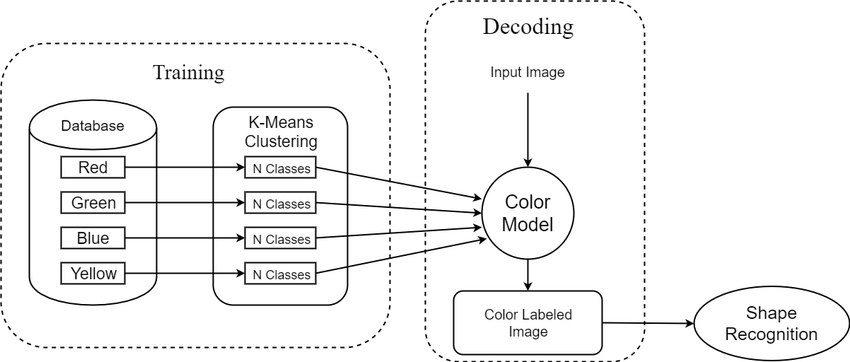
* 1. **DATABASE DESIGN/MODEL:**

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database design in DBMS are to produce logical and physical designs models of the proposed database system.

The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.



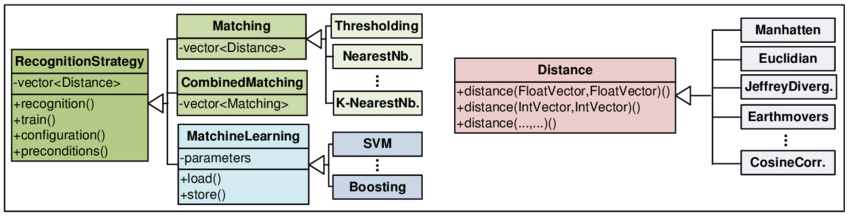


##### Fig.4.3.1 Database Model

## CLASS DIAGRAM:

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

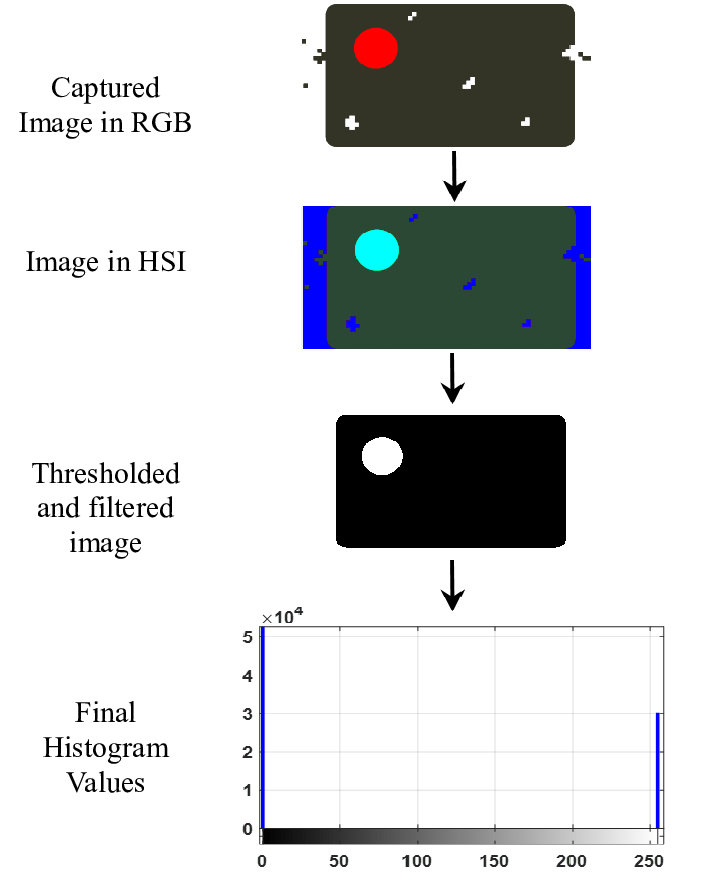


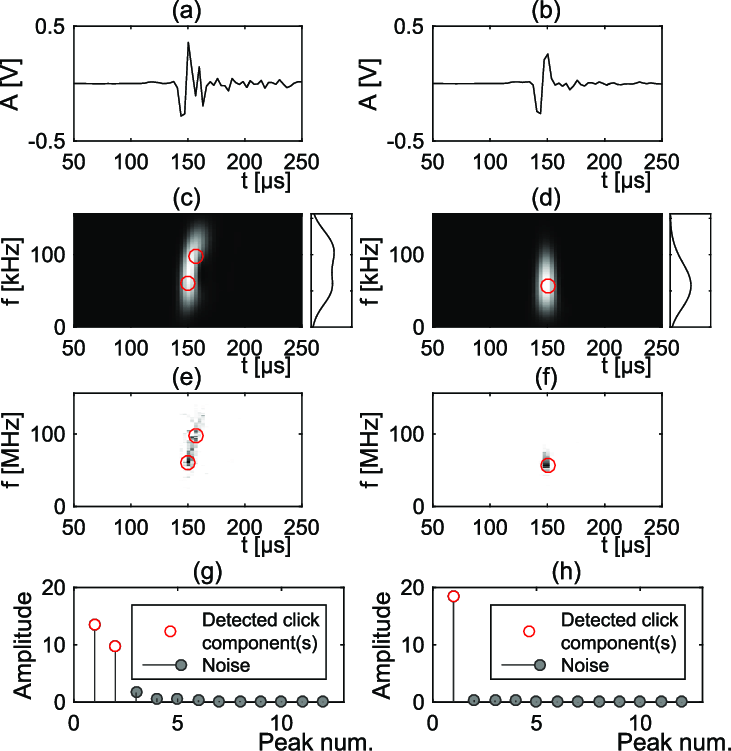
##### Fig.4.5.1 Class Diagram

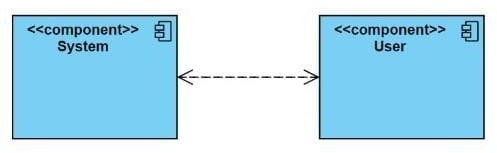
* + - Here System and User are the classes. And in each one has a collection of Objects. Each object has Some of Attributes, Methods and set of Behaviours.

## COMPONENT DIAGRAM:

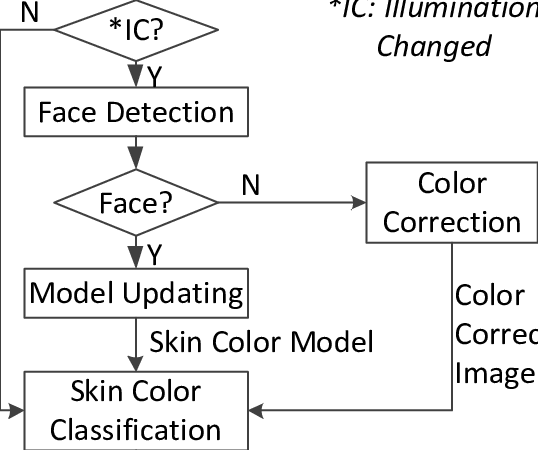
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

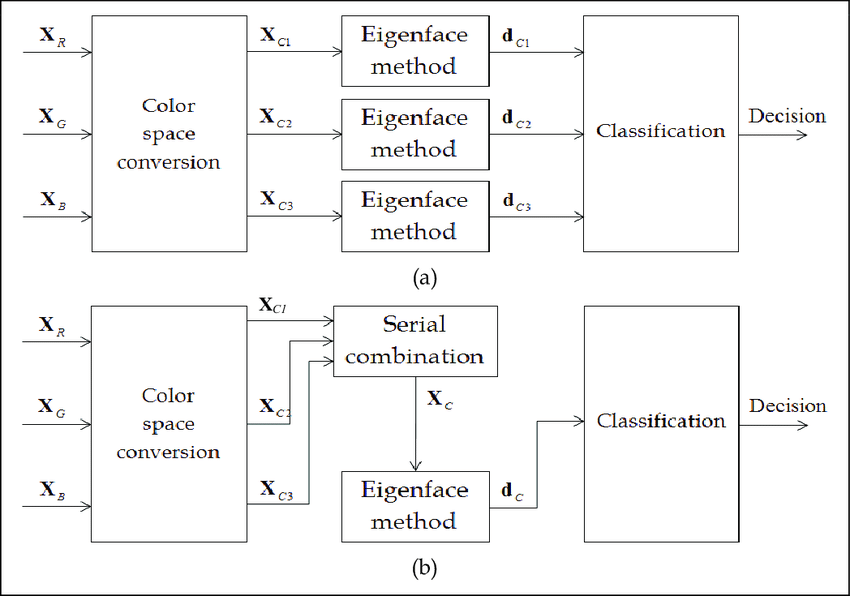






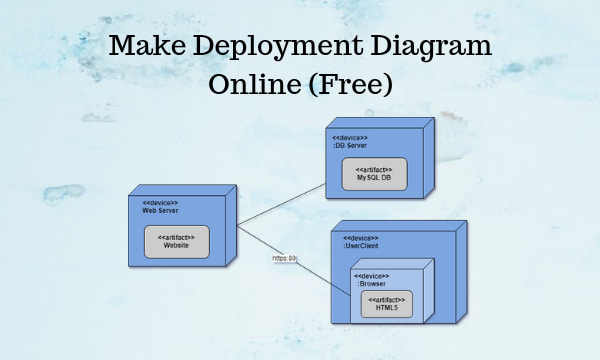
##### https://in.mathworks.com/help/examples/android/win64/ColorDetectionExample_01.png

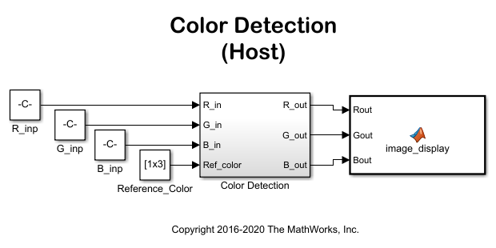




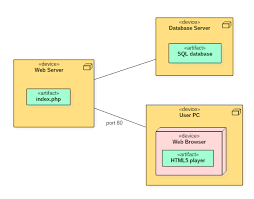
* 1. **Deployment Diagram:**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hard wares used to deploy the application.





##### Fig.4.6.1 Deployment Diagram



# IMPLEMENTATION

## INTRODUCTION TO TECHNOLOGY:

* + 1. **Python**
* Python is a High level, structured, open-source programming language that can be used for a wide variety of programming tasks.
* Python within itself is an interpreted programming language that is automatically compiled into byte code before execution.
* It is also a dynamically typed language that includes (but does not require one to use) object-oriented features.
* NASA has used Python for its software and has adopted is as the standard scripting language for its integrated Planning System.
* Python is also extensively used by Google to implement many components of its Web Crawler and Search Engine & Yahoo! For managing its discussion groups.

**History of Python**

* Python was created by Guido Van Rossum.
* The design began in the late 1980s and was first released in February 1991.

### Why the name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late 70s. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

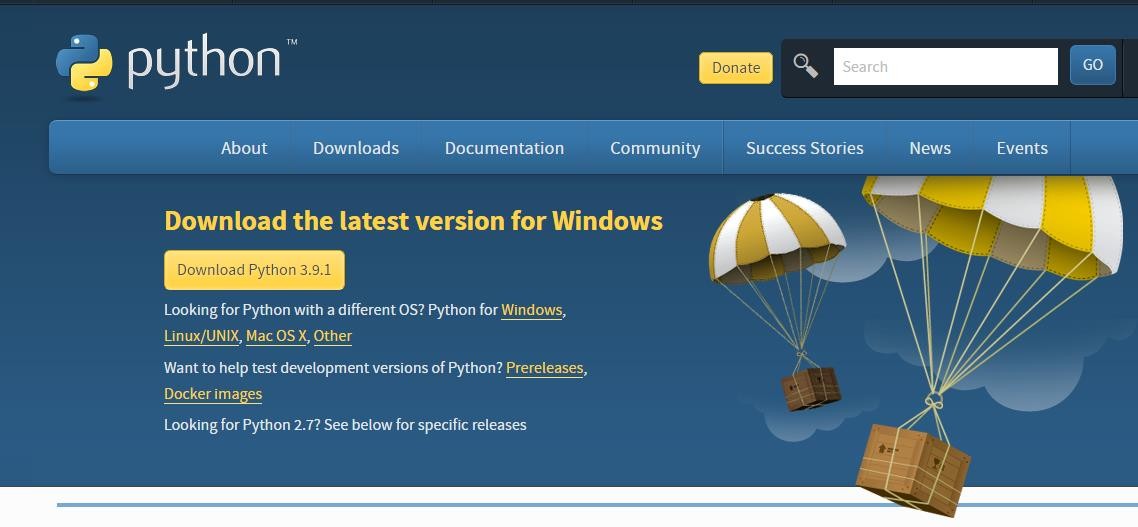
**Python Version History** Implementation started - December 1989 Internal releases - 1990



##### Fig.5.1.1.1 History of Python

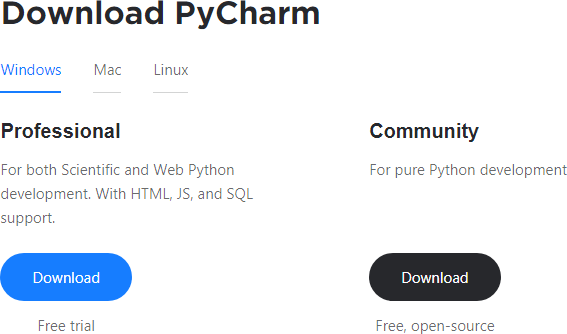
**SOFTWARE INSTALLATION FOR MACHINE LEARNING PROJECTS:**

Installing Python:

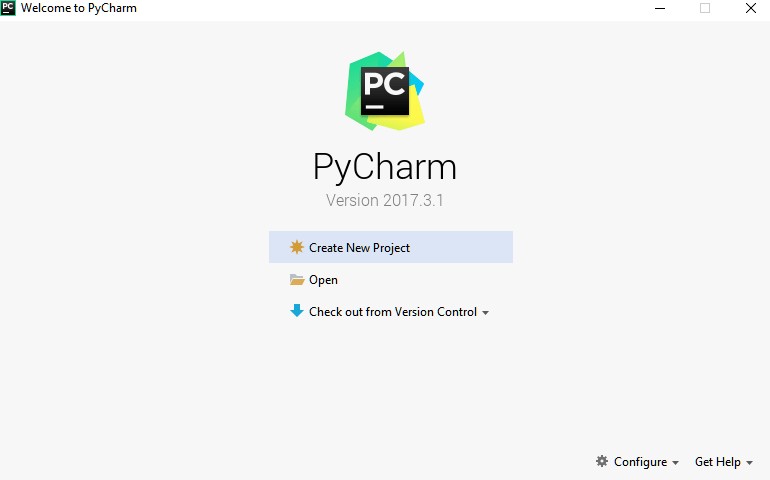
1. To download and install Python visit the official website of Python <https://www.python.org/downloads/>and choose your version.
2. Once the download is complete, run the exe for install Python. Now click on Install Now.
3. You can see Python installing at this point.
4. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

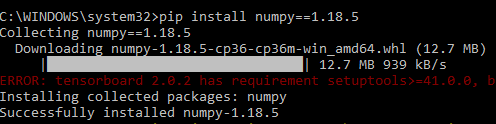
Installing PyCharm:

* 1. To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/>and Click the "DOWNLOAD" link under the Community Section.



* 1. Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.
  2. On the next screen, Change the installation path if required. Click “Next”.
  3. On the next screen, you can create a desktop shortcut if you want and click on “Next”.
  4. Choose the start menu folder. Keep selected Jet Brains and click on “Install”.
  5. Wait for the installation to finish.
  6. Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.
  7. After you click on "Finish," the Following screen will appear.



* 1. You need to install some packages to execute your project in a proper way.
  2. Open the command prompt/ anaconda prompt or terminal as administrator.
  3. The prompt will get open, with specified path, type “pip install package name” which you want to install (like Numpy, pandas, seaborn, scikit-learn, matplotlib.pyplot)

Ex: pip install numpy

### DJANGO,

Django is a Web framework written in Python.A Web framework is a software that supports the development of dynamic Web site,applications and services. It provides a set of tools and functionalities that solves many common problems associated with Web development, such as security features, database access, sessions, template processing, URL

routing, internationalization, localization, and much more. Using a Web framework, such as Django, enables us to develop secure and reliable Web applications very quickly in a standardized way. The development of Django is supported by the Django Software Foundation, and it’s sponsored by companies like JetBrains and Instagram.

Who’s Using Django? It’s good to know who is using Django out there, so to have an idea what you can do with it. Among the biggest Web sites using Django we have: Instagram, Disqus, Mozilla, Bitbucket, Last.fm, National Geographic. Installation The first thing we need to do is install some programs on our machine so to be able to start playing with Django. The basic setup consists of installing

* + - * Python
      * Virtualenv
      * Django

Using virtual environments is not mandatory, but it’s highly recommended.

##### Installing Virtualenv

we are going to use **pip**, a tool to manage and install Python packages, to install

##### virtualenv.

In the Command Prompt, execute the command below:

##### pip install virtualenv

From now on, everything we install, including Django itself, will be installed inside a Virtual Environment.

##### mkdir myproject cd myproject

This folder is the higher level directory that will store all the files and things related to our Django project, including its virtual environment.

let’s start by creating our very first virtual environment and installing Django. Inside the **myproj** folder:

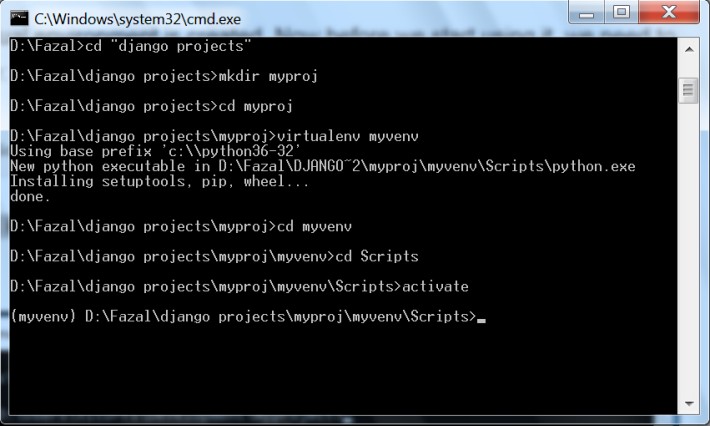
##### virtualenv myvenv

Our virtual environment is created.

Now before we start using it, we need to activate:

##### myvenv\Scripts\activate

You will know it worked if you see (venv) in front of the command line, like this:



to deactivate the venv run the command below:

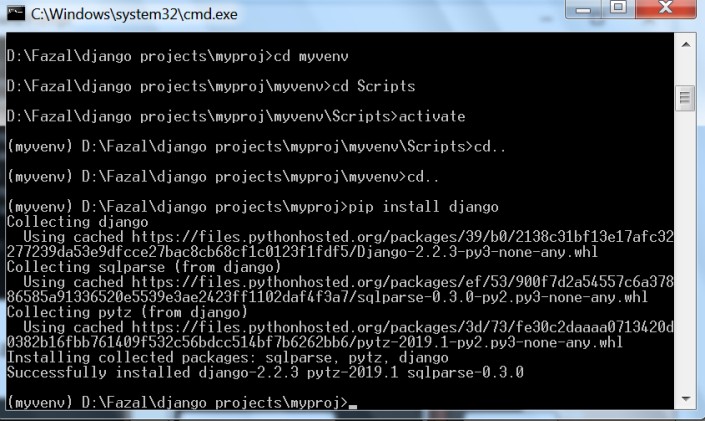
##### venv\Scripts\deactivate.bat

But let’s keep it activated for the next steps.

##### Installing Django

Now that we have the venv activated, run the following command to install Django:

##### pip install django



Starting a New Project

To start a new Django project, run the command below:

##### Django-admin startproject myproject

The command-line utility Django-admin is automatically installed with Django.

After we run the command above, it will generate the base folder structure for a Django project. Our initial project structure is composed of five files:

* manage.py: a shortcut to use the Django-admin command-line utility. It’s used to run management commands related to our project. We will use it to run the development server, run tests, create migrations and much more.
* init .py: this empty file tells Python that this folder is a Python package.
* settings.py: this file contains all the project’s configuration.
* ourls.py: this file is responsible for mapping the routes and paths in our project

For example, if you want to show something in the URL **/about/,** you have to map it here first.

wsgi.py: this file is a simple gateway interface used for deployment. You don’t have to bother about it. Just let it be for now.

Django comes with a simple web server installed.

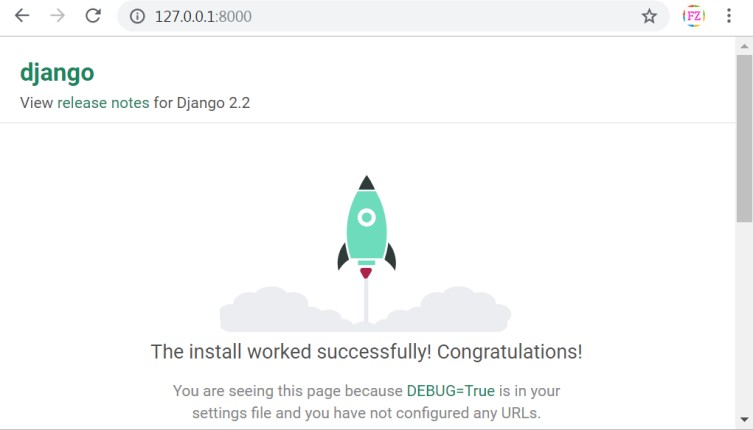
It’s very convenient during the development, so we don’t have to install anything else to run the project locally.

We can test it by executing the command:

##### python manage.py runserver

For now, you can ignore the migration errors; we will get to that later.

Now open the following URL in a Web browser: http://127.0.0.1:8000 and you should see the following page:



Hit CTRL + BREAK to stop the development server.

Django Apps

In the Django philosophy we have two important concepts:

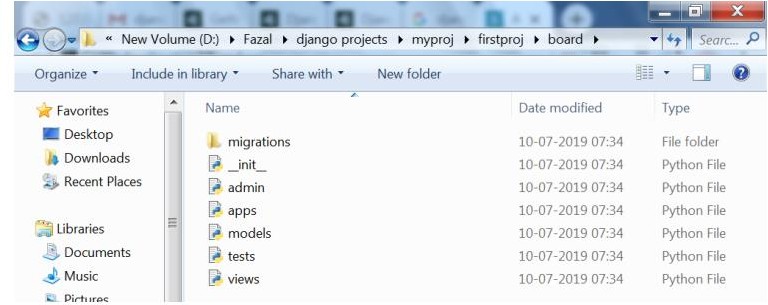
* app: is a Web application that does something. An app usually is composed of a set of models (database tables), views, templates, tests.
* project: is a collection of configurations and apps. One project can be composed of multiple apps, or a single app.

It’s important to note that you can’t run a Django app without a project. Simple websites like a blog can be written entirely inside a single app, which could be named **blog** or **weblog** for example.

let’s create a simple Web Forum or Discussion Board. To create our first app, go to the directory where the manage.py file is and executes the following command:

django-admin startapp boards

Notice that we used the command **startapp** this time.



So, let’s first explore what each file does:

* **migrations/:** here Django store some files to keep track of the changes you create in the models.py file, so to keep the database and the models.py synchronized.
* **admin.py**: this is a configuration file for a built-in Django app called Django Admin.
* **apps.py:** this is a configuration file of the app itself.
* **models.py:** here is where we define the entities of our Web application. The models are translated automatically by Django into database tables.
* **tests.py**: this file is used to write unit tests for the app.
* **views.py**: this is the file where we handle the request/response cycle of our Web application.

Now that we created our first app, let’s configure our project to use it.

To do that, open the **settings.py** and try to find the INSTALLED\_APPS variable: **settings.py**

INSTALLED\_APPS = [

'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles',

]

As you can see, Django already come with 6 built-in apps installed. They offer common functionalities that most Web applications need, like authentication, sessions, static files management (images, java scripts, css, etc.) and so on.

## Python Web Frameworks

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common web framework functionality

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

* URL routing
* HTML, XML, JSON, and other output format templating
* Database manipulation
* Security against Cross-site request forgery (CSRF) and other attacks
* Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

Comparing web frameworks

There is also a repository called compare-python-web-frameworks where the same web application is being coded with varying Python web frameworks, templating engines and object.

## Web framework resources

When learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.

Frameworks are a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.

What is a web framework? is an in-depth explanation of what web frameworks are and their relation to web servers.

Django vs Flash vs Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.

This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.

Python’s web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.

What web frameworks do you use and why are they awesome? is a language agnostic Reddit discussion on web frameworks. It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.

This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?". The votes aren't as important as the list of the many frameworks that are available to Python developers.

Web frameworks learning checklist

Choose a major Python web framework (Django or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.

## STUDY OF THE SYSTEM

Numpy Pandas Matplotlib Scikit –learn **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 multi-dimensional 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. The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

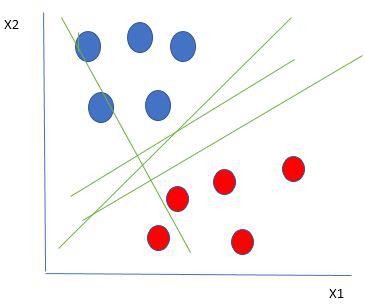
**NumPy:** Base n-dimensional array package **Matplotlib:** Comprehensive 2D/3D plotting **Pandas:** Data structures and analysis

### Algorithms

**Support Vector Machine:**

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyper plane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyper plane depends upon the number of features. If the number of input features is two, then the hyper plane is just a line. If the number of input features is three, then the hyper plane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

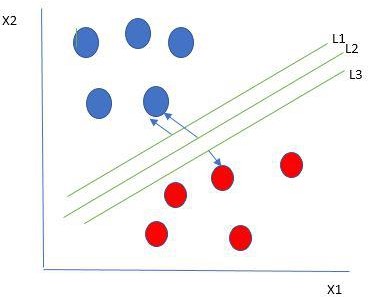
Let’s consider two independent variables x1, x2 and one dependent variable which is either a blue circle or a red circle. Types of SVM



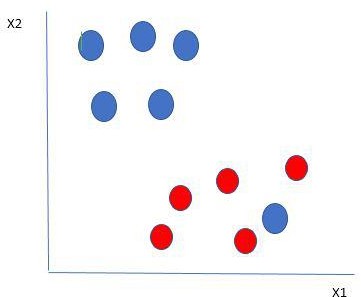
its very clear that there are multiple lines (our hyper plane here is a line because we are considering only two input features x1, x2) that segregates our data points or does a classification between red and blue circles. So how do we choose the best line or in general the best hyper plane that segregates our data points.

### Selecting the best hyper-plane:

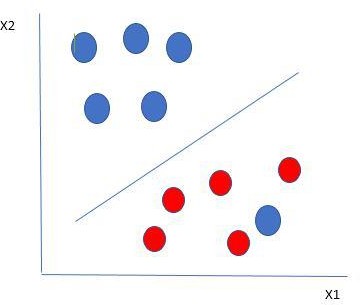
One reasonable choice as the best hyper plane is the one that represents the largest separation or margin between the two classes.



So we choose the hyper plane whose distance from it to the nearest data point on each side is maximized. If such a hyper plane exists it is known as the maximum-margin hyper plane/hard margin. So from the above figure, we choose

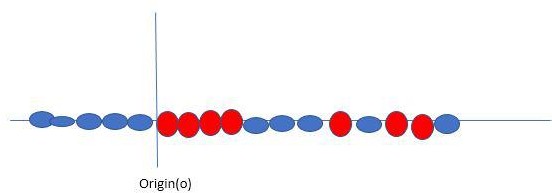


Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It’s simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers

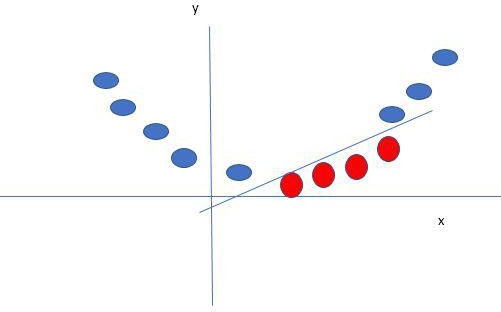


So in this type of data points what SVM does is, it finds maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these type of cases are called soft margin. When there is a soft margin to the data set, the SVM tries to minimize *(1/margin+∧(∑penalty))*. Hinge loss is a commonly used penalty. If no violations no hinge loss. If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Say, our data is like shown in the figure above. SVM solves this by creating a new variable using a kernel. We call a point xi on the line and we create a new variable yi as a function of distance from origin o.so if we plot this we get something like as shown below



In this case, the new variable y is created as a function of distance from the origin. A non- linear function that creates a new variable is referred to as kernel.

**SVM Kernel:**

The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put the kernel, it does some extremely complex data transformations then finds out the process to separate the data based on the labels or outputs defined.

##### Advantages of SVM:

* Effective in high dimensional cases
* Its memory efficient as it uses a subset of training points in the decision function called support vectors
* Different kernel functions can be specified for the decision functions and its possible to specify custom kernels

### Neural Networks

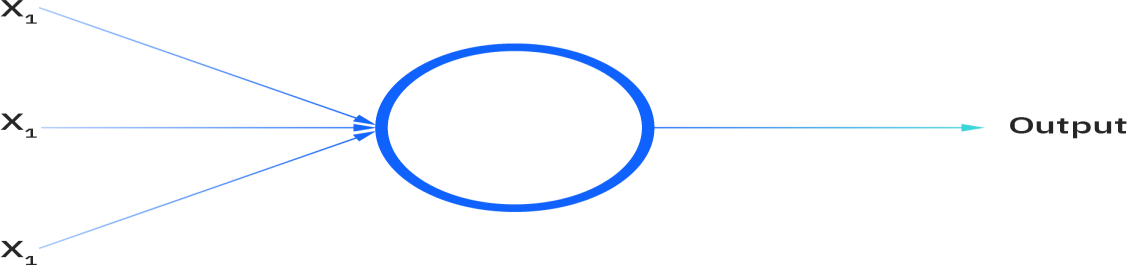
* + Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.
  + Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.
  + Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google’s search algorithm.

### Types of Neural Networks

Neural networks can be classified into different types, which are used for different purposes. While this isn’t a comprehensive list of types, the below would be representative of the most common types of neural networks that you’ll come across for its common use cases:

### Perceptron

The perceptron is the oldest neural network, created by Frank Rosenblatt in 1958. It has a single neuron and is the simplest form of a neural network:



**Fig.5.1.5.1 Perceptron**

Feed forward neural networks, or multi-layer perceptron’s (MLPs), are what we’ve primarily been focusing on within this article. They are comprised of an input layer, a hidden layer or layers, and an output layer. While these neural networks are also commonly referred to as MLPs, it’s important to note that they are actually comprised of sigmoid neurons, not perceptron’s, as most real-world problems are nonlinear. Data usually is fed into these models to train them, and they are the foundation for computer vision, natural language processing, and other neural networks.

### CNN

Convolutional neural networks (CNNs) are similar to feed forward networks, but they’re usually utilized for image recognition, pattern recognition, and/or computer vision. These networks harness principles from linear algebra, particularly matrix multiplication, to identify patterns within an image.

### RNN

Recurrent neural networks (RNNs) are identified by their feedback loops. These learning algorithms are primarily leveraged when using time-series data to make predictions about future outcomes, such as stock market predictions or sales forecasting.

## SAMPLE CODE views.py

from django.shortcuts import render,redirect import pandas as pd

from django.contrib import messages

from sklearn.model\_selection import train\_test\_split from sklearn import svm

import matplotlib.pyplot as plt import seaborn as sns

from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Dense

from sklearn.ensemble import StackingClassifier from sklearn.metrics import accuracy\_score

def index(request):

return render(request,'index.html') def home(request):

global df,X,y

if request.method == "POST": file = request.FILES['myfile'] df = pd.read\_csv(file)

df = df.rename(

columns={'edge\_followed\_by': 'Followed\_by', 'edge\_follow': 'Follow', 'username\_length': 'name\_length',

'username\_has\_number': 'has\_number', 'full\_name\_has\_number': 'full\_name\_number',

'full\_name\_length': 'fullname\_length', 'is\_private': 'private', 'is\_joined\_recently':

'recent',

'has\_channel': 'channel', 'is\_business\_account': 'business\_account', 'has\_guides':

'guides',

'has\_external\_url': 'external\_url', 'is\_fake': 'fake'}) df.drop(['guides'], axis=1, inplace=True) df.drop(['channel'], axis=1, inplace=True)

X = df.drop(['fake'], axis=1) y = df['fake']

return render(request, 'upload.html', {'df': df}) return render(request,'home.html')

def model(request):

global x\_train, y\_train, x\_test, y\_test,a,auc,avc if request.method == 'POST':

name = request.POST['cars']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=100) if name == "SVM":

from sklearn.metrics import accuracy\_score clf = svm.SVC(kernel='linear') clf.fit(x\_train, y\_train)

y\_pred = clf.predict(x\_test)

a = accuracy\_score(y\_pred, y\_test) messages.success(request,"Support Vector Machine Accuracy :") return render(request, 'algorithm.html',{'a':a})

elif name=='NN':

from sklearn.metrics import accuracy\_score model = Sequential()

model.add(Dense(20, input\_dim=x\_train.shape[1], activation='relu')) model.add(Dense(40))

model.add(Dense(1, activation='softmax')) model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy']) model.fit(x=x\_train, y=y\_train, verbose=1, epochs=10)

pre = model.predict(x\_test)

auc= accuracy\_score(pre, y\_test) messages.success(request,"Neural Networks Accuracy") return render(request,'algorithm.html',{'a':auc})

elif name == 'SVM-NN':

from sklearn.metrics import accuracy\_score import tensorflow as tf

model = Sequential()

model.add(Dense(20, input\_dim=x\_train.shape[1], activation='relu'))

model.add(Dense(40)) model.add(Dense(1,kernel\_regularizer=tf.keras.regularizers.l2(0.01),

activation='softmax'))

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy']) model.fit(x=x\_train, y=y\_train, verbose=1, epochs=10)

pre = model.predict(x\_test)

avc = accuracy\_score(pre, y\_test) messages.success(request, "SVM-ANN accuracy") return render(request, 'algorithm.html', {'a': avc})

else:

messages.success(request,"You didn't selected any model") return render(request,'algorithm.html')

return render(request,'algorithm.html')

def prediction(request):

if request.method=='POST':

Followed\_by = request.POST['Followed\_by']

Follow=request.POST['Follow'] name\_length=request.POST['name\_length'] has\_number=request.POST['has\_number'] full\_name\_number=request.POST['full\_name\_number'] fullname\_length=request.POST['fullname\_length'] private=request.POST['private'] recent=request.POST['recent'] business\_account=request.POST['business\_account'] external\_url=request.POST['external\_url']

c=list((Followed\_by,Follow,name\_length,has\_number,full\_name\_number,fullname\_length,pr ivate,recent,business\_account,external\_url))

clf = svm.SVC(kernel='linear') clf.fit(x\_train, y\_train)

y\_pred = clf.predict(x\_test) n=clf.predict([c])

if n==[1]:

messages.success(request,"It is a fake account") return render(request, 'prediction.html')

else:

messages.success(request,"it is not a fake account")

return render(request, 'prediction.html') return render(request,'prediction.html')

def graphs(request):

x = ['Support Vector Machine','Neural Networks','SVM\_NN'] y = [a,auc,avc]

graph = sns.barplot(x, y) plt.title('Performance Comparision') graph.set(ylabel="Accuracy") plt.show()

return redirect('/')

### manage.py

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks.""" import os

import sys def main():

"""Run administrative tasks.""" os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'account.settings') try:

from django.core.management import execute\_from\_command\_line except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you " "forget to activate a virtual environment?"

) from exc execute\_from\_command\_line(sys.argv)

if name == ' main ': main()

### Custom.js

jQuery(document).ready(function($){

/\* Client Carousel \*/

$('.clients-carousel').flexslider({ animation: "slide",

easing: "swing", animationLoop: true, itemWidth: 188,

itemMargin: 0,

minItems: 1,

maxItems: 5,

controlNav: false, directionNav: false, move: 1

});

/\* Back To Top \*/ jQuery('#footer-menu-back-to-top a').click(function(){

jQuery('html, body').animate({scrollTop:0}, 300); return false;

});

/\* Progress Bar \*/

$(function() {

$(".meter > span").each(function() {

$(this)

.data("origWidth", $(this).width())

.width(0)

.animate({

});

});

width: $(this).data("origWidth")

}, 1200);

/\* Tabs \*/ (function() {

var $tabsNav = $('.tabs-nav'),

$tabsNavLis = $tabsNav.children('li'),

$tabContent = $('.tab-content');

$tabsNav.each(function() { var $this = $(this);

$this.next().children('.tab-content').stop(true,true).hide()

.first().show();

$this.children('li').first().addClass('active').stop(true,true).show();

});

$tabsNavLis.on('click', function(e) { var $this = $(this);

$this.siblings().removeClass('active').end()

.addClass('active');

$this.parent().next().children('.tab- content').stop(true,true).hide()

.siblings( $this.find('a').attr('href') ).fadeIn();

e.preventDefault();

});

})();

});

/\* Parallax \*/ jQuery(document).ready(function($){

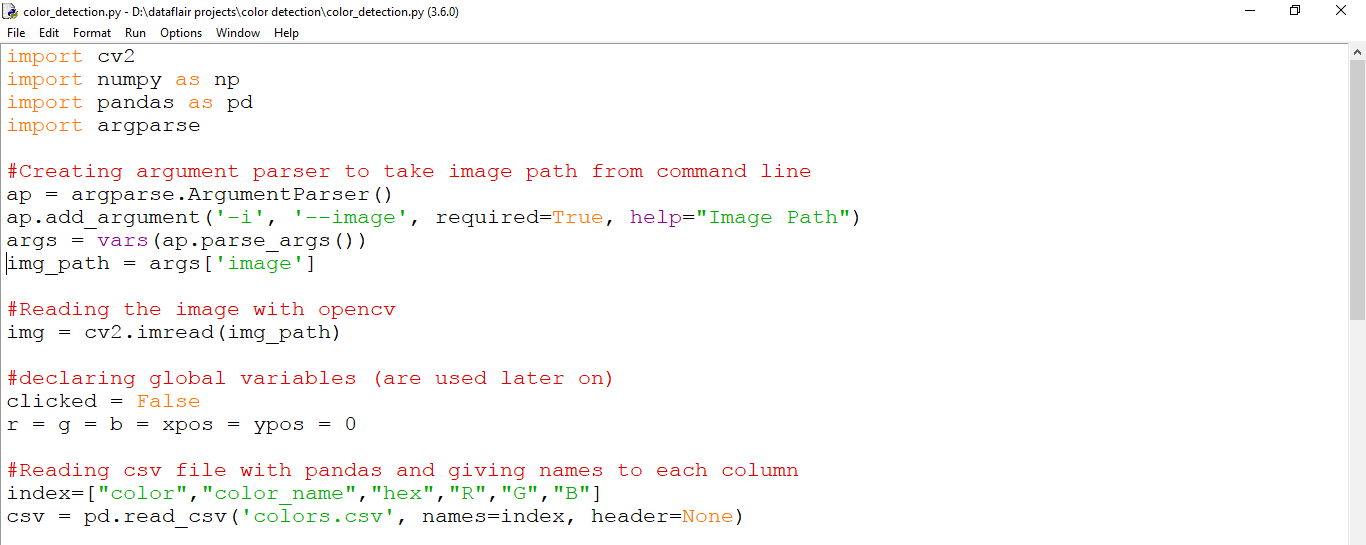
$('#da-slider').cslider({

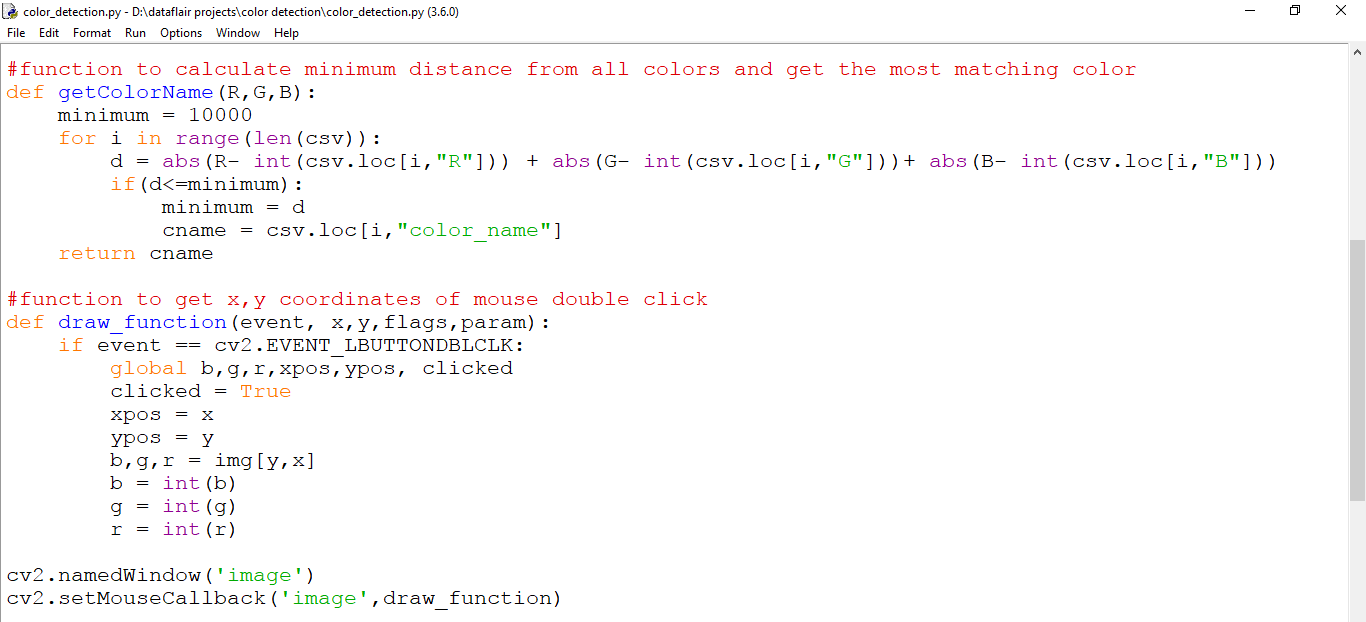
autoplay : true, bgincrement : 450

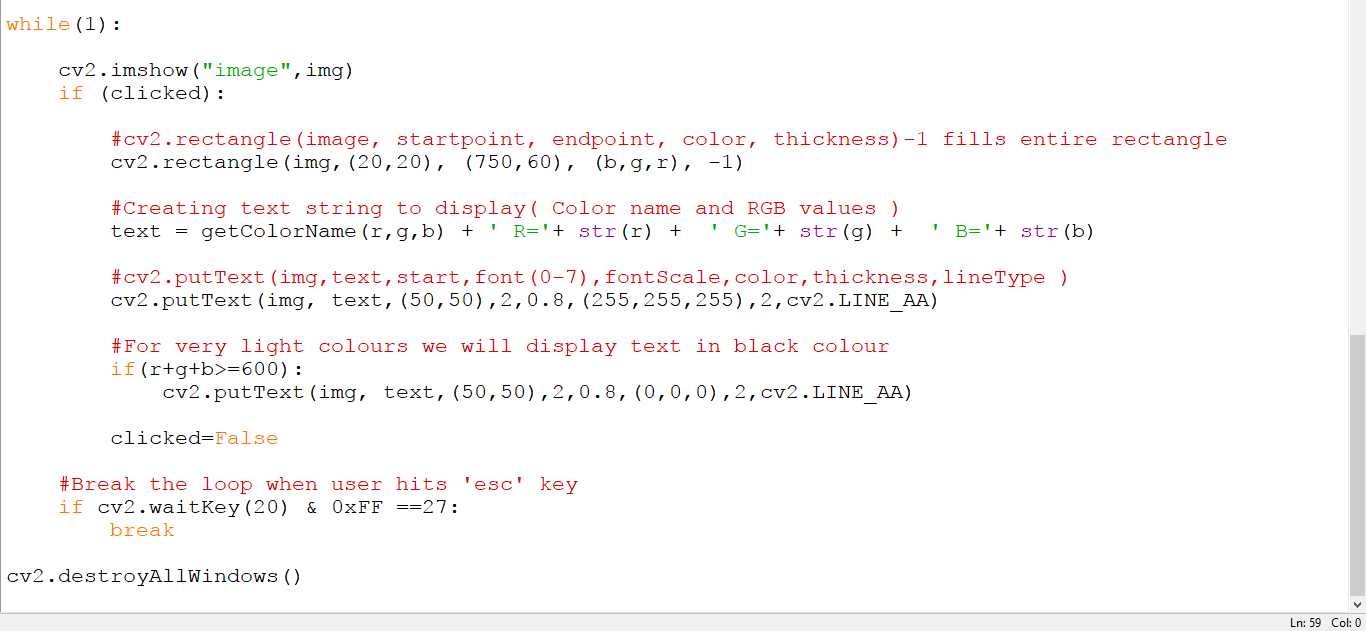
});

});

## SCREENSHOTS:



****



Colors are made up of 3 primary colors; red, green, and blue. In computers, we define each color value within a range of 0 to 255. So in how many ways we can define a color? The answer is 256\*256\*256 = 16,581,375. There are approximately 16.5 million different ways to represent a color. In our dataset, we need to map each color’s values with their corresponding names. But don’t worry, we don’t need to map all the values. We will be using a dataset that contains RGB values with their corresponding names. The CSV file for our dataset has been taken from this link:

[**Colors Dataset**](https://github.com/codebrainz/color-names/blob/master/output/colors.csv)

The colors.csv file includes 865 color names along with their RGB and hex values.

#!/usr/bin/env python

import sys

import re

from xml.sax.saxutils import escape

from optparse import OptionParser

def parse\_files(\*files):

def simplify\_hex\_color(hx):

hx = hx.strip().strip('#').strip()

if len(hx) == 6 and \

hx[0] == hx[1] and hx[2] == hx[3] and hx[4] == hx[5]:

hx = hx[0] + hx[2] + hx[4]

return '#' + hx

def hex\_to\_rgb\_triplet(hx):

hx = hx.strip().strip('#')

if len(hx) == 3:

hx = hx[0]\*2 + hx[1]\*2 + hx[3]\*2

r,g,b = hx[0:2], hx[2:4], hx[4:6]

r,g,b = int(r, 16), int(g, 16), int(b, 16)

return r,g,b

def make\_identifier(name):

ident = re.sub('[^a-zA-Z0-9\_]+', '\_', name)

ident = re.sub('[\_]+', '\_', ident)

ident = ident.strip('\_')

if len(ident) == 0:

ident = '\_'

if ident[0].isdigit():

ident = '\_' + ident

return ident.lower()

def split\_line(line):

fields = line.split('\t')

return (make\_identifier(fields[0]),

fields[0],

simplify\_hex\_color(fields[1]),

hex\_to\_rgb\_triplet(fields[1]))

def process\_line(line):

if len(line) > 0 and line[0].isspace(): return line.strip()

else: return ''

all\_lines = []

for fn in files:

content = open(fn).read()

lines = content.split('\n')

lines = [process\_line(l) for l in lines]

lines = [l.strip() for l in lines if l.strip()]

lines = [split\_line(l) for l in lines]

all\_lines.extend(lines)

out\_dict = {}

for line in all\_lines:

out\_dict[line[0].strip()] = {

"name": line[1].strip().title(),

"hex": line[2].lower().strip(),

"rgb": line[3]

}

return out\_dict

def to\_json(color\_dict):

indent = ' '

code = '{\n'

for k, v in sorted(color\_dict.items()):

code += indent + '"%s": {\n' % k

code += indent\*2 + '"name": "%s",\n' % v["name"]

code += indent\*2 + '"hex": "%s",\n' % v["hex"]

code += indent\*2 + '"rgb": [%d, %d, %d],\n' % v["rgb"]

code += indent + '},\n'

code += '}\n'

return code

def to\_conf(color\_dict):

code = ''

for k, v in sorted(color\_dict.items()):

code += '[%s]\n' % k

code += 'name=%s\n' % v["name"]

code += 'hex=%s\n' % v["hex"]

code += 'rgb=%d;%d;%d\n\n' % v["rgb"]

return code

def to\_sexp(color\_dict):

indent = ' '

code = '(\n'

for k, v in sorted(color\_dict.items()):

code += indent + "(%s '%s' '%s' (%d %d %d))\n" % (k, v["name"], v["hex"],

v["rgb"][0], v["rgb"][1], v["rgb"][2])

code += ')\n'

return code

def to\_xml(color\_dict):

code = '<?xml version="1.0" encoding="utf-8"?>\n'

code += '<colors>\n'

indent = ' '

for k, v in sorted(color\_dict.items()):

name = escape(v["name"])

code += indent + '<color id="%s" hex="%s" red="%d" green="%d" blue="%d">' % (

k, v["hex"], v["rgb"][0], v["rgb"][1], v["rgb"][2])

code += name + '</color>\n'

code += '</colors>\n'

return code

def to\_csv(color\_dict):

code = ''

for k, v in sorted(color\_dict.items()):

code += '%s,' % k

code += '"%s",' % v["name"]

code += '%s,' % v["hex"]

code += '%d,%d,%d\n' % v["rgb"]

return code

def to\_ccode(color\_dict):

indent = ' '

code = '/\* This file is auto-generated, do not edit. \*/\n\n'

code += '#define COLOR\_NAMES\_MAX %d\n\n' % len(color\_dict.items())

code += 'typedef struct ColorInfo {\n'

code += indent + 'const char \*name;\n'

code += indent + 'const char \*hex;\n'

code += indent + 'struct {\n'

code += indent\*2 + 'unsigned char r;\n'

code += indent\*2 + 'unsigned char g;\n'

code += indent\*2 + 'unsigned char b;\n'

code += indent + '} rgb;\n'

code += '} ColorInfo;\n\n'

code += 'typedef enum Color {\n'

first = True

for k, v in sorted(color\_dict.items()):

if first:

code += indent + 'COLOR\_%s=0,\n' % k.upper()

first = False

else:

code += indent + 'COLOR\_%s,\n' % k.upper()

code = code.rstrip()

if code.endswith(','):

code = code[0:-1]

code += '\n'

code += '} Color;\n\n'

code += 'const ColorInfo color\_data[COLOR\_NAMES\_MAX] = {\n'

for k, v in sorted(color\_dict.items()):

code += indent + '{ "%s", "%s", { %d, %d, %d } },\n' % (

v["name"], v["hex"], v["rgb"][0], v["rgb"][1], v["rgb"][2])

code = code.rstrip()

if code.endswith(','):

code = code[0:-1]

code += '\n};\n'

return code

def to\_html(color\_dict):

code = '''<html>

<head>

<title>List of Colors</title>

<style type="text/css">

body, h1, table, tr, th, td {

font-family: monospace;

}

table, th, td {

border: 1px #DCDCDC solid;

}

th {

background-color: #D8D8D8;

}

.swatch {

width: 64px;

}

.color\_hex, .color\_rgb {

text-align: center;

}

</style>

</head>

<body>

<h1>List of Colors</h1>

<p>Source:

<a href="http://en.wikipedia.org/wiki/List\_of\_colors">http://en.wikipedia.org/wiki/List\_of\_colors</a>

</p>

<table cellpadding="2" cellspacing="1" border="1">

<tr>

<th>Swatch</th>

<th>Name</th>

<th>HTML</th>

<th>R</th><th>G</th><th>B</th>

</tr>

'''

for k, v in sorted(color\_dict.items()):

code += ' <tr>\n'

code += ' <td class="swatch" style="background-color: %s;">&nbsp;</td>\n' % v["hex"]

code += ' <td>%s</td>\n' % escape(v["name"])

code += ' <td class="color\_hex">%s</td>\n' % v["hex"]

code += ' <td class="color\_rgb">%d</td>\n' % v["rgb"][0]

code += ' <td class="color\_rgb">%d</td>\n' % v["rgb"][1]

code += ' <td class="color\_rgb">%d</td>\n' % v["rgb"][2]

code += ' </tr>\n'

code += ' </table>\n </body>\n</html>'

return code

FORMATS = {

"ccode": to\_ccode,

"conf": to\_conf,

"csv": to\_csv,

"html": to\_html,

"json": to\_json,

"sexp": to\_sexp,

"xml": to\_xml,

}

def main(args):

p = OptionParser(usage='%prog [options] FILE...',

description='This utility parses the colour list from ' +

'Wikipedia and outputs to various formats. Select the ' +

'table in Wikipedia using your browser and paste to a ' +

'text file and then run this program with that FILE. ' +

'If the data comes from anywhere else, just make sure ' +

'that all of the lines containing the data start with ' +

'a space and are tab-separated, and that the first 2 ' +

'fields are the name of the colour and the RGB hex ' +

'triplet.',

epilog='For "-o", "-" (stdout) is the default. ' +

'For "-f", "json" is the default')

p.add\_option('-o', metavar='FILE', dest='outfile', default='-',

help='file to write to ("-" for stdout)')

p.add\_option('-f', metavar='FMT', dest='format', default='json',

help='format to output ("ccode", "conf", "csv", "html", "json", "sexp", "xml")')

opts, args = p.parse\_args(args)

fmt = opts.format.lower()

if fmt not in FORMATS:

p.error("unknown format '%s', expect one of ccode, conf, csv, html, json, sexp, xml\n" % fmt)

do\_close = False

if opts.outfile == '-':

outfile = sys.stdout

else:

try:

outfile = open(opts.outfile, 'w')

do\_close = True

except IOError as e:

p.error("unable to open file '%s': %s" % (opts.outfile, e.message))

if len(args) < 2:

p.error("no input file(s) specified")

color\_dict = parse\_files(\*args[1:])

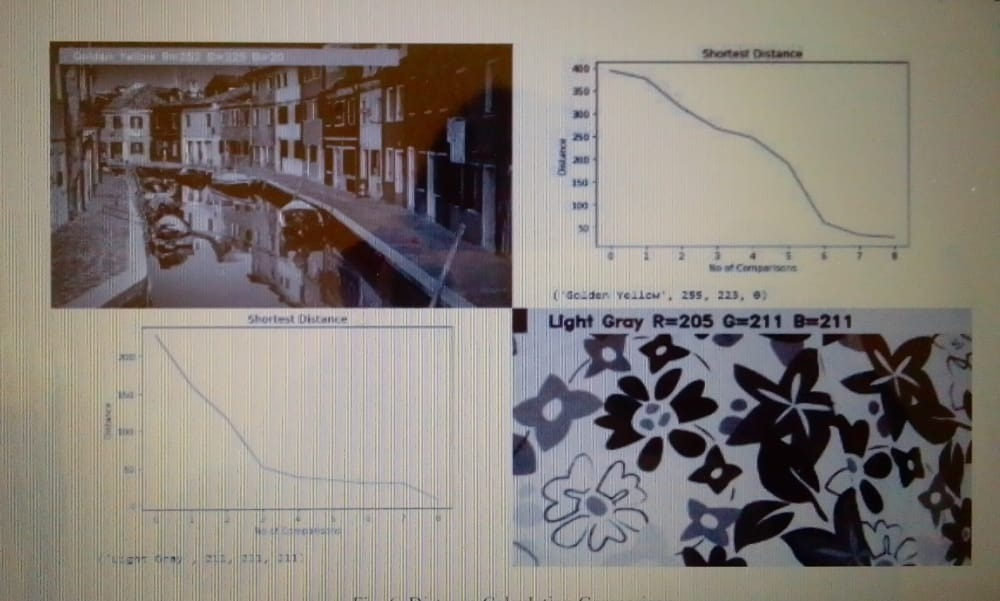
outfile.write(FORMATS[fmt](color\_dict))

if do\_close:

outfile.close()

return 0

if \_\_name\_\_ == "\_\_main\_\_": sys.exit(main(sys.argv))



# 6. TESTING

## INTRODUCTION TO TESTING

* + 1. **Software Testing**

In any software development, testing is a process to show the correctness of program and it needs the design specifications. Testing is needed to prove correctness completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary to ensure completeness of testing, improve the quality of software and reduce the testing costs and to reduce study needs and operation time.

**Goals of Testing**

The following are goals of testing:

* + - 1. Testing is a process of executing a program with the intent of finding error.
      2. A good test case is the one that has a high probability of finding an as at undiscovered error.
      3. A successful test is one that uncovers an as at undiscovered error.

### Testing Methodology Black box testing

Black Box Testing is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.

### White box testing

White Box Testing is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually, the developers are involved in the white box testing.

### Gray box testing

Gray Box Testing is the process in which the combination of black box and white box techniques is use.

## Levels of Testing

### Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

### Test objectives

* + All field entries must work properly.
  + Pages must be activated from the identified link.
  + The entry screen, messages and responses must not be delayed.

### Features to be tested

* + Verify that the entries are of the correct format
  + No duplicate entries should be allowed
  + All links should take the user to the correct page.

## Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications,

e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## Acceptance testing

This is the final stage in the testing process before the system is accepted for operational use. Acceptance testing may reveal errors and omissions in the system requirements definition because real data exercises the system in different ways from the test data.

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

## Regression testing

Regression testing is actually that helps to ensure changes that don’t introduce unintended behavior as additional errors. Regression testing may be conducted manually by executing a subset of all test cases or using automated capture play back tools.

Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes. Regression testing is a normal part of the program development process and, in larger companies, is done by code testing specialists.

## 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 centred on the following items:

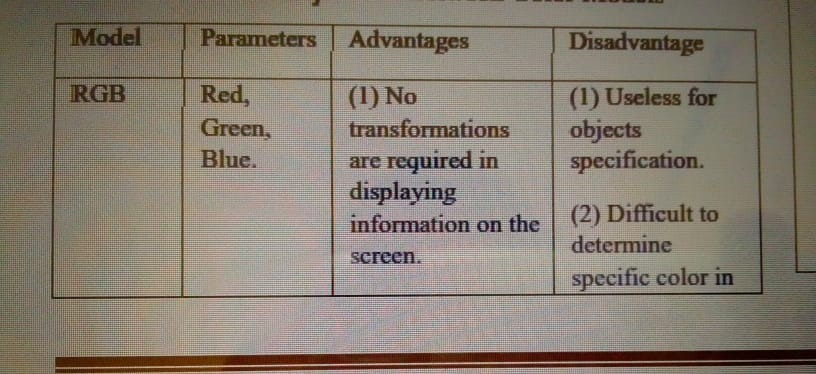
Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

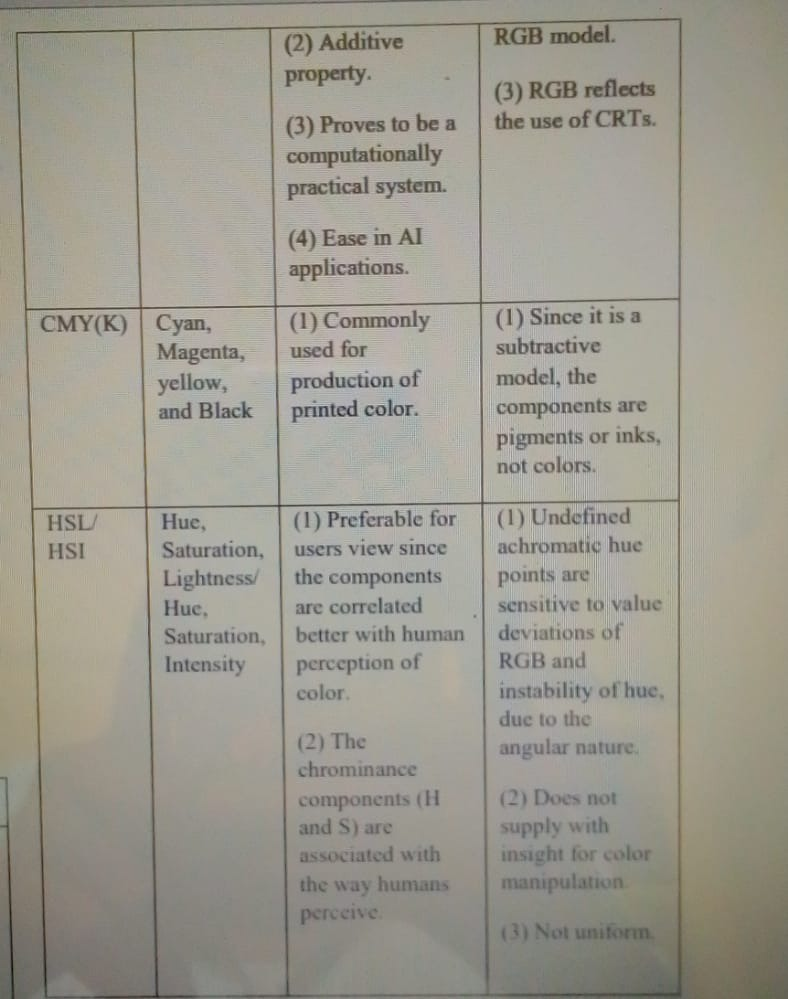
Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

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.

## SAMPLE TEST CASES:

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.



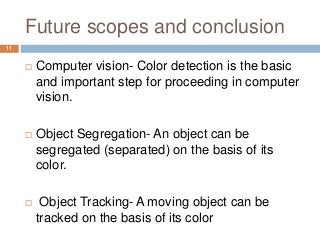


# CONCLUSION

The Color Detection done using computer vision has been a achieved. The GUI has been successful in giving the users an option to choose from thus returning the correct RGB values when the picture is double clicked over a certain color. Using the algorithm of mapping RGB values by finding the shortest distance and managed mean in the three, the accuracy calculated by the error in RGB distance has hit 97%. Color Detection in real life is a blooming concept. But there is still much to uncover. With changing technology in a fast paced and impatient world, we can look forward to heights that now seem unattainable. For instance, the interest of fashion designers or graphic designers in detecting and using certain colors that they are unable to name or find. The future scope of this project ranges from imbuing AI with being able to classify whether the colors detected are completely formal or do the hold the parameters of a device constraining the very aspect of accuracy that must be achieved and thus the color detection might in fact be faulty

# 8.FUTURE ENHANCEMENT

In this Python project with source code, we learned about colors and how we can extract color RGB values and the color name of a pixel. We learned how to handle events like double-clicking on the window and saw how to read CSV files with pandas and perform operations on data. This is used in numerous image editing and drawing apps.



# 9.REFERENCES

[1] Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Pearson (2018)

[2] Project in Python – Color Detection using Pandas & OpenCV. <https://data->

flair.training/blogs/project-in-pythoncolor-detection/

[3]Jake Frankenfield, Gordon Scott. Artificial Intelligence (AI).

<https://www.investopedia.com/terms/a/artificial>

intelligenceai.asp#:~:text=Artificial%20intelligence%20(AI)%20refers% 20to,as%20learning%20and%20problem%2Dsolving.

[4] Victor Wiley, Thomas Lucas. Computer Vision and Image Processing: A Paper Review. International Journal Of Artificial Intelegence Research. ISSN: 2579-7298, Vol 2, No 1, June 2018, pp. 28-36 DOI: 10.29099/ijair.v2i1.42

[5] Computer Vision. What it is and why it matters. https://www.sas.com/en\_in/insights/analytics/computervision.html

[6] R. S. Berns, “Principles of Color Technology” (3rd edition New York: Wiley, 2000)

[7] Yang, F., Lu, H., Zhang, W., Yang, G.: ‘Visual tracking via bag of features’, IET Image Process., 2012, 6, pp. 115–128 (doi: 10.1049/iet-ipr.2010.0127)

[8] Hasting, G. & Rubin, Alan. (2012). Color spaces - a review of historic and modern color models\*. African Vision and Eye Health. 71. 10.4102/aveh.v71i3.76.

[9] Behic Guven. Building a Color Recognizer in Python. Towards Data Science. https://towardsdatascience.com/building-a-color-recognizerin-python-4783dfc72456

[10] G. Wyszecki and W. S. Styles, “Color Science: Concepts and Methods, Quantitative Data and Formulae” (2nd edition New York: Wiley, 1982)

[11] Berns, RS & Reiman DM, “Color managing the third edition of Billmeyer and Saltzman's Principles of Color Technology”, Color Research & Application, Vol.27, No.5, (2002), pp.360-373.

[12] Gonzalez RC, Woods RE & Eddins S.L, Digital Image Processing Using MATLAB, Pearson Education, Inc, (2004).

[13] Arash Abadpour and Shohreh Kasaei. Principal Color and Its Application to Color Image Segmentation. <http://abadpour.com/files/html/178si/W178si.html>

[14] Senthamaraikannan D, Shriram S & William J, “Real time color recognition”, International Journal of Innovative Research In Electrical,Electronics, Instrumentation And Control Engineering, Vol.2, No.3,(2014).

[15] Karan Bhanot. Color Identification in Images. Towards Data Science. https://towardsdatascience.com/color-identificationin-images-machine-learning-application-b26e770c4c71

[16] Ray Siddheswar, and Rose H. Turi. "Determination of number of clusters in k-means clustering and application in color image segmentation." In Proceedings of the 4th international conference on advances in pattern recognition and digital techniques, pp. 137-143. 1999.

[17] Shun-Hung Tsai, Yu-Hsiang Tseng, A novel color detection method based on HSL color space for robotic soccer competition, Computers & Mathematics with Applications, Volume 64, Issue 5, 2012, Pages 1291-1300, ISSN 0898- 1221, <https://doi.org/10.1016/j.camwa.2012.03.073>.

[18] John Kennedy, Steven White. CMY and CMYK Color Spaces. https://docs.microsoft.com/enus/windows/win32/wcs/cmy-and-cmyk-color-spaces

[19] OpenCV Python Tutorial – Implementation of Computer Vision https://data-flair.training/blogs/opencv-pythontutorial/

[20] Resti, Yulia & Burlian, Firmansyah & Yani, Irsyadi & Rosiliani, Dinda. (2020). Analysis of a cans waste classification system based on the CMYK color model using different metric distances on the k-means method. Journal of Physics: Conference Series. 1500. 012010. 10.1088/1742- 6596/1500/1/012010.

[21] Cheng-Jin Du, Da-Wen Sun, 18 - Quality Evaluation of Pizzas, Editor(s): Da-Wen Sun, In Food Science and Technology, Computer Vision Technology for Food Quality Evaluation, Academic Press, 2008, Pages 427-446, ISBN 9780123736420, https://doi.org/10.1016/B978-012373642- 0.50021-1.