Project Report
on
COLOR DETECTION

Submitted by :

Y.Tulasi Meghana (R170941)
J.S.Thanuja (R170100)
H.Vandana (R171152)

Under the guidance of :

Ms. SK.Shabana

Assistant professor

Computer Science and Engineering Department

Rajiv Gandhi University of Knowledge Technologies AP-IIIT,
RK Valley,Idupulapaya, Kadapa - 516 330 Andhra Pradesh, India.

CERTIFICATE

This is to certify that the project work titled “COLOR DETECTION” is a bonafied project work submitted by Y.Tulasi Meghana (R170941), J.S.Thanuja (R170100),H.Vandana (R171152) in the department of COMPUTER SCIENCE AND ENGINEERING in partial fulfillment of requirements for the award of degree of Bachelor of Technology in Computer science and engineering for the year 2021-2022 carried out the work under the supervision of Ms.SHABANA.

P.Harinadha
Head of CSE department

Projectguide
Ms.SK.Shabana

.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I also express my sincere gratitude to our respected Head of the Department P.HARINADHA for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Ms.SK.SHABANA for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

1. ABSTRACT

Color detection 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. We use concepts of Computer vision in this project. Teaching the computers how to learn from images and videos. Pandas and OpenCV libraries used in python languages. We use Shortest distance algorithm to find the minimum distance between the datapoints Open Source Computer Vision Library. OpenCV was designed for computational efficiency and with a robust specialize in real-time applications.

2. INTRODUCTION TO COLOR DETECION

In this color detection python project, We are going to build the application in python language and get results to use the pandas and OpenCV libraries are used in this project which can automatically detect the name of the color. So we have a dataset which have color names and its value.

Then we will calculate the distance from each color and find the shortest one,With the advancement of AI Computer vision came into the image in late 1960s. Its whole purpose was to extend the intellect of the synthetic mechanism available by installing the cameras into them and describe whatever they saw just like humans' sensory systems.

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

WHAT IS COLOR DETECTION:

In this color detection Python project, we are going to build an application through which you can automatically get the name of the color by clicking on them.So for this, we will have a data file that contains the color name and its values. Then we will calculate the distance from each color and find the shortest one.

3.PURPOSE :

In this color detection Python project, we are going to build an application through which you can automatically get the name of the color by clicking on them.So for this, we will have a data file that contains the color name and its values. Then we will calculate the distance from each color and find the shortest one.

4.SCOPE :

The color sensor detects the color of the surface, usually in the RGB scale. Color is the result of interaction between a light source, an object, and an observer. Color sensors have a variety of applications including detection of the environment, choosing the right product and sorting. There are different types of colour detection sensors: brightness, contrast, luminescence, RGB, print mark sensors.

ADVANTAGES OF COLOR DETECTION:

- ➔ It helps in sorting of objects based on three color approach. It also helps in counting of objects. Automated system can be built using color sensors which help in completion of work in less time. Moreover human intervention is not needed.

- ➔ Powerful and large memory color sensor ICs are available at low cost. This has driven its use in many applications.

- ➔ It is easy to change or modify manufacturing setups without even re-programing the sensor device. This is beneficial in low volume manufacturing applications having frequent color variations.

➔ With the advancement of technology and memory loaded with color intensity data, color sensor controller can store and can make color matching decisions on unlimited number of colors virtually.

DISADVANTAGES OF COLOR DETECTION:

➔ The approach is costly for small scale industries.

➔ It does color matching or identification in applications requiring only pass/fail output.

5.REQUIREMENT SPECIFICATION:

SOFTWARE REQUIREMENTS:

Programming languages	**:**	**Python version 3.7**
Packages	**:**	**Pandas,Open CV,argparse**
Operating system	**:**	**Windows,linux**
ram(random access memory)	**:**	**minimum(4gb)**

HARDWARE REQUIREMENTS:

- **RAM :2GB(or)above**
- **Graphic card:500MB(or)above**
- **Hard disk:2TB**

The following libraries are to be installed :

IDLE python:

IDLE is Python's Integrated Development and Learning Environment.IDLE is integrated development environment (IDE) for editing and running Python 2.x or Python 3programs. The IDLE GUI (graphical user interface) is automatically installed with the Python interpreter. IDLE was designed specifically for use with Python.

IDLE has a number of features to help you develop your Python programs including powerful syntax highlighting. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously. Output windows, such as used for Edit / Find in Files, are a subtype of edit window. They currently have the same top menu as Editor windows but a different default title and context menu. IDLE's menus dynamically change based on which window is currently selected. Each menu documented below indicates which window type it is associated with.

OpenCV-Python :

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any

readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV- Python works, it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increasesnumber of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision process.

Pandas :

Pandas is a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries.

Pandas is built on top of two core Python libraries—matpotlib for data visualization and Numpy for mathematical operations. Pandas acts as a

wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code. For instance, pandas' `.plot()` combines multiple matplotlib methods into a single method, enabling you to plot a chart in a few lines.

Before pandas, most analysts used Python for data munging and preparation, and then switched to a more domain specific language like R for the rest of their workflow. Pandas introduced two new types of objects for storing data that make analytical tasks easier and eliminate the need to switch tools: Series, which have a list-like structure, and DataFrames, which have a tabular structure

DATAFRAMES:

While series are useful, most analysts work with the majority of their data in DataFrames. DataFrames store data in the familiar table format of rows and columns, much like a spreadsheet or database. DataFrames makes a lot of analytical tasks easier, such as finding averages per column in a dataset.

You can also think of DataFrames as a collection of series—just as multiple columns combined make up a table, multiple series make up a DataFrame.

6.Analysis and Design :

Analysis:

Today’s project will be exciting and fun to build. We will be working with colors and we will get to learn about many concepts throughout this project.

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

SYSTEM DESIGN

The most creative and challenging phase of the life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementations of the candidate system. The design may be defined as **"**the process of applying various techniques and principles for the purpose of defining a device, a process or a system with sufficient details to permit its physical realization**"**.

DATA FLOW DIAGRAM

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program.

Data flow

Data move in a specific direction from an origin to a destination.

Data Flow Diagram Levels:

A data flow diagram has been classified into different levels. The different levels are as follows:

1. Context level or Level-0
2. Top level or Level-1

1.Context level diagram:

This level shows the overall context of the system and its operating environment and shows the whole system as just one process then get the response on the window.

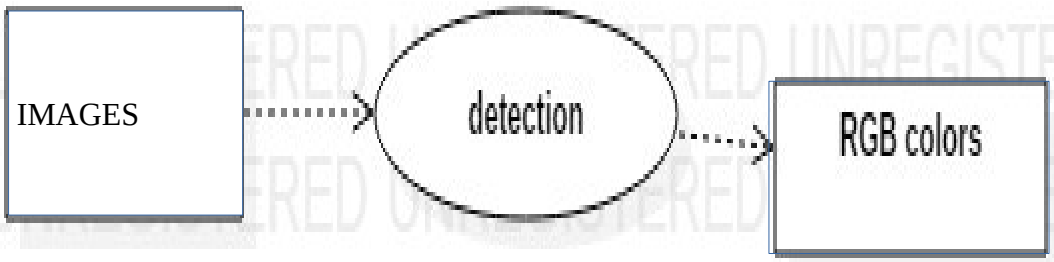


Fig: content level / Level 0 data flow diagram

2. Top level diagram:

This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major high-level processes of the system and their interrelation. A process model will have one and only one level-1 diagram. A level -1 diagram must be balanced with its parent context level diagram.

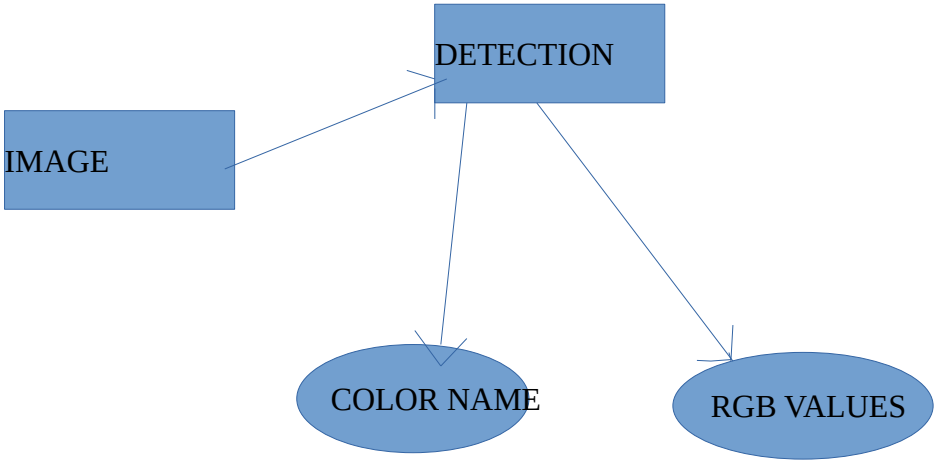


Fig : level 1 data flow diagram

6.1 UML Diagrams

The Unified Modeling Language (UML) is a graphical language for visualizing, specifying,
constructing and documenting of a software intensive system. The UML gives a standard way to write a system blueprints, covering conceptual things, such as classes written in a specified programmed
language, database schemas and reusable software components

Use case Diagram

ER diagram

The above diagrams are described below with explanation with their diagrams.

Use-case Diagram:

A use-case in software engineering and systems engineering is a description of a system behavior as it responds to a request that originates from outside of that system. In other words, a use case describes “who” can do “what” with the system in question. The use case technique is used to capture a system’s behavioral requirements by detailing scenario-driven threads through the functional requirements.

In software engineering a use case in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

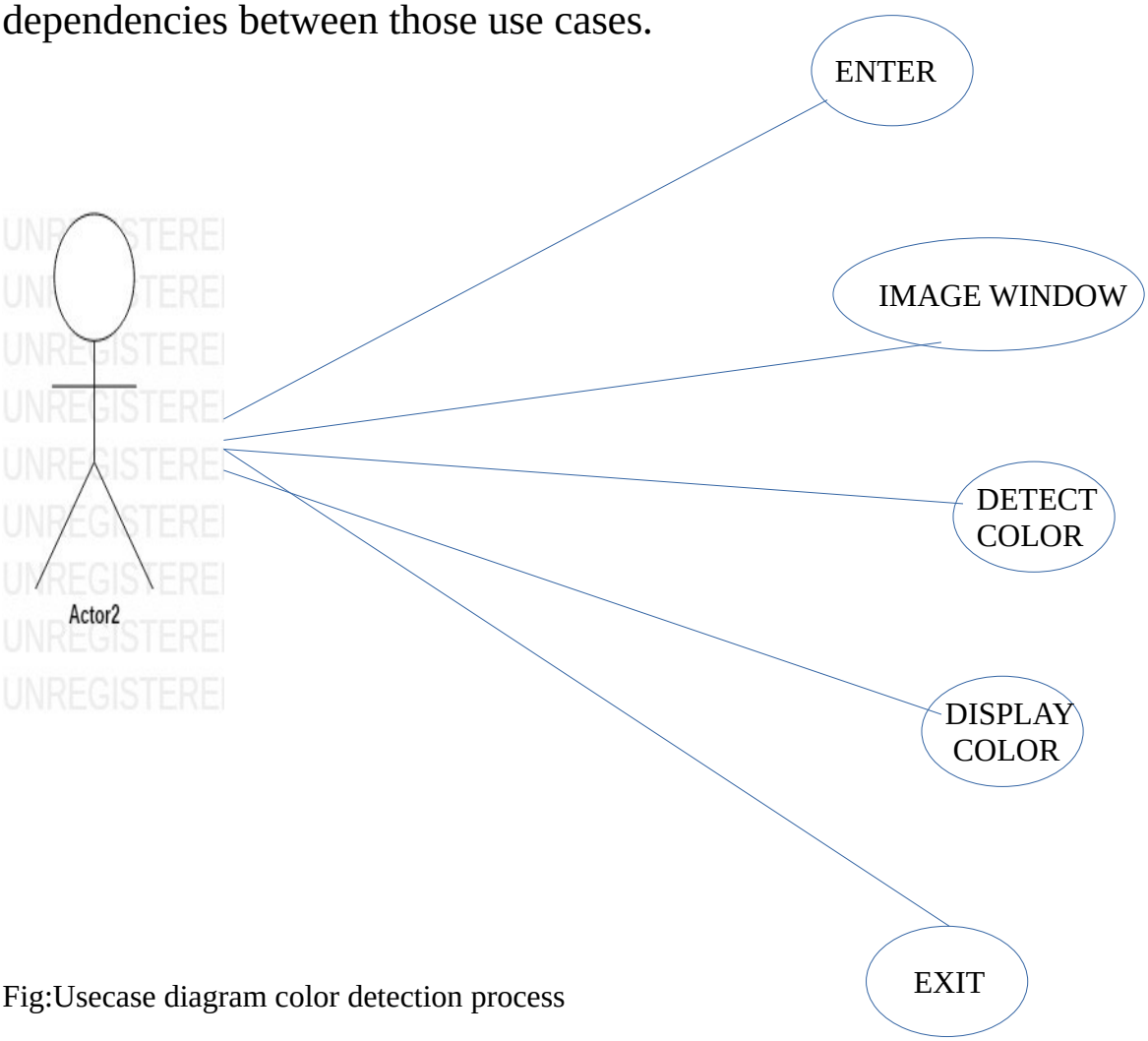
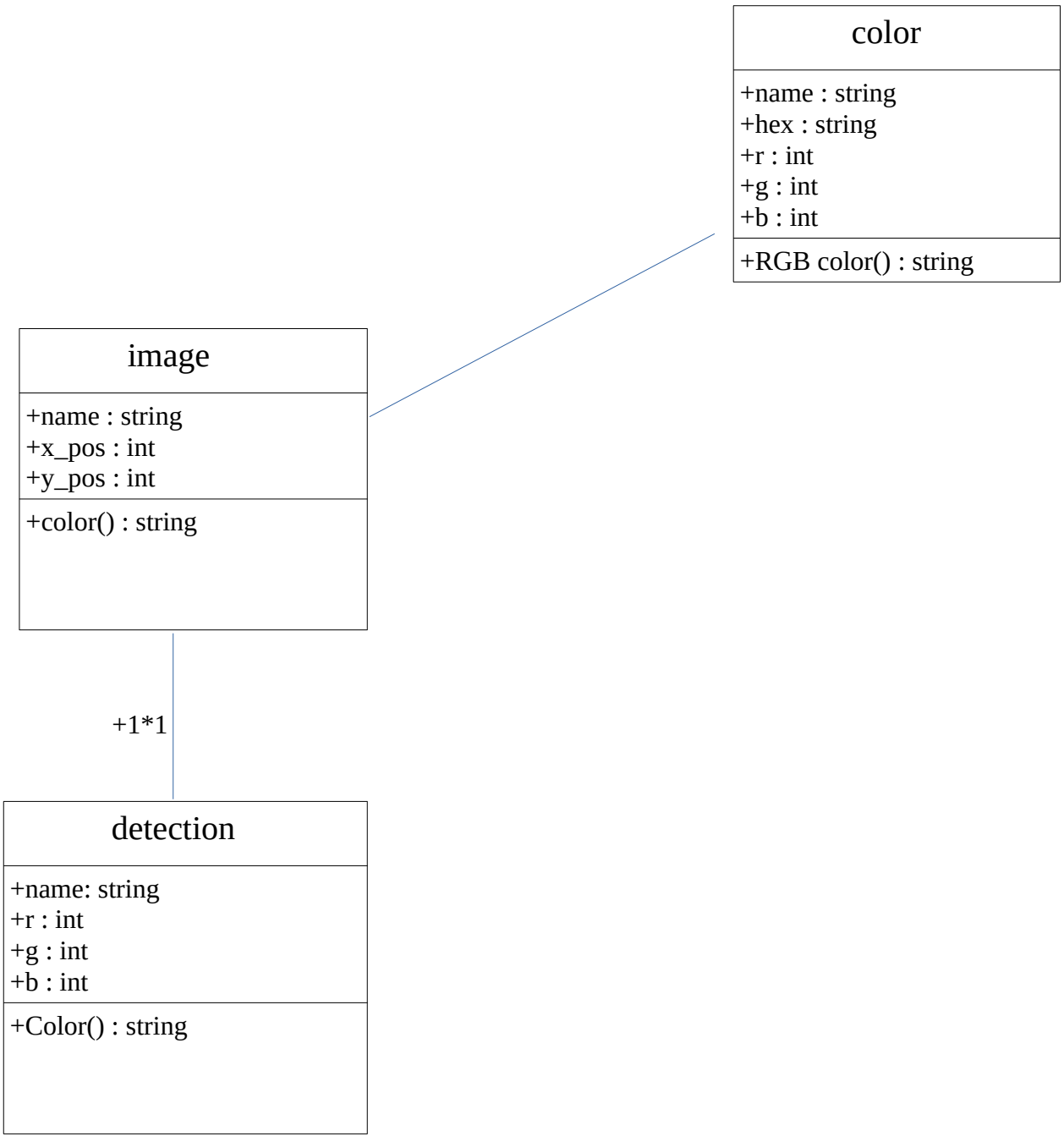


Fig:Usecase diagram color detection process

6.2 ER diagram :



7.

Color Detection Project using python:

The Dataset

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.

--> Taking an image from the user

Using argparse we take input image 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)
```

-->Pandas used to read the csv datasetfile:

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

```
index=["color","color_name","hex","R","G","B"]
csv = pd.read_csv('colors.csv', names=index, header=None)
```

-->Doubleclick event on image 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)
```

With these lines, we named our window as ‘image’ and set a callback function which will call the **draw_function()** whenever a mouse event occurs.

-->Calculating the x and y positions along with R,G,B values of the selected pixel:

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)
```

-->Shortest distance algorithm:

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
```

-->Running the application:

Open command prompt and run the python code file with the image path as argument to it.

The while loop starts and image window will be opened and double click anywhere on the window to detect the colors.

```
while(1):

    cv2.imshow("image",img)
    if (clicked):

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

-->Exiting the application:

The while loop runs forever ,so to stop it we use the ESC key to destroy the windows.

8.Implementation and Testing :

After all phase have been perfectly done, the system will be implemented and the system can be used.

System Testing

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2 .Integration testing

1.Unit Testing :

Unit testing is commenced when a unit has been created and effectively reviewed . In order to test a single module we need to provide a complete environment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses .A procedure to call the functions of the unit under test with appropriate parameters.

Unit testing for image window :

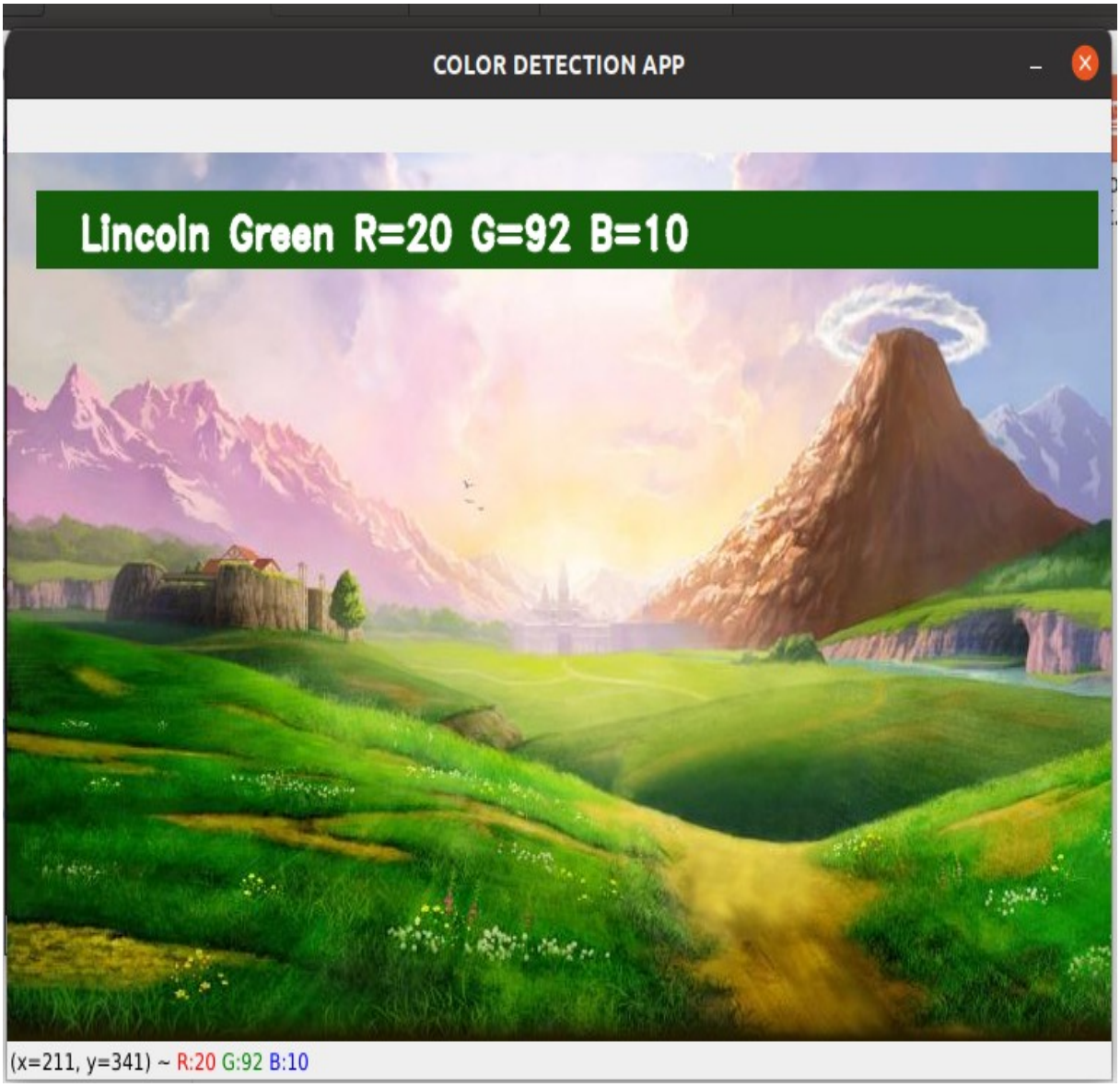
On the image window we check if it is giving color name and its RGB values as we double clicked on any pixel on the image or not.It should be displaying the values without any errors.

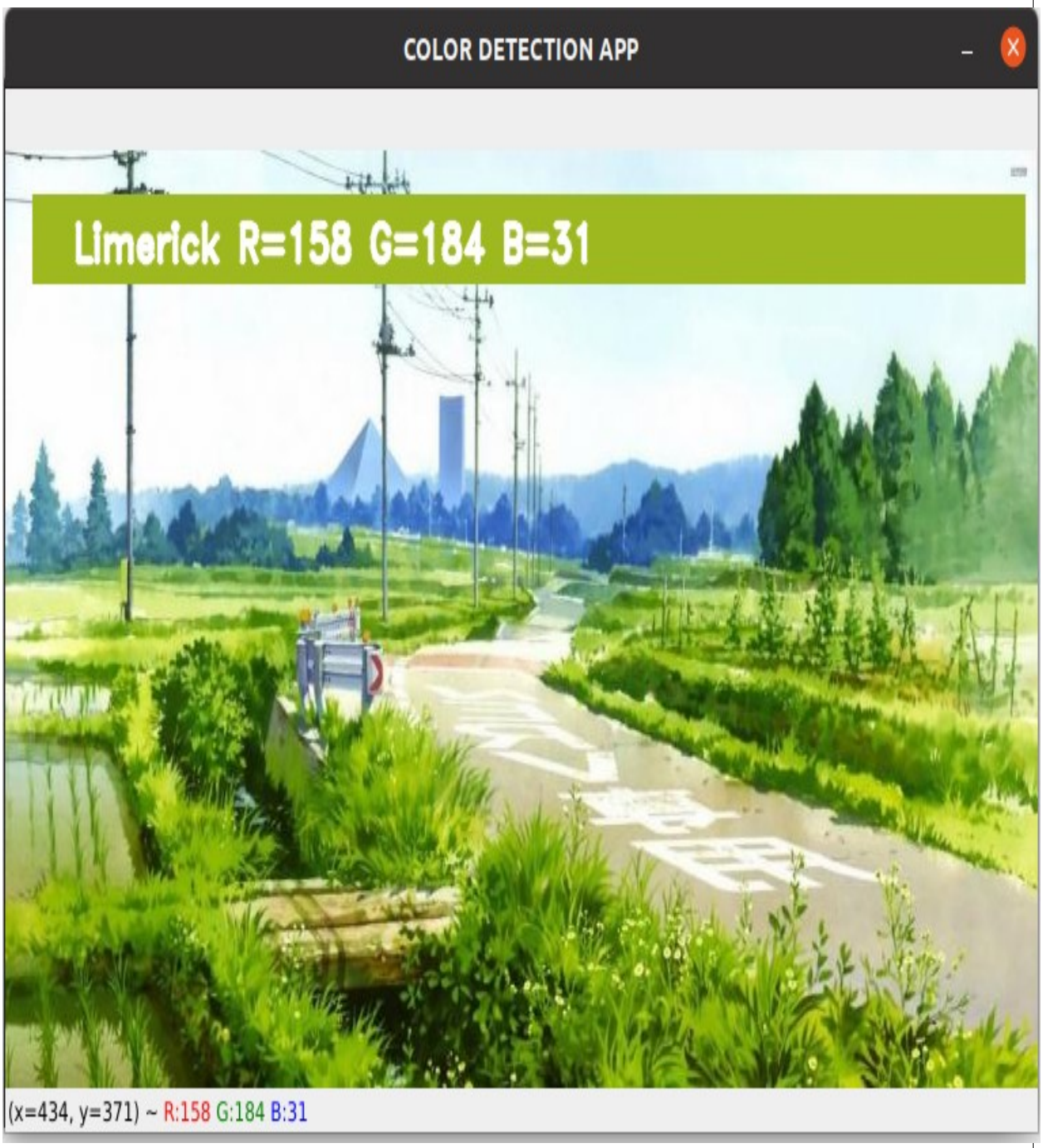
2.Integration Testing :

In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

9. Evaluation:




COLOR DETECTION APP
Limerick R=158 G=184 B=31
(x=434, y=371) ~ R:158 G:184 B:31

10. Conclusion :

Color detection python project is used to recognize colors.In this Research area with all research methodology, future scope, and results we learned about colors and therefore the way we will extract color RGB values and the color name of a pixel. We use CSV files with pandas and apply the OpenCV library to perform operations on data. this is often utilized in numerous image editing and drawing apps. 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. Pandas and OpenCV libraries used in python languages.There is always hope for advancement and enhancement in future.

11. References :

Article on Color Detection of RGB Images published in researchgate:

https://www.researchgate.net/publication/349355136_Color_Detection_of_RGB_Images_Using_Python_and_OpenCv/related#fullTextFileContent

Color detection using python by U Dutta A Raj:

http://103.47.12.35/bitstream/handle/1/9155/BT4033_PR%20-%20Prashant%20Johri.pdf?sequence=1&isAllowed=y