RNS INSTITUTE OF TECHNOLOGY

BENGALURU - 98

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Presentation on Internship

Dominant Color Detection

AGENDA

ABSTRACT

- Image segmentation, image matching, object recognition, visual tracking in the fields of image processing and computer vision all require color detection.
- Reports indicate that with the help of this application we can check the percent of different color.
- Dominant color detection process help us in getting the various other image processing works done easily.
- Dominant color detection software is used in getting dominant colours and this can be used in creating better pictures,image recognition,image processing etc
- This software is used by various editing and drawing apps.

ABOUT THE COMPANY

NASTECH – New Age Solutions & Technologies

- ***NASTECH is formed with the purpose of bridging the gap between Academia and Industry.***
- Nastech is one of the leading Global Certification and Training service providers for technical and management programs for educational institutions.
- They collaborate with educational institutes to understand their requirements and form a strategy in consultation with all stakeholders to fulfill those by skilling , reskilling and upskilling the students and faculties on new age skills and technologies.
- Industry and project oriented student training programs.
- Certification programs mapped to Global Certification Exams from Microsoft/EC- Council/Google/AWS/ Adobe).

INTRODUCTION

- Color detection helps us in recognizing the colors and then this can be used in various image editing apps.
- The proposed system focuses on how to identify the most dominant colors in a particular image with the help of computer vision and deep learning algorithm by using the OpenCV, Tensor flow and pandas library
- The detection is carried out to see which colors are used in a particular image.
- K-means algorithm is used to find the most dominant color.

- **K-Means clustering** is the most popular unsupervised learning algorithm. It is used when we have unlabelled data which is data without defined categories or groups. The algorithm follows an easy or simple way to classify a given data set through a certain number of clusters, fixed apriori.

LITERATURE SURVEY

- According to researchers, 1000 different shades of light are there and in them we can detect 100 different levels of red-green shades. We can also see 100 levels of yellow blue shades. So, total till now 10 million different colors are identified.
- Red, green, and blue are the three primary colors that make up any color available. Each color value in a computer is defined as a number between 0 and 255. A color can be represented in around 16.5 million different ways.
- Colour detectors are majorly used to grade coloured products, distinguish coded markings, detect the presence of adhesives or data codes on a package. The technology has a wide range of applications in various industries such as **textile, automation, automotive, food, printing, pharmaceutical**, and many more.

REQUIREMENTS

❖ HARDWARE REQUIREMENTS

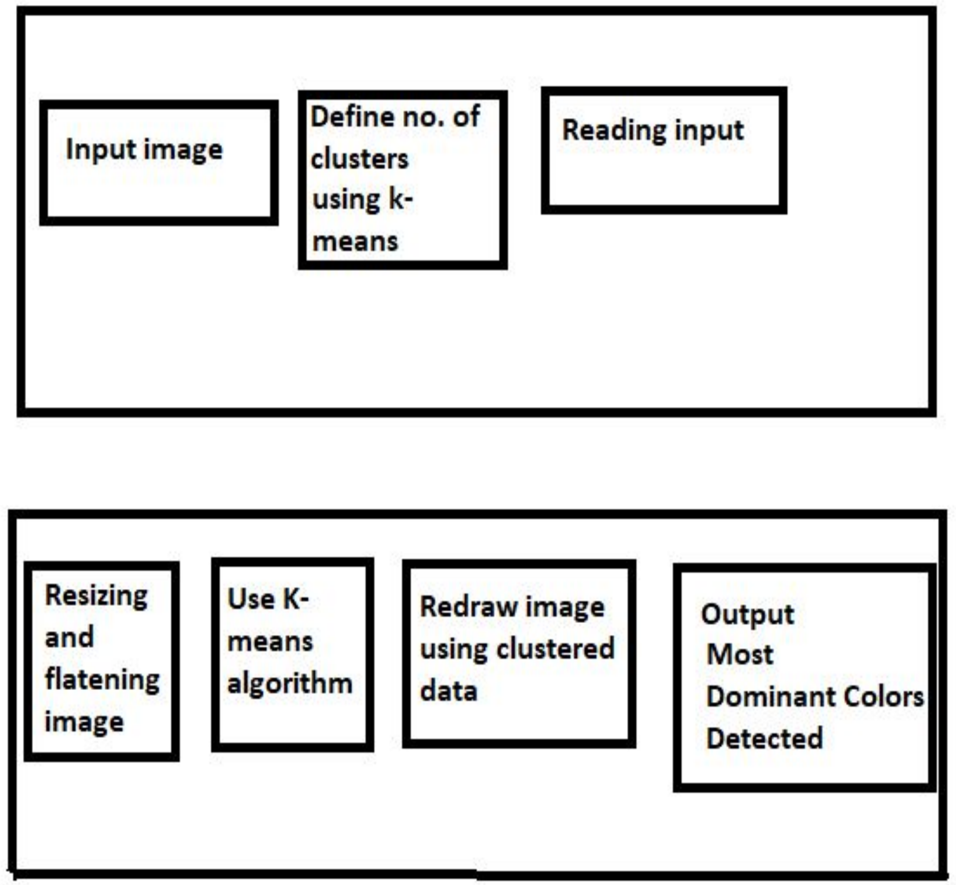
• Processor	: Any Processor above 500 MHz
• RAM	: 512Mb
• Hard Disk	: 2 GB
• Input device	: Standard Keyboard and Mouse
• Output device	: High Resolution Monitor

❖ SOFTWARE REQUIREMENTS

• Operating system	: Windows 10
• IDE	: Anaconda Jupyter Notebook
• Tools/Technologies	: Python, Computer Vision, OpenCv, Numpy library, matplotlib and imutils

SYSTEM DESIGN

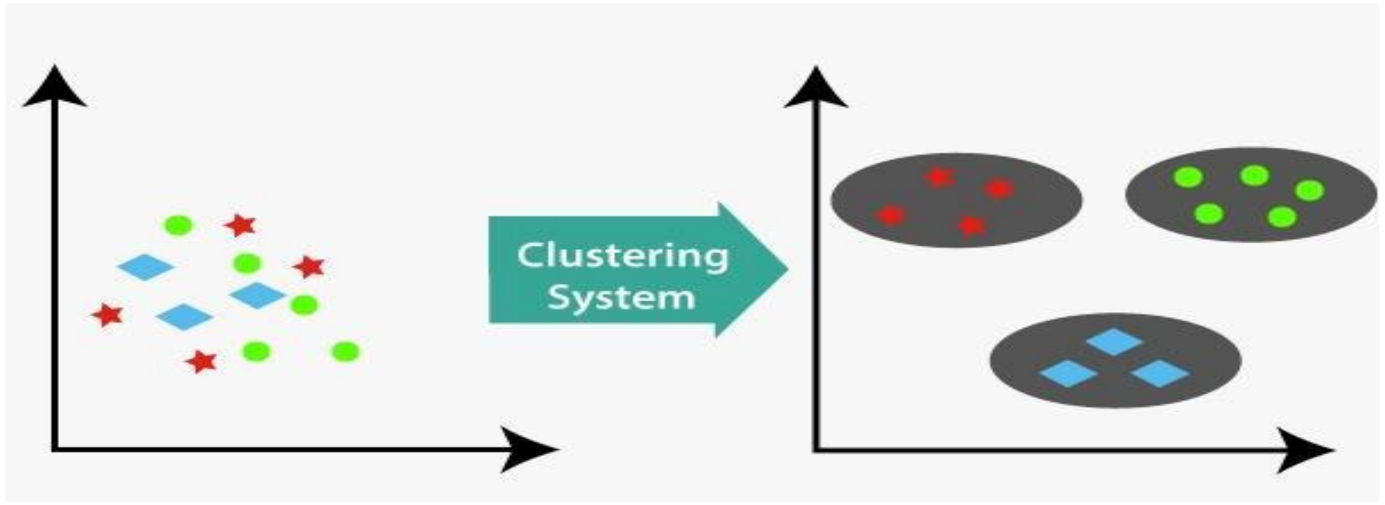
- As depicted in the figure, the project is divided in two phases.
- Phase 1 deals with loading, preprocessing and training the data set. In this phase, first we will give the input image. For the image the no. of clusters will be defined using K-means algorithm.
- After defining clusters, the algorithm will read the input.
- In the second phase of the most dominant color detection, first resizing and flattening of the image is done.
- After flattening, K-means algorithm is used.
- After successfully applying K-means algorithm, we have to redraw the image using clustered data.
- Now, the output will be the most dominant colors.


Input image
Define no. of clusters using k-means
Reading input
Resizing and flatening image
Use K-means algorithm
Redraw image using clustered data
Output Most Dominant Colors Detected

SYSTEM DESIGN

Algorithm 1 k-means algorithm

```
1: Specify the number $k$ of clusters to assign.
2: Randomly initialize $k$ centroids.
3: **repeat**
4:     **expectation:** Assign each point to its closest centroid.
5:     **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.
```



DETAILED DESIGN

FUNCTIONAL MODULES

- The entire project is divided into 3 modules:
  - Image input
  - Image processing
  - Dominant color detection

1. Image input
   - The first step is to import all the required modules along with OpenCV, and then load the image but make sure the image is inside the same folder as the code file.
   - This module takes input as an image whose name we have to mention. "imread" function is used to read the function.
   - Along with reading the input, one copy of image is also made for future use.

2. Image processing
   - After taking the image input, the next part is image processing. In image processing we will resize the image to get more correct results.
   - After resizing, flattening of image is done. We have to fit our image in Kmeans Clustering Algorithm. In this step, the flattened image is working as an array containing all the pixel colors of the image. These pixel colors will now be clustered into 5 groups. These groups will have some centroids which we can think of as the major color of the cluster (In Layman's terms we can think of it as the boss of the cluster).

IMPLEMENTATION

❖ TRAINING CODE

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import imutils
clusters = 5 # try changing it
img = cv2.imread('5.png')
org_img = img.copy()
print('Org image shape --> ',img.shape)
img = imutils.resize(img,height=200)
print('After resizing shape --> ',img.shape)
flat_img = np.reshape(img,(-1,3))
print('After Flattening shape --> ',flat_img.shape)
kmeans = KMeans(n_clusters=clusters,random_state=0)
kmeans.fit(flat_img)
dominant_colors = np.array(kmeans.cluster_centers_,dtype='uint')
percentages =
(np.unique(kmeans.labels_,return_counts=True)[1])/flat_img.shape[0]
p_and_c = zip(percentages,dominant_colors)
p_and_c = sorted(p_and_c,reverse=True)
block = np.ones((50,50,3),dtype='uint')
plt.figure(figsize=(12,8))
for i in range(clusters):
plt.subplot(1,clusters,i+1)
block[:] = p_and_c[i][1][::-1]
plt.imshow(block)
plt.xticks([])
plt.yticks([])
plt.xlabel(str(round(p_and_c[i][0]*100,2))+'%')
bar = np.ones((50,500,3),dtype='uint')
plt.figure(figsize=(12,8))
plt.title('Proportions of colors in the image')
start = 0
i = 1
for p,c in p_and_c:
end = start+int(p*bar.shape[1])
if i==clusters:
bar[:,start:] = c[::-1]
else:
bar[:,start:end] = c[::-1]
```

IMPLEMENTATION

```
start = end
i+=1
plt.imshow(bar)
plt.xticks([])
plt.yticks([])
rows = 1000
cols = int((org_img.shape[0]/org_img.shape[1])*rows)
img =
cv2.resize(org_img,dsize=(rows,cols),interpolation=cv2.INTER_LIN
EAR)
copy = img.copy()
cv2.rectangle(copy,(rows//2-250,cols//2-90),(rows//2+250,cols//2+11
0),(255,255,255),-1)
final = cv2.addWeighted(img,0.1,copy,0.9,0)
cv2.putText(final,'Most Dominant Colors in the
Image',(rows//2-230,cols//2-40),cv2.FONT_HERSHEY_DUPLEX,0.
8,(0,0,0),1,cv2.LINE_AA)
start = rows//2-220
```

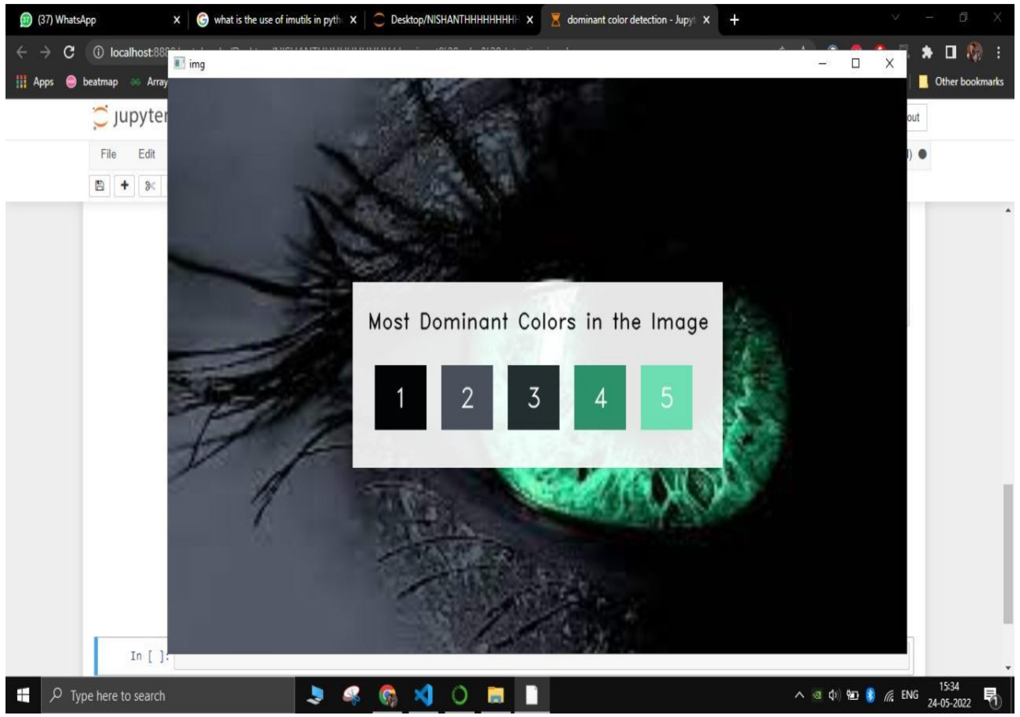
```
for i in range(5):
end = start+70
final[cols//2:cols//2+70,start:end] = p_and_c[i][1]
cv2.putText(final,str(i+1),(start+25,cols//2+45),cv2.FON
T_HERSHEY_DUPLEX,1,(255,255,255),1,cv2.LINE_A
A)
start = end+20
plt.show()
cv2.imshow('img',final)
cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.imwrite('output.png',final)
```

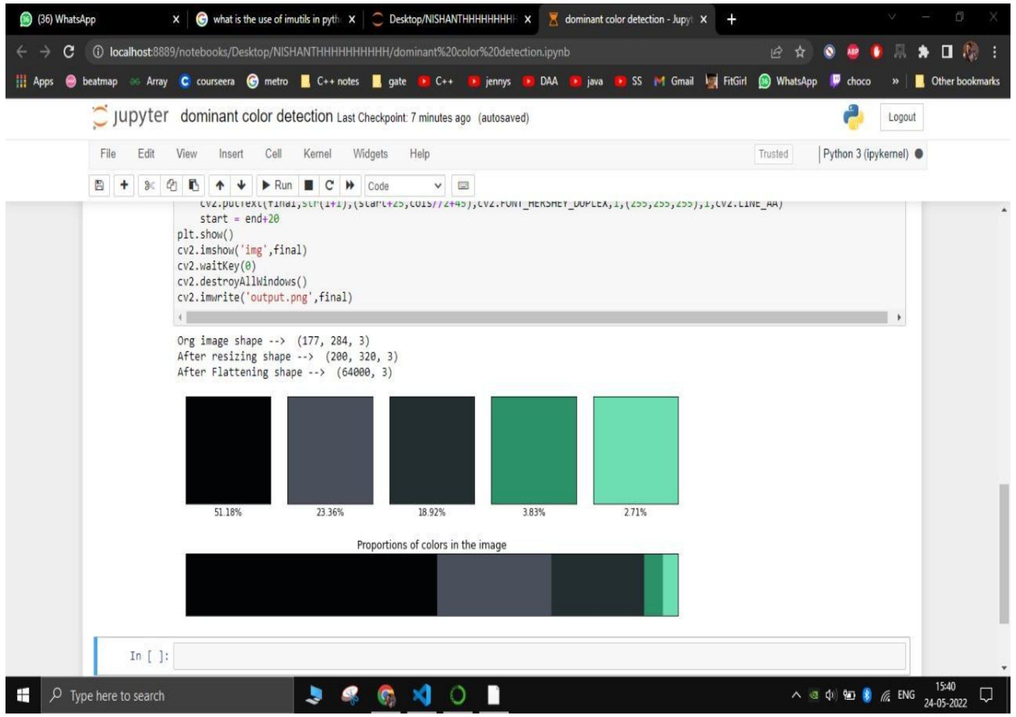
RESULTS

- It can also be observed that the most dominant colors can easily be detected using this application program.
- It displays the most dominant colors starting from one with increasing index.
    1. The greater the index, lesser the dominance of the color.
    2. Lesser the index, greater its dominance.

- This can work on any kind of picture.
- It shows the top 5 dominant color of the picture.

RESULTS




jupyter dominant color detection Last Checkpoint: 7 minutes ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
    cv2.putText(final,str(i+1),(start+25,cols//2+45),cv2.FONT_HERSHEY_DUPLEX,1,(255,255,255),1,cv2.LINE_AA)
    start = end+20
plt.show()
cv2.imshow('img',final)
cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.imwrite('output.png',final)
```

```
Org image shape -->  (177, 284, 3)
After resizing shape -->  (200, 320, 3)
After Flattening shape -->  (64000, 3)
```

51.18%
23.36%
18.92%
3.83%
2.71%
Proportions of colors in the image

```
In [ ]:
```

CONCLUSIONS

- Now a days, for making life much easier dominant color detection can be used. This is a boon in the area of pitures.
- This application for finding most dominant color feature can be used in robot also to perform different tasks.
- In this project, a deep learning model for dominant color detection was developed with the help of K-means clustering algorithm using Python, Computer Vision, OpenCv, Numpy library, matplotlib and imutils
- Dominant Color Detection has found its application such in Image segmentation, image matching, object recognition, visual tracking.

LIMITATIONS

- The system Will detect only top dominant colors.

- Sometimes it can be possible that we need minor color details also. This application lack in that situation.

FUTURE ENHANCEMENTS

- Currently we are doing for one image color detection, we can do dominant color detection of various images at once.

- We can increase the number of dominant color count that we are getting.

- We can do this dominant color detection to identify minor details like traffic signal color in videos.

ESTD : 2001
An Institute with a Difference

REFERENCES

[1] https://www.geeksforgeeks.org/import-module-python/

[2] **https://stackabuse.com/introduction-to-image-processing-in-python-with-opencv/**

[3]**https://www.geeksforgeeks.org/extract-dominant-colors-of-an-image-using-python/#:~:text=Dominant%20colors%20are%20displayed%20using,with%20there%20corresponding%20standard%20deviations**

[4] **https://www.kaggle.com/code/prashant111/k-means-clustering-with-python/notebook**

ESTD : 2001
An Institute with a Difference

Question and Answer

THANK YOU