PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

APRIL 2022: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER

UE20MA251- LINEAR ALGEBRA

Project / Seminar

Session: Jan-May 2022

Branch : Computer Science and Engineering

Semester & Section : Semester IV Section B

Sl No.	Name of the Student	SRN	Marks Allotted (Out of 10)
1.	Anushka Jalori	PES1UG20CS071	
2.	Anushka Pandey	PES1UG20CS072	
3.	Arushi Pandey	PES1UG20CS077	
4.	Athira Prajod	PES1UG20CS089	

Name of the Course Instructor :**Prof. Chaitra GP**

Signature of the Course Instructor

(with Date) : _____

INTRODUCTION

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it.

Fundamental Image Processing Steps -

1) Image Acquisition
2) Image Enhancement
3) Image Restoration
4) Colour Image Processing
5) Compression
6) Representation
7) Recognition

In this project, we have implemented some of the above steps using a few methods, and included some applications of Image Processing.

APPLICATION OF LINEAR ALGEBRA TECHNIQUES AND THEORETICAL BACKGROUND

1) Image reconstruction using SVD

Image reconstruction is the creation of a two- or three-dimensional image from scattered or incomplete data. It is widely used in many fields like medical, and investigation.

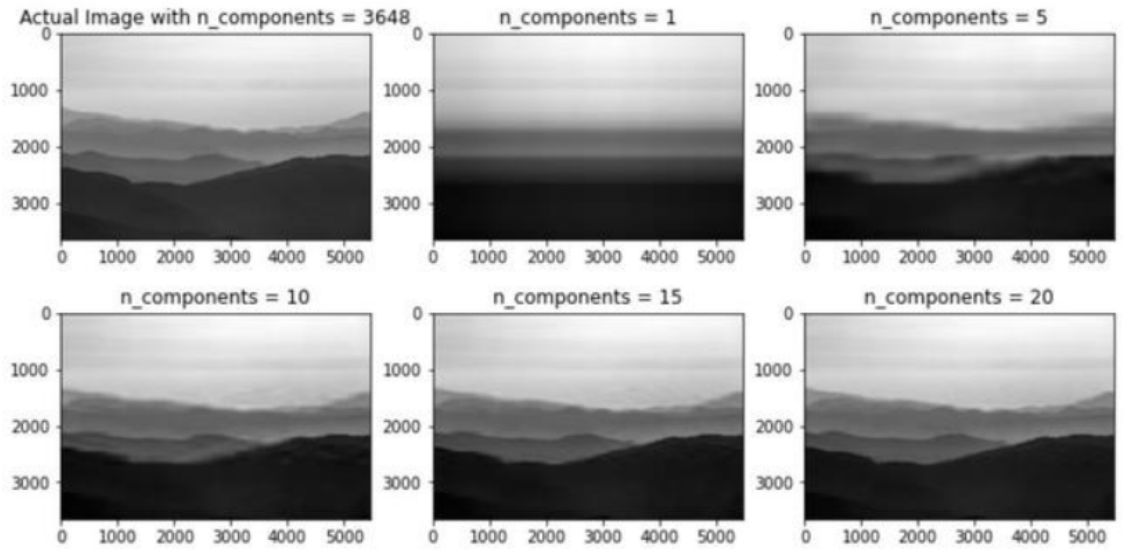
Singular Value Decomposition aka SVD is one of the many matrix decomposition Technique that decomposes a matrix into 3 sub-matrices namely U, S, V where U is the left eigenvector, S is a diagonal matrix of singular values and V is called the right eigenvector.

Code :

```
comps = [3648, 1, 5, 10, 15, 50]

plt.figure(figsize=(12, 6))



for i in range(len(comps)):

      low_rank = u[:, :comps[i]] @ np.diag(s[:comps[i]]) @
v[:comps[i], :]



      if(i == 0):

      plt.subplot(2, 3, i+1),

      plt.imshow(low_rank, cmap='gray'),

      plt.title(f'Actual Image with n_components = {comps[i]}')
```

```
        else:

        plt.subplot(2, 3, i+1),

        plt.imshow(low_rank, cmap='gray'),

        plt.title(f'n_components = {comps[i]}')
```

Output -



Here, we see that the original image had 3648 components, but we were able to recreate an approximation of the original image (which looks like it) using just 20 components.

2) Image enhancement using Histogram Equalisation :

Image enhancement is the technique of improving the quality and information content of an image. One of the ways is increasing the contrast in an image so that the different parts of it are distinct. It is widely used in the medical field to produce a distinct image of lungs, heart and other vital organs very clearly.

Technique used is : Histogram Equalisation

A histogram is plotted against the intensity (X axis) and pixel count (Y axis) . In a low contrast image most of the pixel values will be concentrated in a particular intensity interval , whereas in enhanced images, the pixels will be distributed across all the intensity values.

We have used 2 types of histogram equalisation : Normal, Contrast limited adaptive histogram equalisation (CLAHE).

Code :

Opencv and matplotlib libraries are used to read the image, plot the histogram , change the colour to grayscale.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

Libraries imported

```
gray_img_eqhist=cv2.equalizeHist(gray_img)


hist=cv2.calcHist(gray_img_eqhist,[0],None,[256],[0,256])

plt.plot(hist)

#plt.imshow(gray_img_eqhist)
plt.show()
```

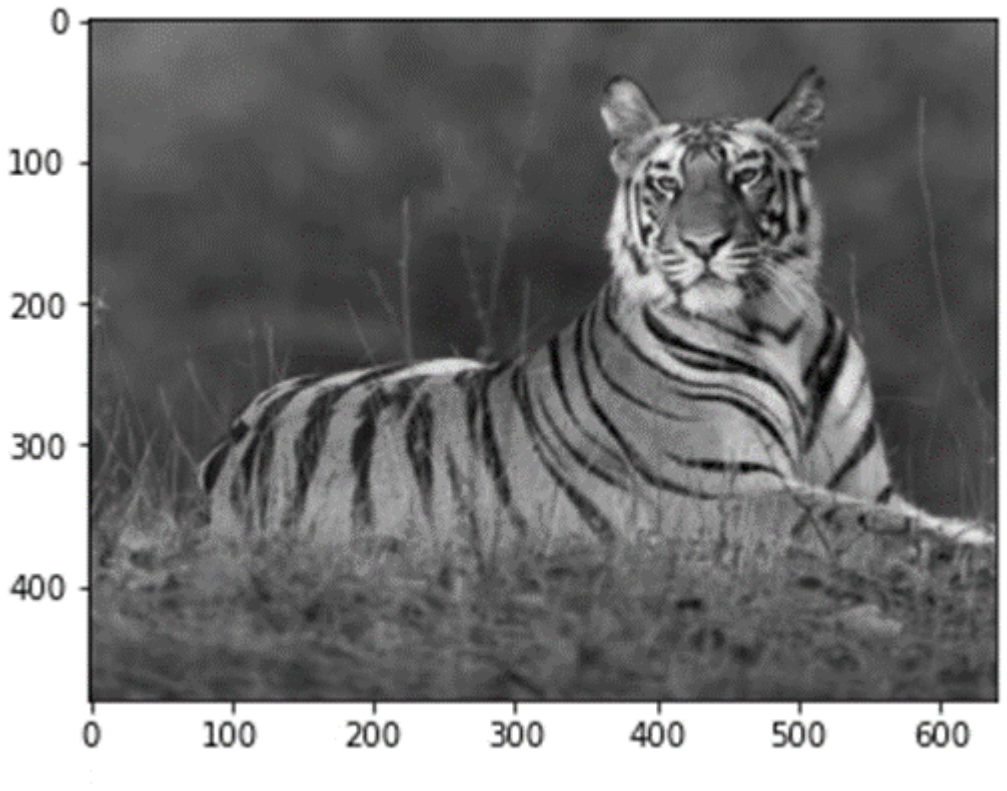
Histogram Equalisation

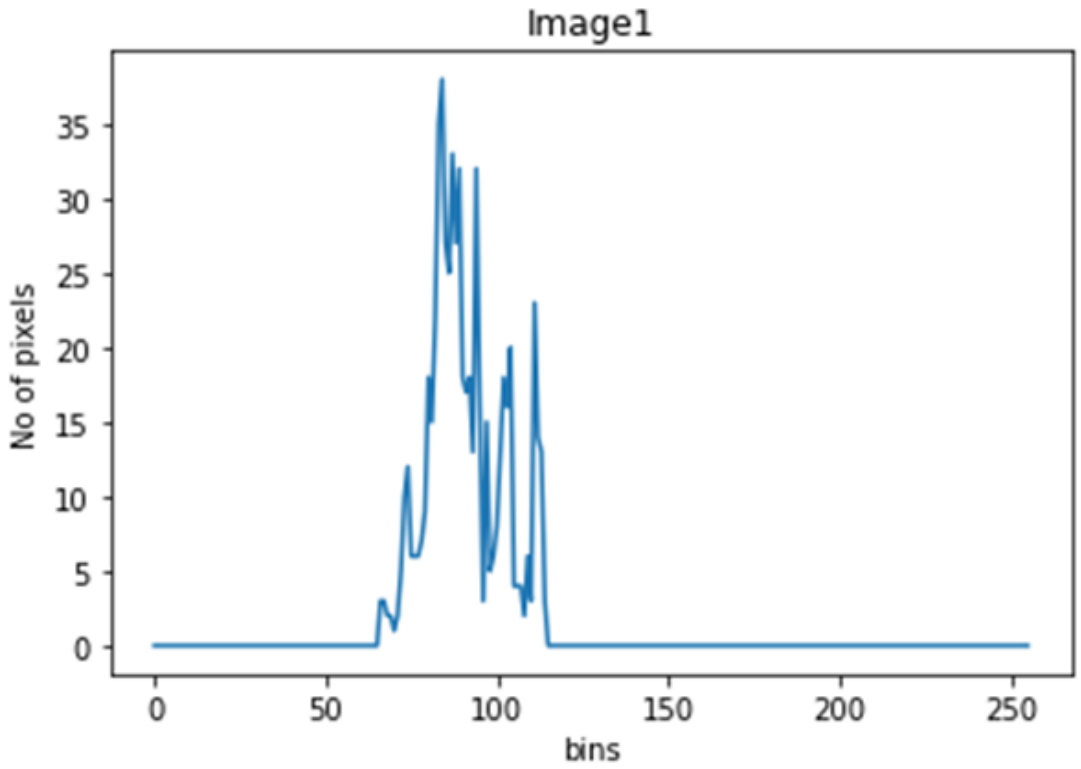
```
clahe = cv2.createCLAHE(clipLimit = 40)
final_img = clahe.apply(gray_img)

plt.imshow(final_img, cmap='gray')
```

CLAHE

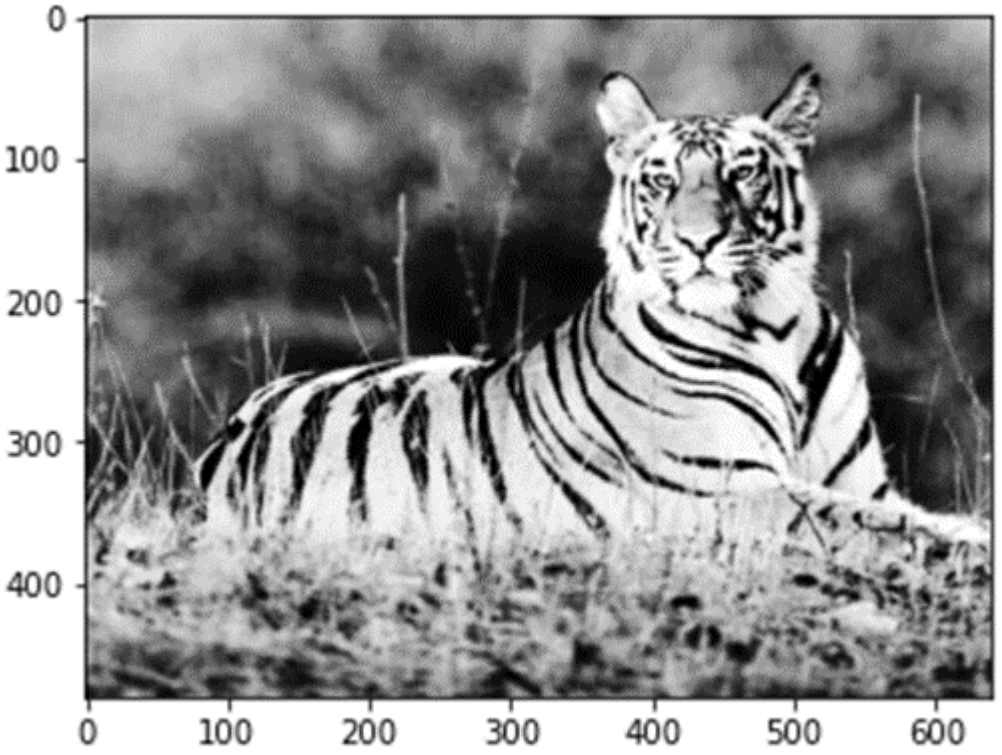
Output Screenshot –

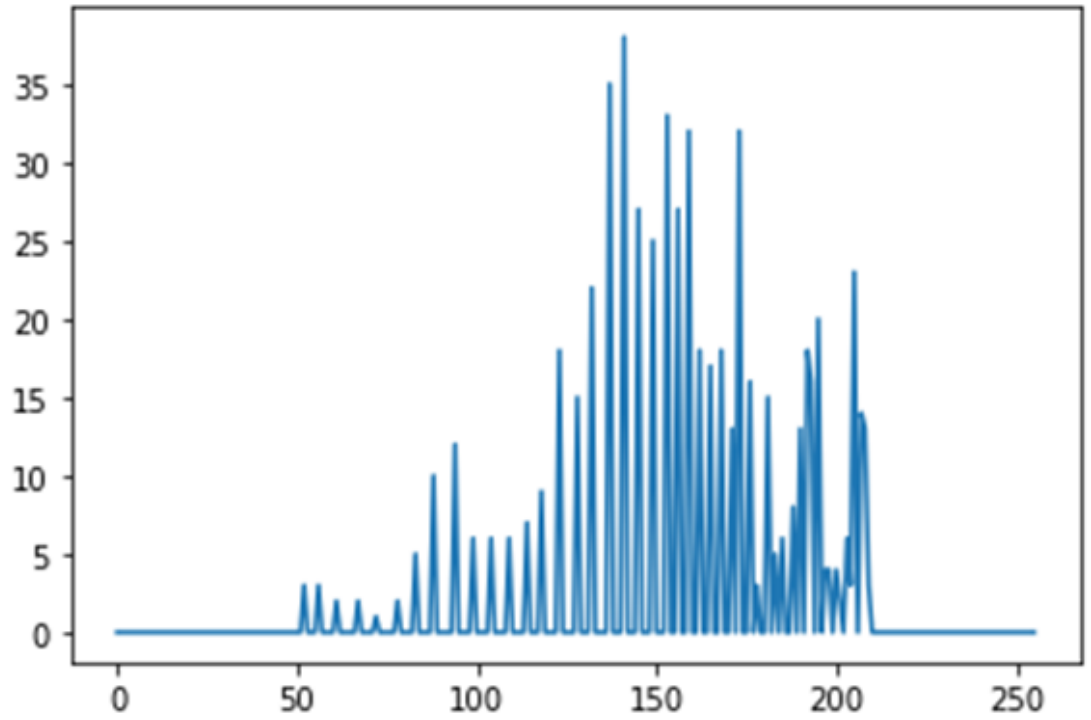
Original image :



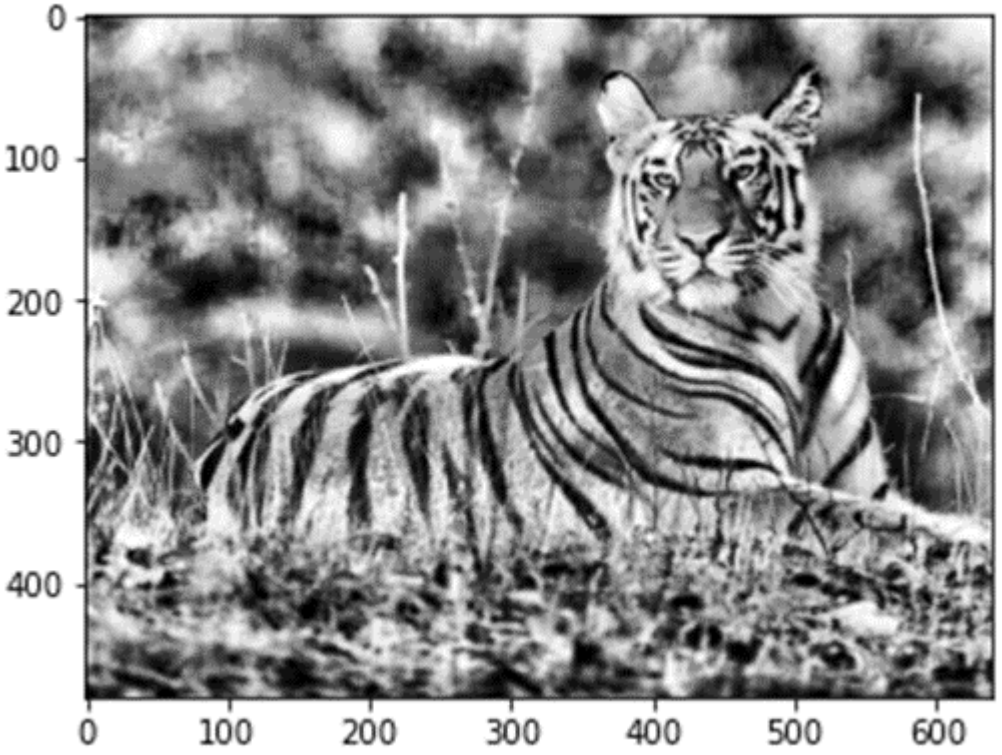


Histogram of image after histogram equalisation :

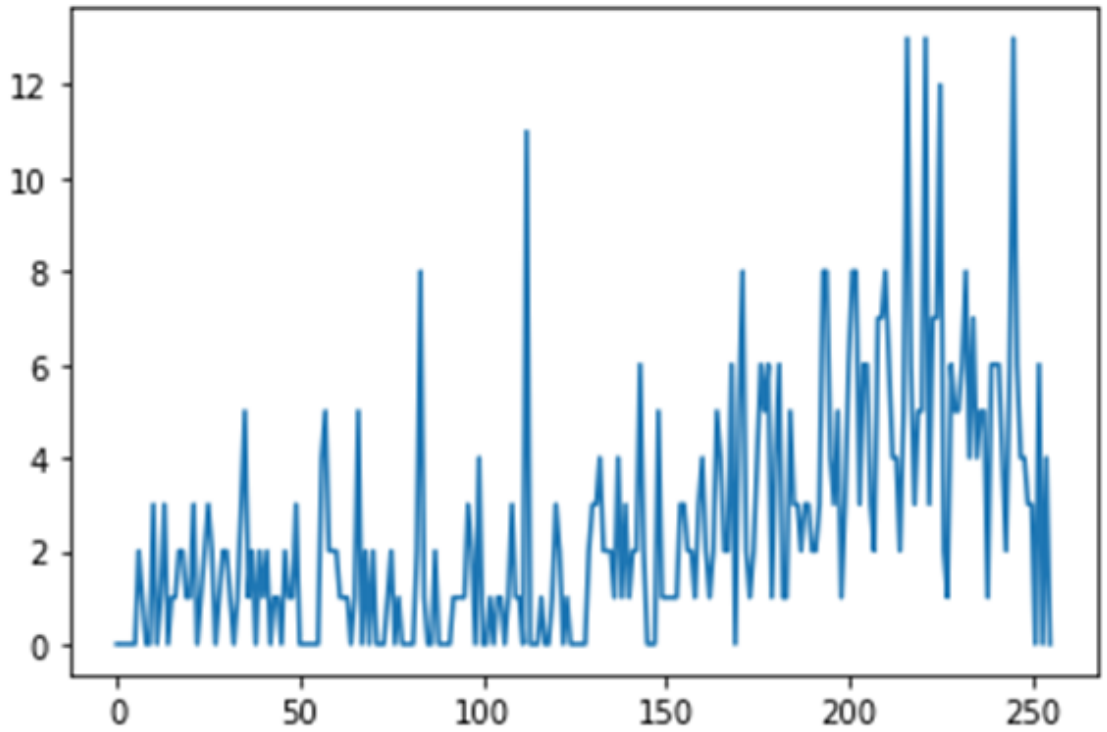




Histogram of image after CLAHE :



```
[<matplotlib.lines.Line2D at 0x2685e95c040>]
```



3)Colour detection of images

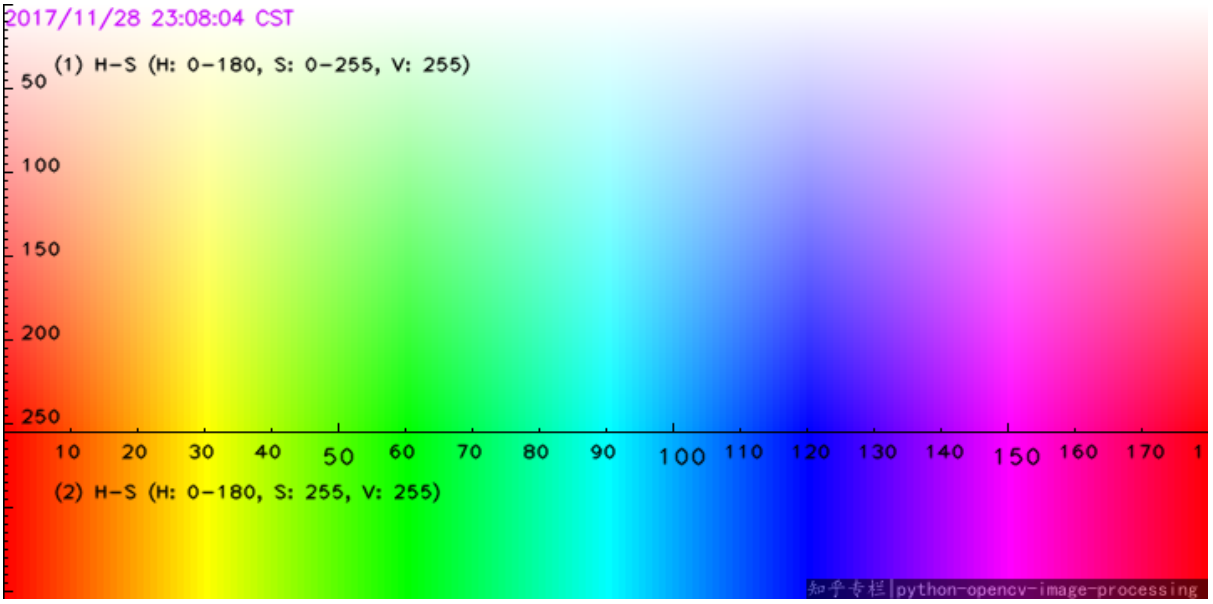
The code detects green colour in the image.
It masks red and blue colour

You can think of the colour masking operation as if looking at the colored canvas through some tinted glass or colour filter. So, by masking off the blue and red channels, you are only allowing the green component of pixels to be updated, and therefore it is as if you were looking through a red tinted glass.

In the below image

X axis stands for Hue
Y1 axis stands for Saturation
Y2 axis stands for Value

We can locate the point according to the HSV form of coordinates from some given RGB coordinates following which an appropriate range can be selected.Using this range we can detect an area of colour in our image.



Finding HSV ranges

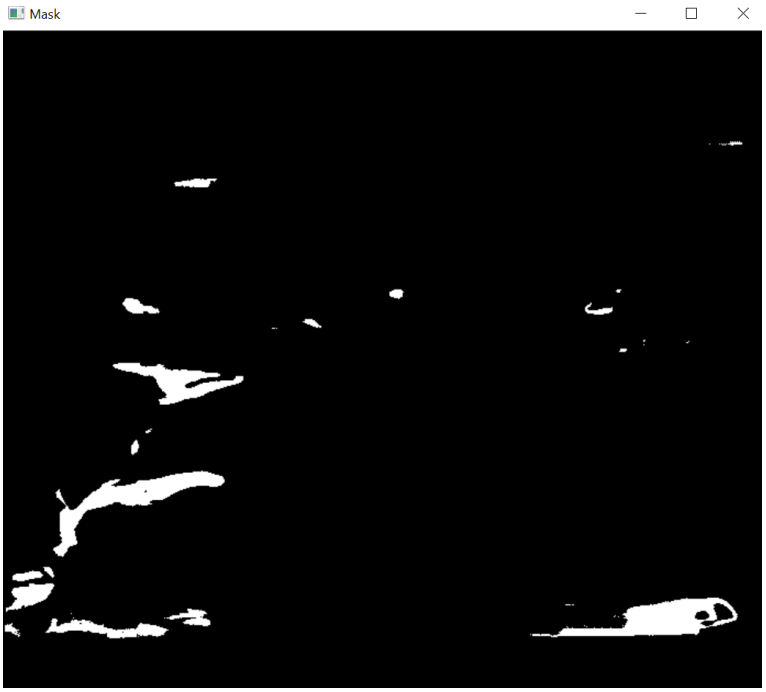
```
# Convert Image to Image HSV
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

green = np.uint8([[[0, 255, 0]]])

# Convert Green color to Green HSV
hsv_green = cv2.cvtColor(green, cv2.COLOR_BGR2HSV)

# Print HSV Value for Green color
print(hsv_green)
```

Masking and showing the image

```
# Defining lower and upper bound HSV values for green color
lower = np.array([30, 100, 20])
upper = np.array([80, 255, 255])

# Defining mask for detecting color
mask = cv2.inRange(hsv, lower, upper)

# Display Image and Mask
cv2.imshow("Image", image)
cv2.imshow("Mask", mask)
```

Original Image

Output Screenshot

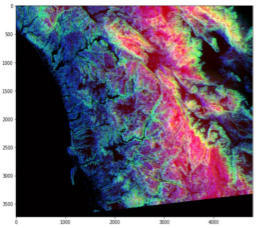


4)Colour Identification in Satellite Images

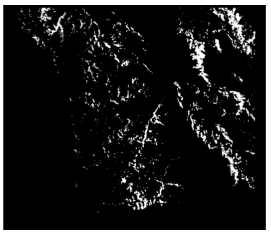
Masking of specific colours to detect the regions of vegetation, ice cloud, desert, ocean etc. in various satellite images.

Output Screenshot:

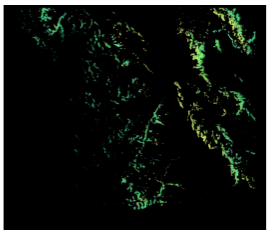
Original Image



Mask Green

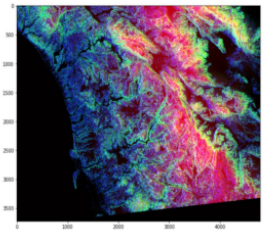


Result

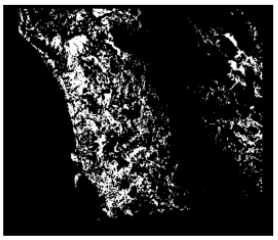


Green colour detects vegetation in the area

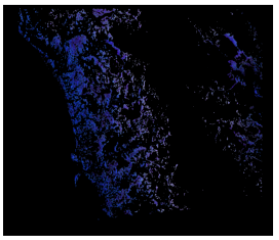
Original Image



Mask Blue

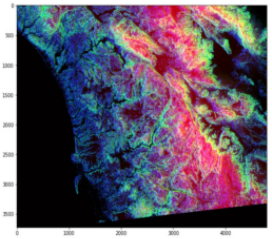


Result

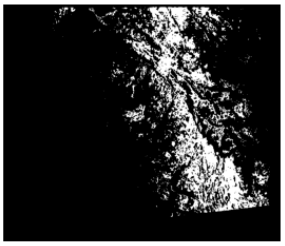


Blue colour detects the ocean

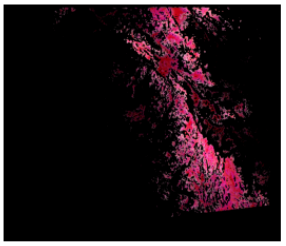
Original Image



Mask Red

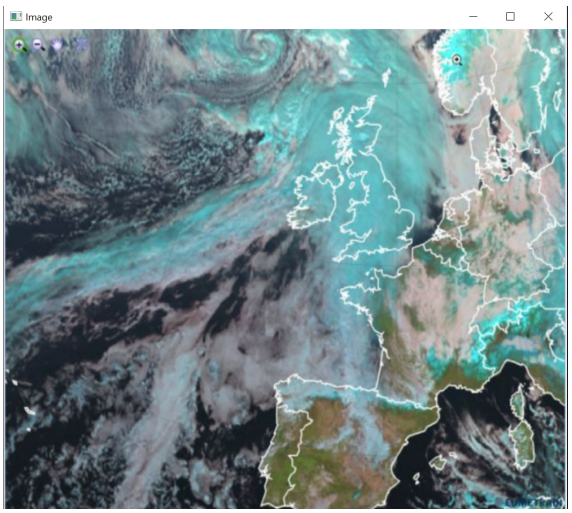


Result



Red color detects dust cloud/desert.

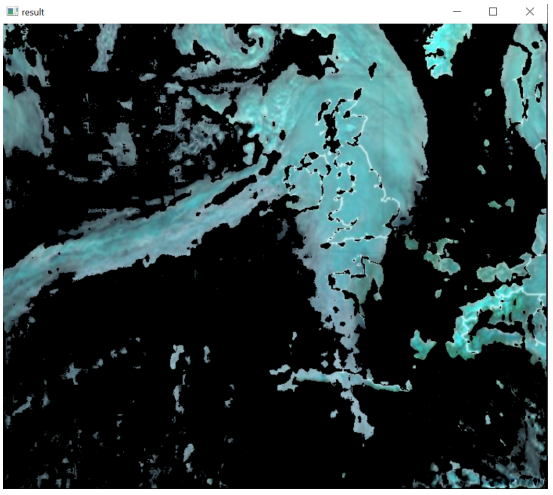
2)**Original Image**



Masked Image

Result

Ice clouds, snow detected in these regions.



Explanation of the result :

1) In the graph of the original image, most of the pixel values are concentrated in a particular intensity interval.
2) After histogram Equalisation, the pixel values are distributed across a comparatively larger intensity interval, thus enhancing the contrast.
3) After CLAHE, the pixel values are even more distributed and thus further increasing the contrast.

1)Region detection in Satellite images-

Masking specific colors yields to detection of different landforms, clouds, snow etc.

Future scope :

1)This technique should be extensively used in the medical field, to emphasise the wanted features and obscure the others.

2) Historical maps classification has become an important application in today's scenario of ever changing land boundaries. Historical map changes include the change in boundaries of cities/states, vegetation regions, water bodies and so forth. Change detection in these regions are mainly carried out via satellite images. Hence, an extensive knowledge on satellite image processing is necessary for historical map classification applications.

References :

1)https://towardsdatascience.com/histogram-equalization-5d1013626e64#:~:text=Histogram%20Equalization%20is%20a%20computer,intensity%20range%20of%20the%20image.

2)https://www.geeksforgeeks.org/histograms-equalization-opencv/#:~:text=Histogram%20equalization%20is%20a%20method,represented%20by%20close%20contrast%20values.

3)http://www.eumetrain.org/satmanu/Basic/RGB_Channels/print.htm#page_2.0.0

Research paper-

1)https://www.researchgate.net/publication/342322385_Image_Processing_Techniques_for_Analysis_of_Satellite_Images_for_Historical_Maps_Classification-An_Overview