COMPUTERVISION
ASSIGNMENT-1

QUESTION:1

Idea:

In this question of finding difference between two images given, I have used difference between the images, thresholding, dilation and contour finding concept to highlight the mismatch region between two images.

Algorithm:

1.Loading the image and dividing them in to two pictures.

2.Converting the loaded images in to grayscale.

3.Finding the difference between those Gray scaled images because the dissimilar points will get highlighted.

4.I have applied thresholding in order to highlight the bright pixels in the difference image.

5.After thresholding, dilation is carried out in order to expand the highlighted image pixels.

6.Now I have applied CHAIN_APPROX_SIMPLE algorithm to find the contours and drawn a rectangle around those dissimilar spots in the image.

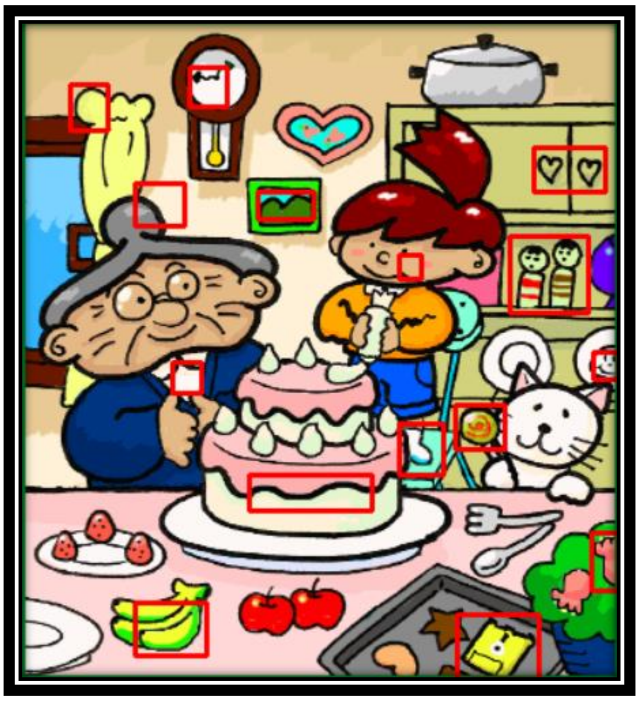
Collab Link:

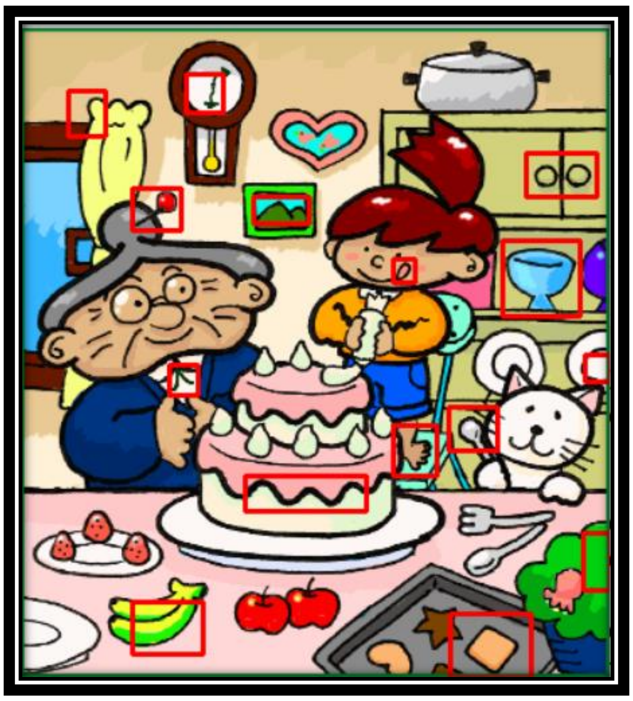
https://colab.research.google.com/drive/1XhysD8buZrnREtx0xmyIHUbl21Pa6JPn?usp=share_link

Limitation:

1.For finding counter area I have tuned the threshold value, For any other similar images threshold needs to be set again.

Result:





QUESTION:2

Idea:

In this question of finding the pixel distance between two states, I have used Euclidean distance and mouse cursor concept in order to find the distance between the two states.

Algorithm:

1.At first, I have read the India map and stored it in a variable.

2.Then I have found threshold image and then I have found text using pytesseract. image_to_data function.

3.Then I have found coordinates of the required state name and drawn rectangle around the text and found distance between the centres of rectangles (here between two states)

4.Then I have drawn a line between two states and showed the output image .

Limitations:

1.The limitation of this approach is I am unable to extract all the text from the given India map.

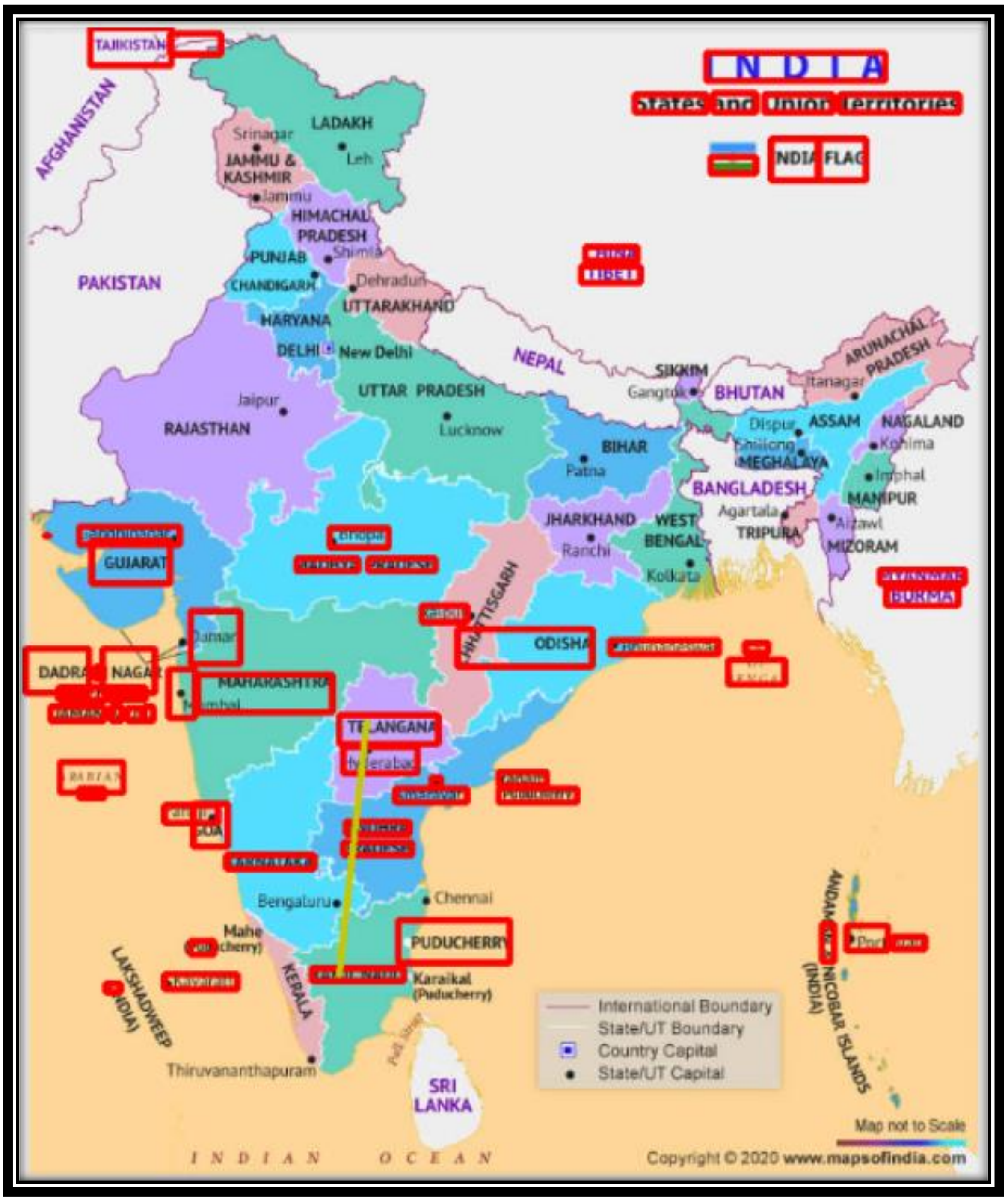
2.As some of the words on the map are too small pytesseract is assigning them random values.

Collab Link:

https://colab.research.google.com/drive/1TbAwAIOfId0Lzytc9muSz__OIUzNDcyr?usp=share_link

Result:

```
Distance between TAMILNADU and TELANGANA is: 184.0896520720271
```


INDIA
States and Union Territories
INDIA FLAG
TAJIKISTAN
AFGHANISTAN
PAKISTAN
LADAKH
Leh
Srinagar
JAMMU & KASHMIR
Jammu
HIMACHAL PRADESH
Shimla
PUNJAB
CHANDIGARH
Dehradun
UTTARAKHAND
HARYANA
DELHI
New Delhi
NEPAL
SIKKIM
Gangtok
BHUTAN
ARUNACHAL PRADESH
Itanagar
Jaipur
RAJASTHAN
UTTAR PRADESH
Lucknow
Dispur
ASSAM
NAGALAND
Kohima
Shillong
MEGHALAYA
BIHAR
Patna
BANGLADESH
Imphal
MANIPUR
JHARKHAND
Ranchi
WEST BENGAL
Kolkata
Agartala
TRIPURA
Aizawl
MIZORAM
GUJARAT
Bhopal
CHHATTISGARH
ODISHA
Daman
DADRA
NAGAR
MAHARASHTRA
Mumbai
TELANGANA
Hyderabad
GOA
Bengaluru
Chennai
PUDUCHERRY
Mahe
Karaikal (Puducherry)
KERALA
LAKSHADWEEP (INDIA)
Thiruvananthapuram
SRI LANKA
ANDAMAN & NICOBAR ISLANDS (INDIA)
International Boundary
State/UT Boundary
Country Capital
State/UT Capital
Map not to Scale
INDIAN OCEAN
Copyright © 2020 www.mapsofindia.com

QUESTION:3

Idea:

In the given question of finding area and perimeter of the circle,I have used HoughCircles method to calculate the given parameters.

Algorithm:

1.First loading the given circle image in to a variable.

2.Converting the given image in to gray scale image.

3.Then,I have applied median blur filter in order to remove the noise in the gray scale image.

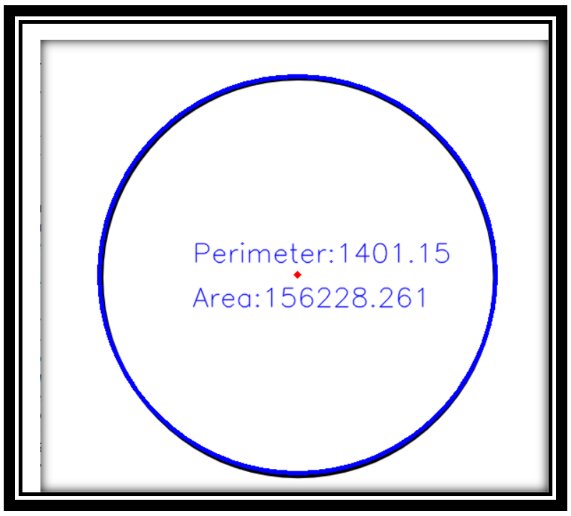
4.After noise removal,HoughCircles method is used to detect the circle in the image and this method returned center and radius of the given circle in the image.

5.Now,Using this radius I have calculated area(Pi*r*r) and perimeter(2*Pi*r) and displayed those values on the image using putText command.

Collab Link:

https://colab.research.google.com/drive/1b-FfFUCUwyiBmK0ROfNeBVfq0MreEqlM?usp=share_link

Result:



QUESTION:4

Idea:

In this question of finding angle between minutes and hour hands I have used Hough lines method to find the minute and hour hand and found angle between them.

Algorithm:

1.First loading the given image in to a variable.

2.Converted the given image in to gray scale image.

3.Then,I have applied canny edge detector in order to find the edges in the image.

4.After this,HoughLinesP method is used to detect the lines in the image by setting some threshold that gives us only hours and minutes hand.

5.Now,Using simple mathematical slope formula , I have found the angle between the lines given.

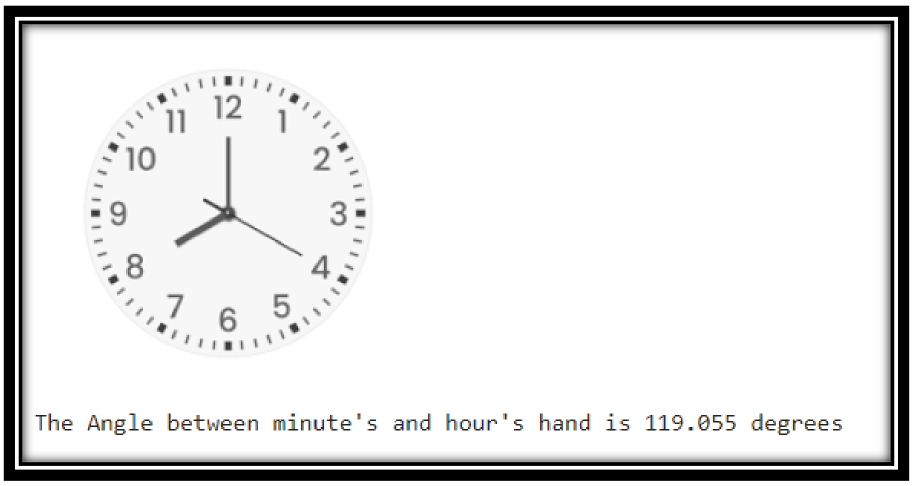
Collab Link:

https://colab.research.google.com/drive/1XQS6IXmZLVwU3ALuo85ES9gsC-KJOs2F?usp=share_link

Limitations:

For any other different clock threshold values needs to be adjusted according to the image.

Results:



```
The Angle between minute's and hour's hand is 119.055 degrees
```

QUESTION:5

Idea:

In this I have chosen Vectorial Memorial, Kolkata images for my implementation.

Here I have performed various image operations i.e., addition of noise in the image, subtraction, averaging of images, noise removal and convolution.

a) Resize all images to 256 × 256. Convert it to Gray**.**

Algorithm:

1. First, I have read images in to variables.
2. They have been resized to 256 x 256 size.
3. Then the RGB image is converted to Gray scale and showed the results.

Collab Link:

https://colab.research.google.com/drive/1IobMJ0eFrHFaJ1GYDgnNvVusTIoZKZjh?usp=share_link

Results:

Image1 and its Gray scale:

Image2 and its Gray scale:

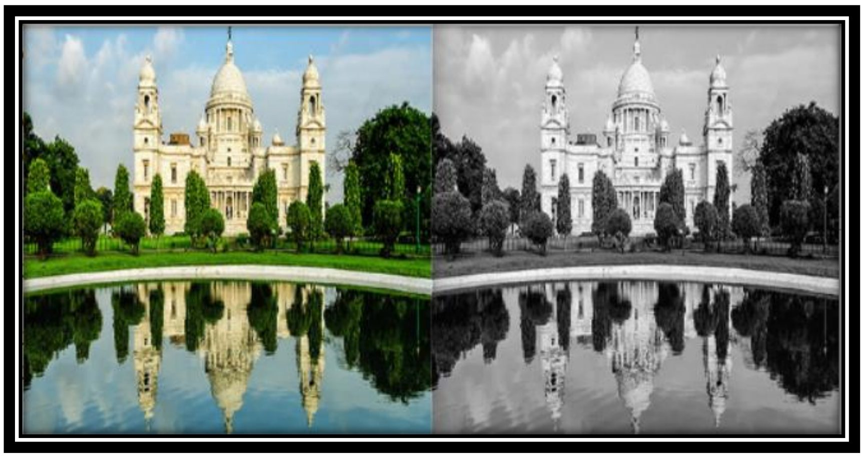
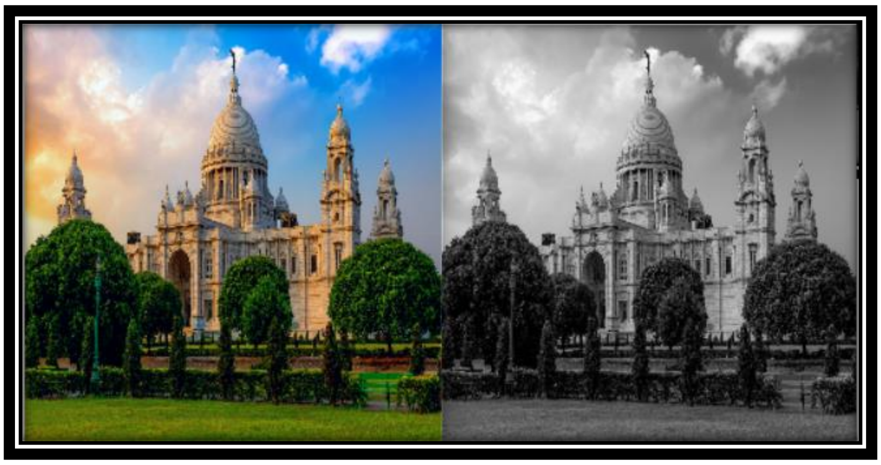


Image3 and its Gray scale:



b) Show the average of all three images.

Algorithm:

1. First, I have read images in to variables

2. Then I have resized all the images in to 256x 256 size.

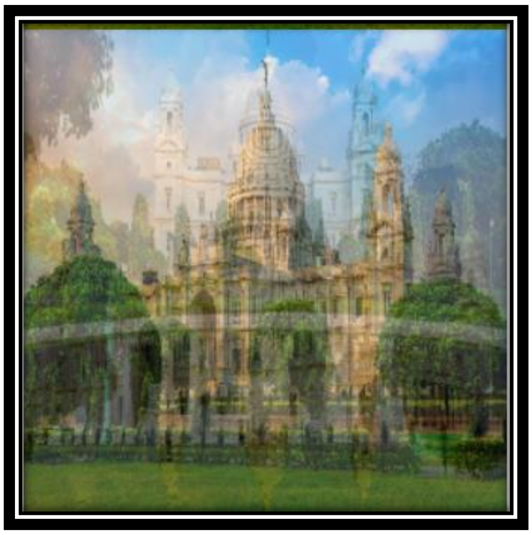
3. Then I have collected three images in to an array.

4. Then I have averaged all the images with equal percentage and stored the result in a variable(avg_image).

5.Finally I have displayed the results.

Results:

Average Image:

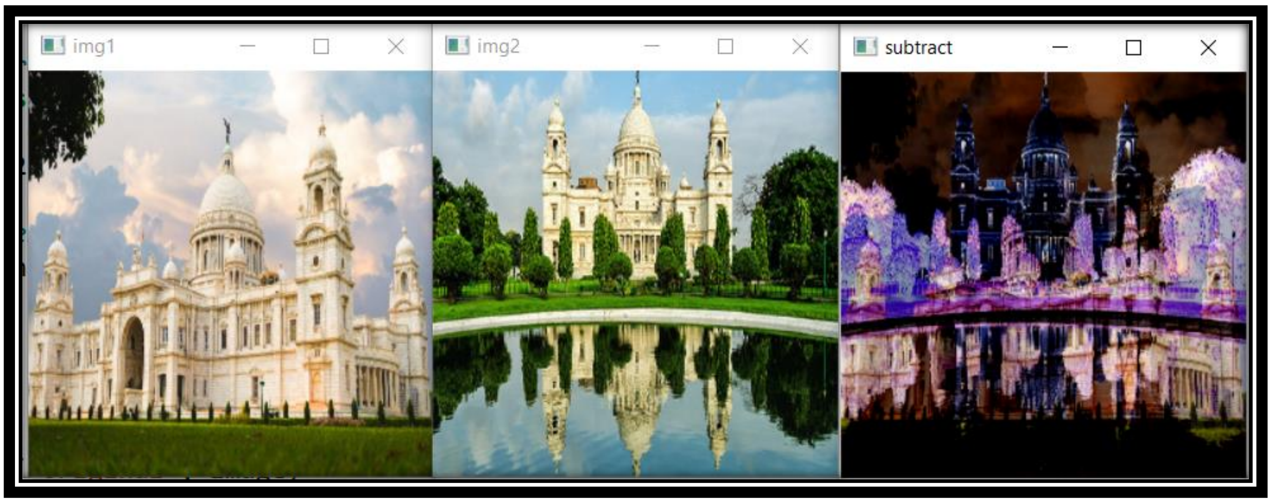


c) Subtract Image 1 with Image 2.

Algorithm:

1. First, I have read images in to variables.
2. They have been resized to 256 x 256 size.
3. Then, I have subtracted image 1 with image 2 and showed the results.

Results:



d) Add salt noise with 5% probability in one of the images.

Algorithm:

1.First, I have loaded image in to a variable.

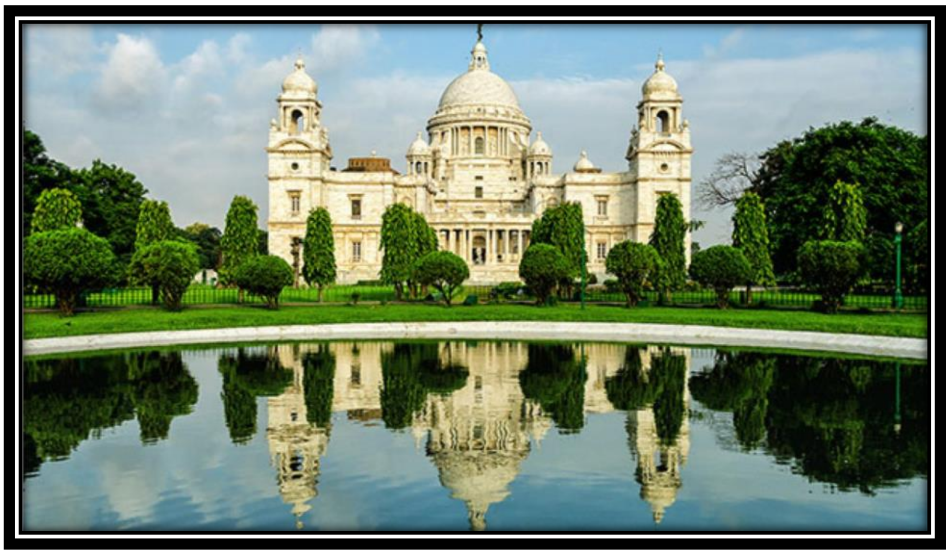
2.Then I have defined a function that takes image and probability as parameter and produces salt and pepper noise image.

3.In this function Based on the probability, I have divided in to salt pixel or pepper pixels (i.e., if pixel value less than probability they are assigned with value zero and vice-versa.).

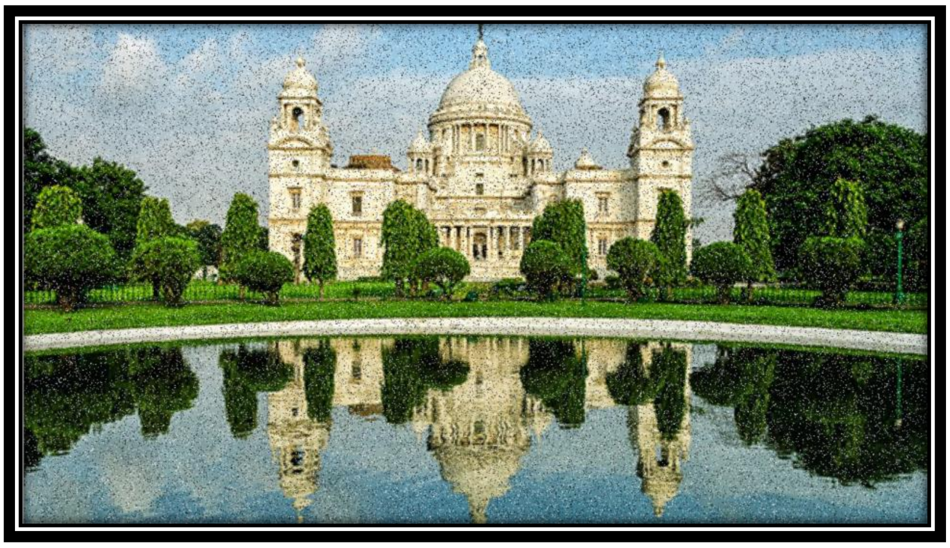
4.Then the final output has been displayed as shown below.

Results:

Original Image:



Noise Image:



e) Remove the noise.

Algorithm:

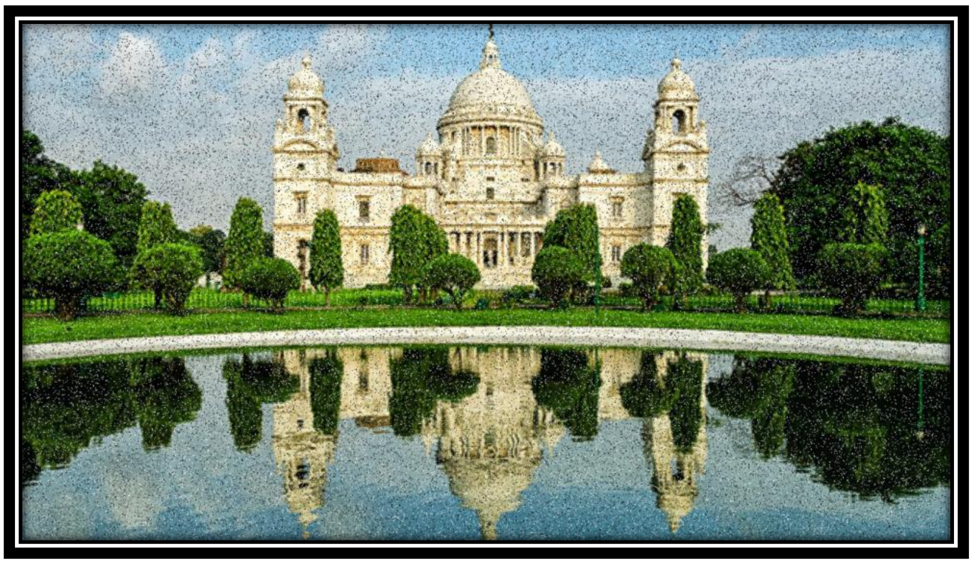
1.The salt and pepper added in the previous question is taken as input.

2.Then median filter is used which averages the pixels under the kernel to produce a single pixel value. This removes the noise to produce original image.

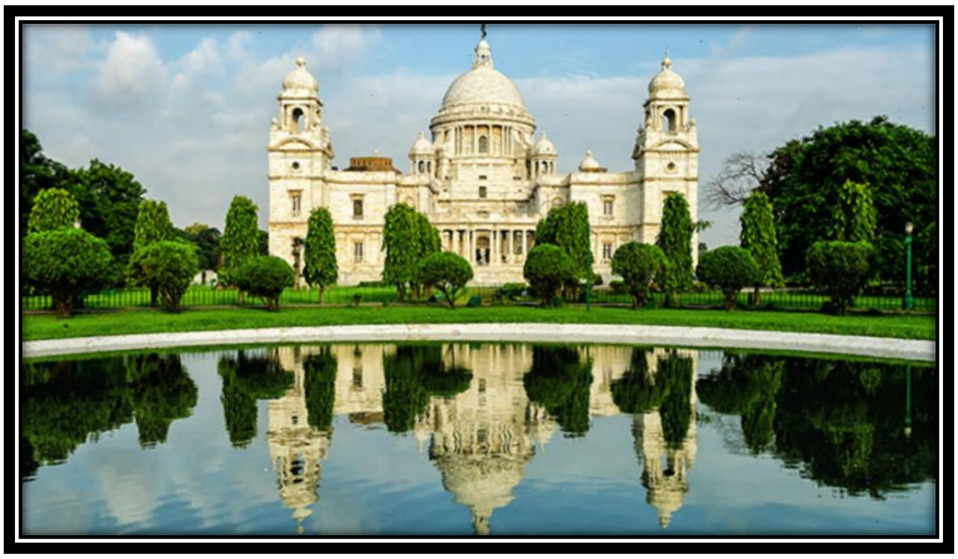
3.Then the final output has been displayed as shown below.

Results:

Noise Image:



Filtered Image:



f) Use the following 3×3 kernel: {−1, −1, −1; 0, 0, 0; 1, 1, 1} for performing convolution in one of the images and show the output.

Algorithm:

1.First, I have loaded image in to a variable.

2.Then I have defined a kernel given in the question.

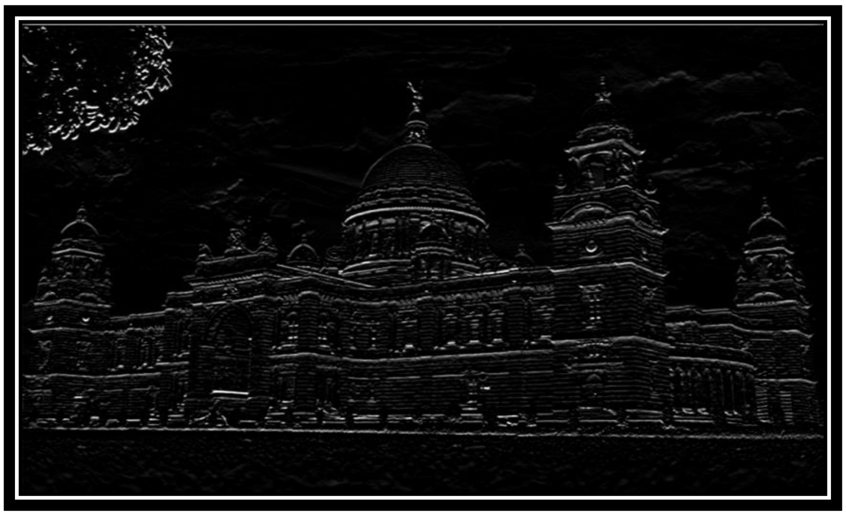
3.By using "cv2.filter2D()" command I have convolved the image with the given kernel.

4.Then the final output has been displayed as shown below.

Results:

Original Image:

Convolved Image:



6) (Digit Recognition) You will be given 100 handwritten images of 0 and 1. You have to compute horizontal projection profile features and use Nearest Neighbour and SVM classifiers to recognize the digits. Report accuracy and show some visual examples.

Idea:

I have tried to implement the code but was not able to implement whole code in python.

Algorithm (I Have thought of):

1. Reading data given and splitting them in to training and testing data.
2. Then training SVM and Nearest Neighbour to recognize the test digits.
3. Then showing the accuracy and precision of different models.

Collab Link:

https://colab.research.google.com/drive/1u4eBcdewVMsOq79ItTJok_Vbyf6sIMNK?usp=share_link

7) White on Black or Black on White:

Idea:

Here in this question of finding background colour I have gradient method and average of most frequently occurring colours in the given image.

Algorithm:

1.First, I have loaded an image in to a variable.

2.I have found what colour occurred the most in the most in the given image. I splatted the image into its three colour channels and then converting them in to a 1-dimensional array.

3.Now I have stored frequency of RGB values separately in different variables.

4. Then I have taken the topmost common colours in the image and then take the average of the 3 colour channels to get an average colour value for the background of the image.

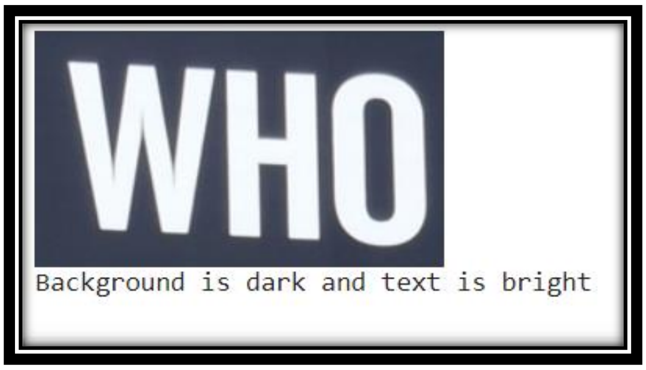
5.Then I defined a function which tells us background colour based on average pixel values.

6. Finally I have displayed image and printed colour of background.

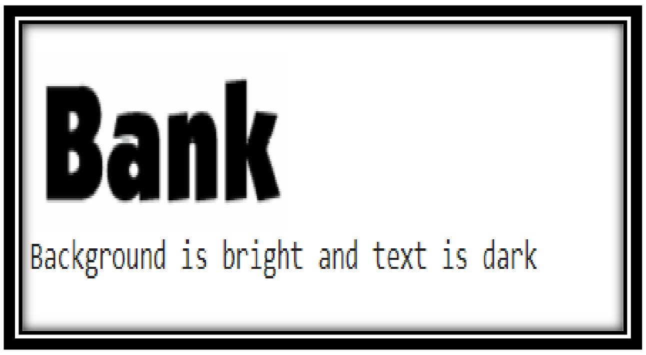
Collab Link:

https://colab.research.google.com/drive/1_5PTVJVVv6ZamkWhtgCZI6uIiaaZjrge?usp=share_link

Results:



```
Background is dark and text is bright
```



Bank

Background is bright and text is dark

8)Template Matching:

Idea:

In this question of template matching, I have written my name and found the one of the letters on the image.

Algorithm:

1.First, I have loaded image and template image in to two variables.

2.Then I have converted my image in to Gray scale.

3.By using "cv2.matchtemplate" command, which simply slides the template image over the input image (as in 2D convolution) and compares the template and patch of input image under the template image. It returns a grayscale image, where each pixel denotes how much does the neighbourhood of that pixel match with template.

3.Then, I have used "cv2.TM_CCORR_NORMED" method, I have found some bright spots on the image (i.e., matching of two images).

4.By setting threshold value, I have filtered the value above the threshold and put a rectangle around it.

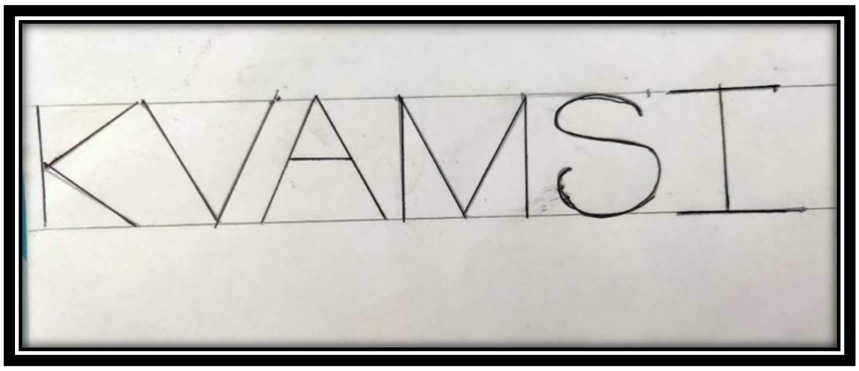
5.Then the final output has been displayed as shown below.

Collab Link:

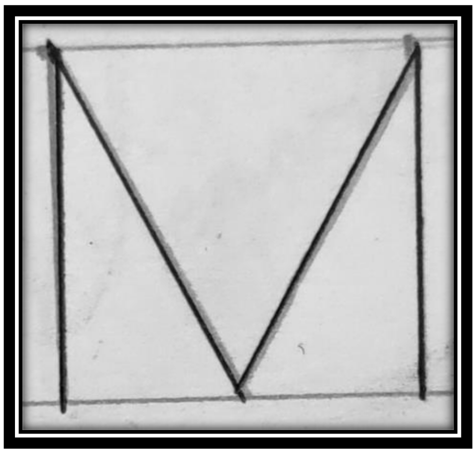
https://colab.research.google.com/drive/1-ja9oIdmV9P-kYmE7wN_iVsqeXv7oLRt?usp=share_link

Results:

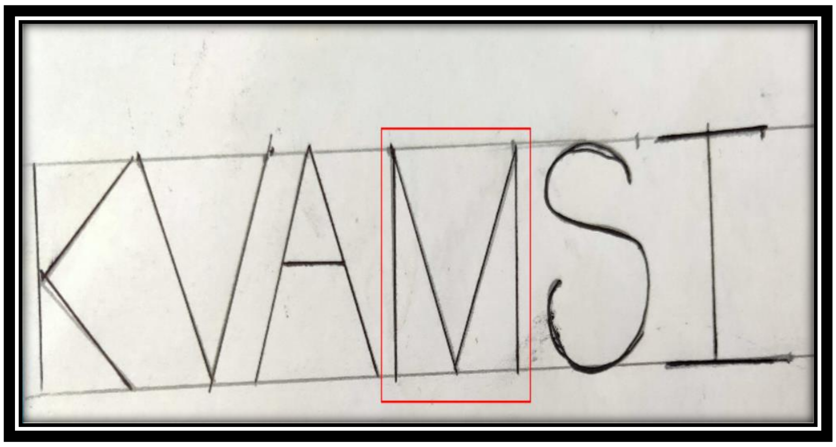
Original image:



Template image:



Output:



9) (Histogram Equalization)

Idea:

In this question of histogram equalization, I have plotted histogram of images with bin size=10 as mentioned in question and image equalisation task is carried out.

Algorithm:

1.First, I have loaded image in to a variable.

2.Then by using "cv2.calcHist" function I have calculated the number of pixels that are having same intensity levels with bun size=10(i.e., whole intensity levels from 0 to 255 were accommodated in to 10 bins).

3.Using plot command I have plotted the histogram.

4.Now after converting original image in to Gray scale image I have applied equalisation technique to it. Here Histogram equalization is done for adjusting image intensities to enhance its contrast.

5.To implement this, I have used "cv2.equalizeHist" method.

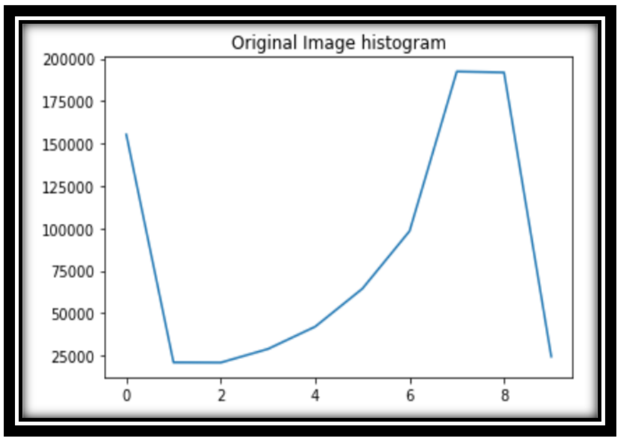
6.Then the final output has been displayed as shown below.

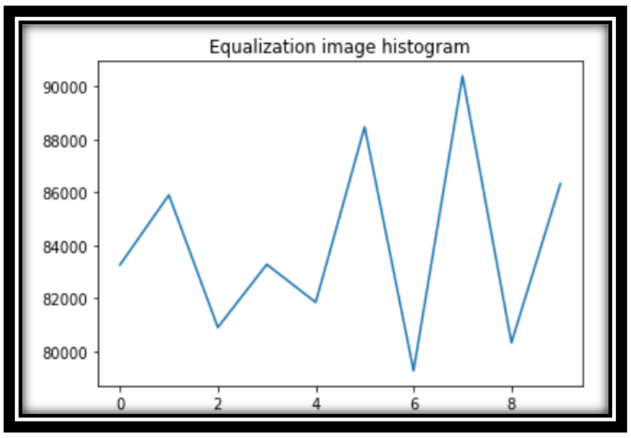
Collab Link:

https://colab.research.google.com/drive/1LXqsx8DDb9RORXaXlM8yy_Oa4stjpkJy?usp=share_link

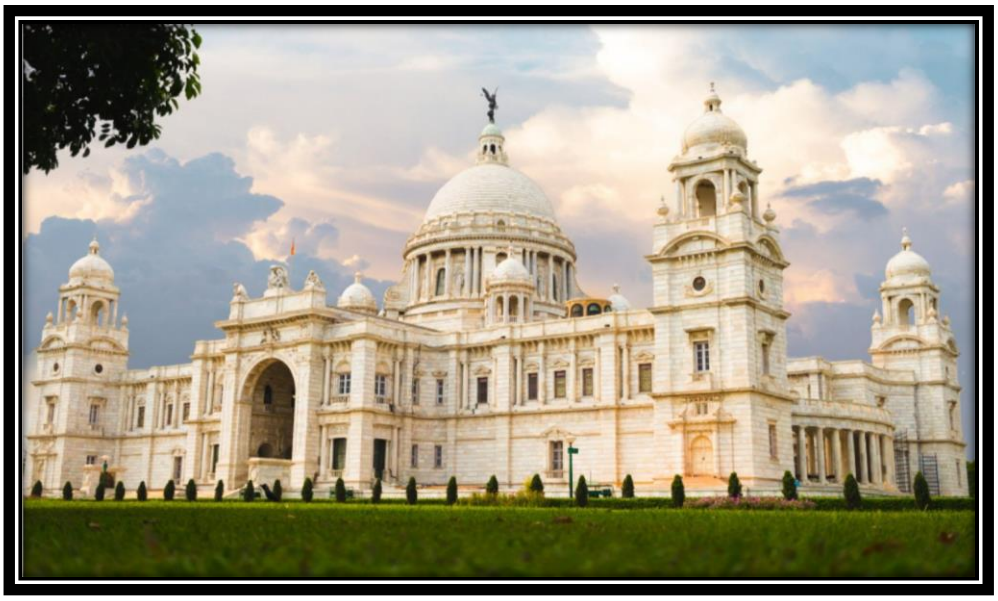
Results:

Histogram:

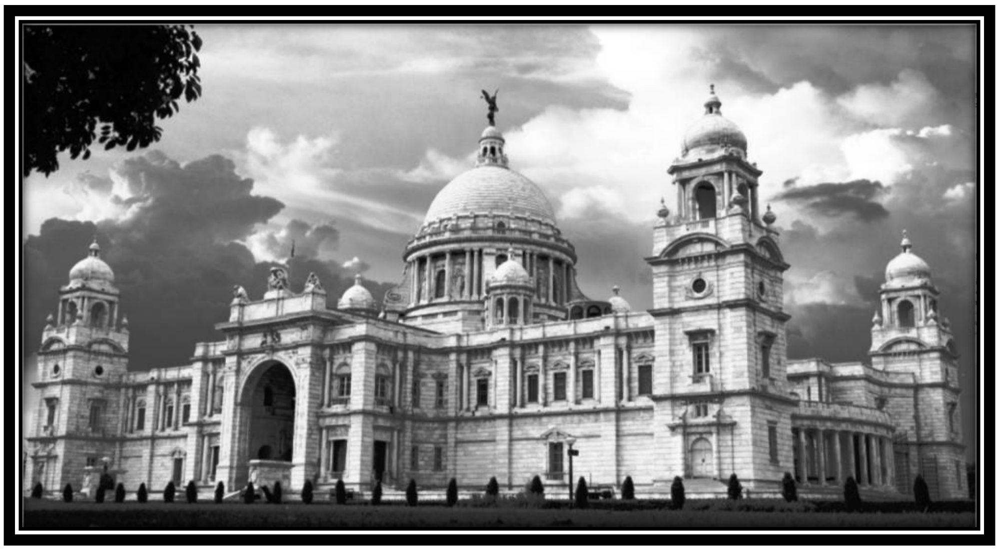




Original Image:



Equalisation output:



10) Reading Mobile Number

In the given question of finding last three digits of given phone number I have used pytesseract library.

Algorithm:

1.First, I have loaded image in to a variable.

2. Then I have defined aa function that returns threshold image of given input image.

3.By using "pytesseract. image_to_data" function image data is converted in to dictionary.

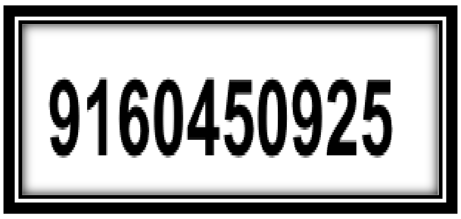
4.By using slicing of array, I have extracted last three digits of the phone number and printed them on the screen.

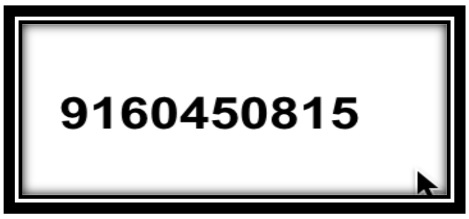
Collab Link:

https://colab.research.google.com/drive/1Nc8cSFodl8gym9d0Z3-dI-LdJnUVoh2k?usp=share_link

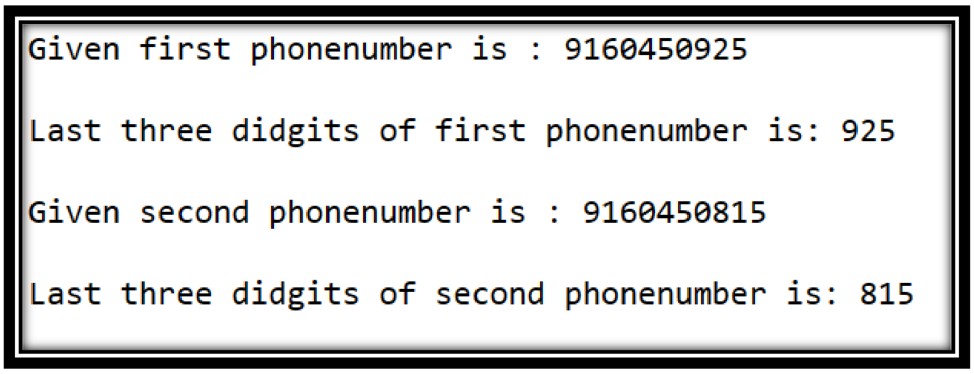
Results:

Original Image:


9160450925


9160450815

Output:



```
Given first phonenumber is : 9160450925

Last three didgits of first phonenumber is: 925

Given second phonenumber is : 9160450815

Last three didgits of second phonenumber is: 815
```

References:

1) https://docs.opencv.org/4.x/
2) https://github.com/askitlouder
3) https://nptel.ac.in/courses/106106224
4) https://numpy.org/doc/
5) https://www.opcito.com/blogs/extracting-text-from-images-with-tesseract-ocr-opencv-and-python
6) https://pyimagesearch.com/2021/05/12/image-gradients-with-opencv-sobel-and-scharr/
7) https://learnopencv.com/contour-detection-using-opencv-python-c/
8) ttps://www.tutorialspoint.com/how-to-detect-a-rectangle-and-square-in-an-image-using-opencv-python