

PROJECT REPORT

Q1.Given a stitched image containing two very similar scenes,Find out the differences. (a) Submit your implementation. (b) Write down your algorithm in brief. (c) Show the image where differences are suitably marked. (d) Write down scenarios when your implementation may not work.

Ans:

a)implemented in google colab

b)Algorithm:-

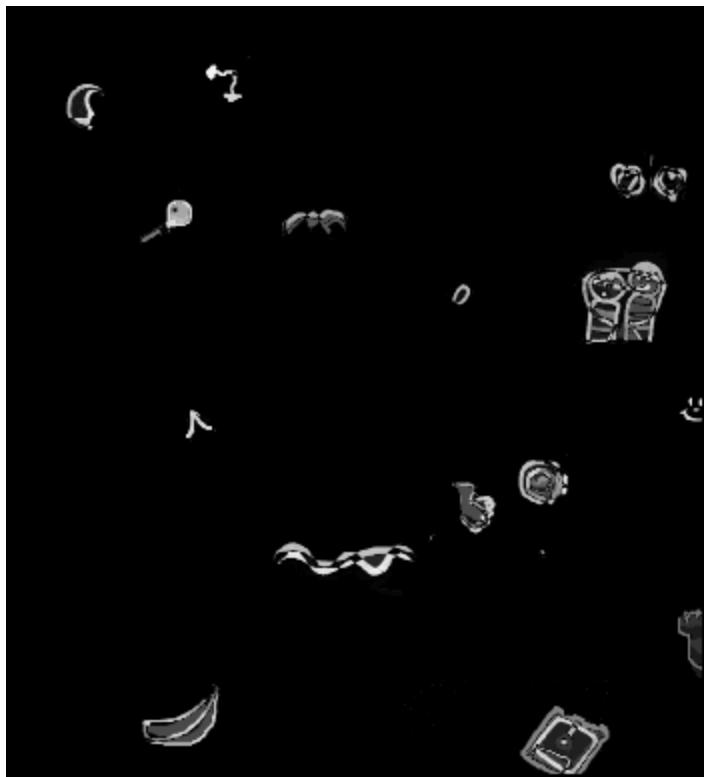
1. Sliced the image into two parts.
2. Read the two sliced images and displayed the two images after splitting..
3. Find the difference between the two slices using ImageChops.
4. Used thresholding to give a clear picture on the differences using cv.THRESH_BINARY
5. Use a kernel to dilate the differences and create a suitable object for the rectangle which will be used to mark the differences in the stitched image.

c)Results:

1)SLICED IMAGE:



Difference in image:



More threshold image of Diff image:



Marked the differences in the image using Rectangle shape;

COMPARISON vs REAL and FAKE



'Org image'

'marked Image'

d)Scenarios when the code may not work is when the stitched image will be of different ratio other than 1:1 ,In that case the image will not work or if both images will have some other aspects and parameters changed ,then in taking difference we cannot get proper results.

Colab link:

https://drive.google.com/file/d/1N-NSM5C2L_cQei54SBU38wLT8hYpWeC_/view?usp=sharing

References:<https://www.geeksforgeeks.org>

Youtube

Q2.Given an image of the map of India, find out the pixel distance between two states.

Ans-

- 1.first of all we install easeocr library for extracting texts from image.
- 2.after it import the required library like numpy, cv, easeocr, matplotlib, PIL, etc and read and shown map of india.
- 3.Next we will set the english language to detect the text from image
- 4.After it run the easeocr module command to extract all the text from the image and append it in the list.
- 5.Easeocr will return the coordinates(pixel values) of the four edges of the texts.
- 6.We have noticed that the states is in capital letters in the given map of india.
7. Using loop we have dropped lowercase character from the list

8. Hence we have transformed the list with only the states and its coordinates values.and in it we have statically chosen two random states of which we have to calculate the distances.
- 9.we have calculated the centroid of the 4 coordinate points using the average value of x and y values for both the states.
10. We have got the 2 centroid points which lies exactly in the middle of the text of states.
11. Finally we have performed Euclidean distance and manhattan distance to both the centroids of the states.

Assumptions:-

1.Assume the centroid of the text of the states as the centre measuring point between two states.

2.we have to use only easeocr library in the code to solve the problem as its return type gives the coordinates of the 4 edges of the text obtain after extraction

Limitations:

1. Texts extracted from the images are not accurate. It is broken , two words' state name is considered as two different states or sometimes it has detected special characters in the form of texts.
2. Though we have taken the text centroid as a reference to measure ,suppose for some states it is written outside then the measuring point will shift outside ,hence will give the inaccurate result.
3. Programme implemented for the above problem is static because of the inaccuracies in the results of lists.

Results:Here we will be directly picking two states and through their respective index we find the centroid of the coordinates of the text and use euclidean and manhattan distance.

#Distance between JAMMU AND HIMACHAL

euclidean distance between two states is 69.63476143421474

manhattan distance between two states is 97.0

Colab link:-

<https://colab.research.google.com/drive/1arkYtUY3w951V4Ejj8p4beACxl1j51tz?usp=sharing>

References: Geeks for geeks implementation of EaseOcr

Q3.Given an image of a circle, find out the area and perimeter in the pixel unit.

Ans:

1. import the required library like numpy, cv, matplotlib, PIL, etc and read and show the image of circle.
2. Convert the image to the array form .
3. Visualise the array and get the height and width of the image.
4. Divide the height of the image by two and get the centre of the height and obtain the array of that rows.
5. We have seen that due to the black line the value becomes 0 at the circumference and at the other part its 0 as its complete white.
6. Considering that row passing through the centre of the circle we have subtracted the pexel value of two extremes point two get the diameter.
7. After obtaining the diameter , we have find the area and perimeter of the circle by using $\pi * (d/2)^2$ and $\pi * d$ respectively.

Assumptions:

1. Consider that the Circle is placed at the center of the given image as we have just taken midpoint value of the height and then proceeded further assuming that the center is passing through that rows . if it would not be in center then that particular row we cannot be able to find the diameter.
2. We have taken the average of the black pexels as the middle value of 3 black pexel considering the pexel value

Limitations:

1. When we place the circle at the center of the image then only it will work.

Results:

DIAMETER OF THE CIRCLE :-429

AREA OF THE CIRCLE:-144472.185

Perimeter of the Circle:-1347.06

References:Geeks for geeks /Youtube

Colab

link:<https://colab.research.google.com/drive/136IQ-Z8ah6RMEI1nifgObBZjTqjda13o?usp=sharing>

Q4. Find the angle between the minute hand and the hour hand of the two clocks given in the figures.

Ans:

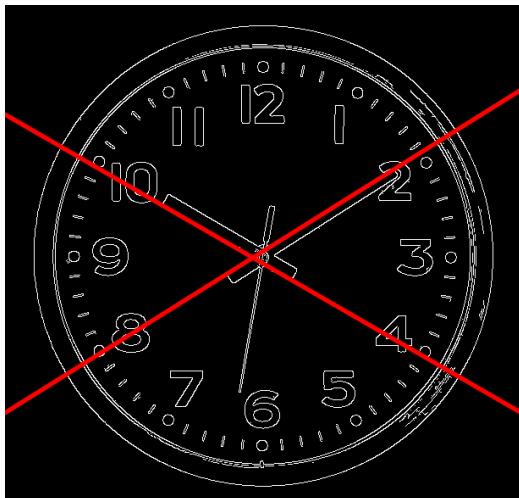
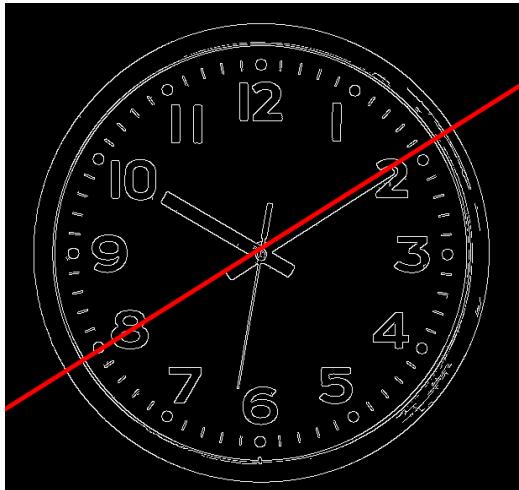
1. import the required library like numpy, cv2, Canny edge, HoughLine, matplotlib, PIL, etc and read and show the image of clocks.
2. Now convert both images to grayscale image.
3. Find the edges in the image using Canny image.
4. After finding edges in the image supply it to the Canny Hough Line transform for detecting lines of the clock with the threshold value of 150 and 50 to the two clocks as uploaded in the colab code.
5. Print the Lines and we have seen that rho and theta value gets printed.
6. After that using lines parameter, we have printed all lines to the required threshold on the clock using red colour and observe that which two lines we are getting passed over minute and hour hand more accurately.
7. Using the list of that image access the theta value of the lines stored in the line parameter and subtract the two theta values to get the angle in radians.
8. Convert the radians to degree.
9. Again subtract the angle from 180 to get the obtuse angle i.e. proper angle between the minute hand and hour hand.

USE ABOVE PROCEDURE FOR BOTH THE CLOCK AND ANGLE WILL BE CALCULATED.

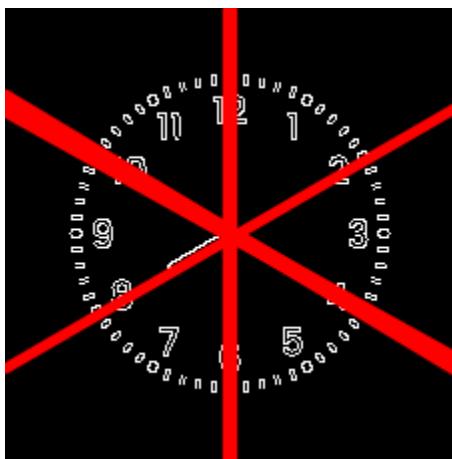
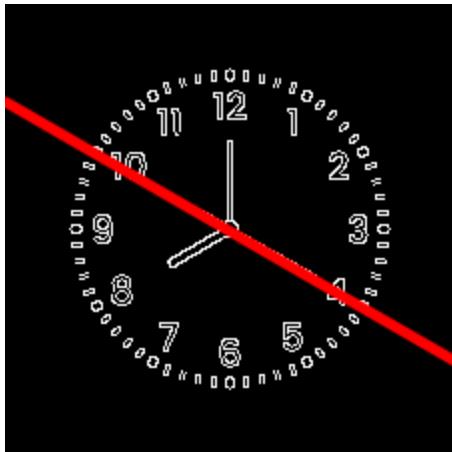
Results: the rho and theta value of both the clocks as detected by the Hough Line Transform

1. [[[407. 1.012291]]]
- 2.
3. [[112. 2.0943952]]]
- 4.
5. [[421. 0.99483764]]]
6. 3
7. [[[42. 2.0943952]]]
- 8.
9. [[111. 0.]]]
- 10.
11. [[45. 2.0769417]]]
- 12.
13. [[113. 0.]]]
- 14.
15. [[43. 2.0769417]]]
- 16.
17. [[40. 2.0943952]]]
- 18.
19. [[37. 2.1118484]]]
- 20.
21. [[39. 2.1118484]]]
- 22.

23. [[155. 1.0471976]]
- 24.
25. [[151. 1.0471976]]
- 26.
27. [[-110. 3.106686]]]



CLOCK-1:The above two lines are the detected lines between which we will find the angle just by subtracting the theta values of the above results.



CLOCK-2: The above two lines are the detected lines between which we will find the angle just by subtracting the theta values of the above results.

Angle between minute and hour hand for clock 2 time 10:10 o clock: 117.99999592711777
Angle between minute and hour hand for clock 1 at time 8:00 o clock : 119.9999982617879

Assumptions:

1. We have taken edges of grayscale and then used Houghline detector to that edges with the fixed threshold i.e 150, 50 in the HoughLine Transform parameters.
2. Angle are taken tentative by looking the lines which is shown in the results and then statically observing the theta values in the Lines parameter.
3. Thickness of the minute and hour hands should be same

Limitations:

1. It is calculated statically, if there will be different image supplied then we will not get using this code, we again have to find the lines over minute and hour hand and calculate the theta values.

References: <https://www.geeksforgeeks.org/line-detection-python-opencv-houghline-method/>

Colab

[link:https://colab.research.google.com/drive/137RoSJy1eCF8LfZwPaZRYeDq9xRhmo_e?usp=sharing](https://colab.research.google.com/drive/137RoSJy1eCF8LfZwPaZRYeDq9xRhmo_e?usp=sharing)

Q5. Choose the three images from the landmark image dataset such that its names starts with your name first letter and perform the following operations:

Chosen images are:

Amsterdam

Akshardham

Auli

1)(a0) Resize all images to 256 × 256.





-Convert the resized image to Gray.





B) average of all the three images:



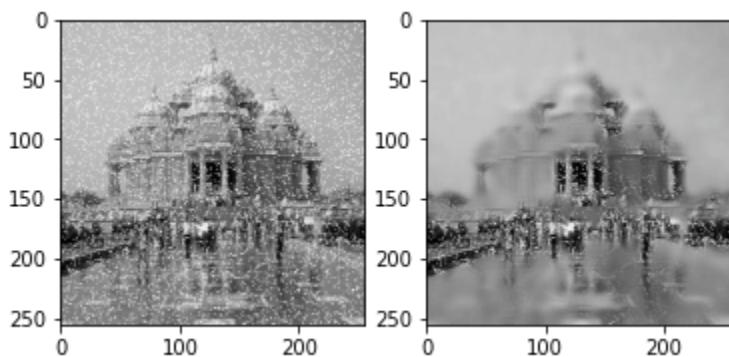
(c) Subtract Image 1 with Image 2.



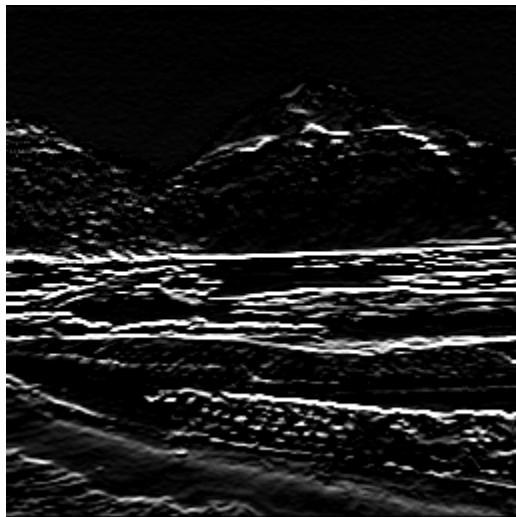
(d) Add salt noise with 5% probability in one of the images.



(e) Re-remove the noise.



(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.



References: All the functions are referred from Geeksforgeeks

Colab

link: https://drive.google.com/file/d/1vu4Ux2GDYArrow1pt_usG8IdcKtdpokn/view?usp=sharing

Q6. 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.

Ans.

1. Import the Zipped folder named TRAIN.ZIP,TEST.ZIP to the colab manually by mounting from the google drive.

2. Unzip the folder and we will see that two TRAIN AND TEST named file is created in colab. Store 100 of each 0 and 1 from train and test dataset to the imagetrain list and image test list
 3. After that store the horizontal projections of each image by np.sum axis=1 in xtrain,ytrain,xtest,ytest .
 4. Now first of all train model K Nearest Neighbour for K=7 using sklearn knnclassifier.
 5. Now test the model by passing the test dataset
 6. We have checked the model prediction score and also training vs testing graph has been observed.i.e.-93.5%
 7. Similarly we have imported Svm library using Sklearn
 8. After it train the SVM linear classifier and test the output of the model using test dataset.
 9. We have also seen the prediction score i.e.-95%

Results:

[0	0	0	0	0	1055	1651	1830	1726	1610	1403	1273	1448	1394
1317	1193	1135	1233	1144	1158	999	1168	1517	2079	1066	0	0	0	1
[0	0	0	0	765	1508	1987	2360	2729	3104	2998	2662	1955	1803
1688	1856	1873	2001	1818	2416	3020	2739	2164	1595	0	0	0	0	1
[0	0	0	0	0	666	971	2069	2254	1321	1198	1087	980	946
999	960	975	939	899	1036	1137	1258	1653	1808	1004	0	0	0	1
[0	0	0	604	1001	1210	1365	1728	1421	1372	1773	1657	1606	1581
1506	1437	1644	1642	1762	1845	2567	2407	1408	0	0	0	0	0	1
[0	0	0	0	938	1388	1840	2271	2450	2205	2285	1947	1898	1747
1622	1371	1591	1559	1576	1639	2443	2775	2373	1805	0	0	0	0	1
[0	0	0	0	0	510	1187	1615	1948	1786	1845	1872	1207	1138
1091	1033	1027	1120	1246	1236	1234	1304	1561	1879	1117	0	0	0	1
[0	0	0	0	1543	2303	3344	3692	2712	1896	1810	1670	1810	1704
1648	1600	1877	2157	2122	2497	3481	3443	2851	1844	0	0	0	0	1
[0	0	0	0	486	836	1346	1819	2077	1854	1510	1552	1657	1316
1192	1183	1155	1279	1394	1640	2045	2471	2032	932	0	0	0	0	1
[0	0	0	0	376	765	1288	2082	2850	3053	2509	1706	1446	1504
1216	1327	1309	1229	1472	2532	3674	3318	2652	1089	0	0	0	0	1
[0	0	0	0	0	1498	1967	2326	2404	2868	2764	2285	2206	1858
2006	2006	2007	1781	1603	1604	2091	2993	2646	2163	1197	0	0	0	1
[0	0	0	0	0	98	1749	2690	2636	2376	1959	1928	1972	1762
1773	1603	1606	1668	1460	1597	2065	2808	2314	958	0	0	0	0	1
[0	0	0	0	672	1279	1573	1964	2381	2129	1744	1651	1495	1324
1345	1339	1185	1246	1120	1331	1996	2367	1866	1088	0	0	0	0	1
[0	0	0	0	544	688	1276	2023	2411	2414	2054	1461	1374	1412
1513	1441	1439	1514	1383	1586	1889	2591	2545	1838	0	0	0	0	1
[0	0	0	0	0	194	1452	2576	2967	2945	2043	1828	1652	1800
1663	1824	1862	1855	2044	2290	2865	2946	2222	1063	0	0	0	0	1
[0	0	0	0	618	1252	1283	1520	1464	1514	1545	1497	1529	1560
1722	1723	1736	1849	1763	1721	1722	1933	1778	1054	0	0	0	0	1
[0	0	0	0	0	19	742	1782	2960	3830	3648	2990	2417	1835
1801	1984	1733	1698	1825	2066	2940	3144	2343	750	0	0	0	0	1
[0	0	0	0	647	1000	1285	1618	1876	2534	2307	2157	2233	2335
1810	1798	1668	1875	1847	2456	2694	2461	1743	922	0	0	0	0	1
[0	0	0	0	542	1192	1527	1422	1424	1368	1372	1431	14	1

- ## 2 Horizontal projection of the images as the list

Test results of KNN model to Test dataset is passed through it:

.Prediction score and Visualisation of single image

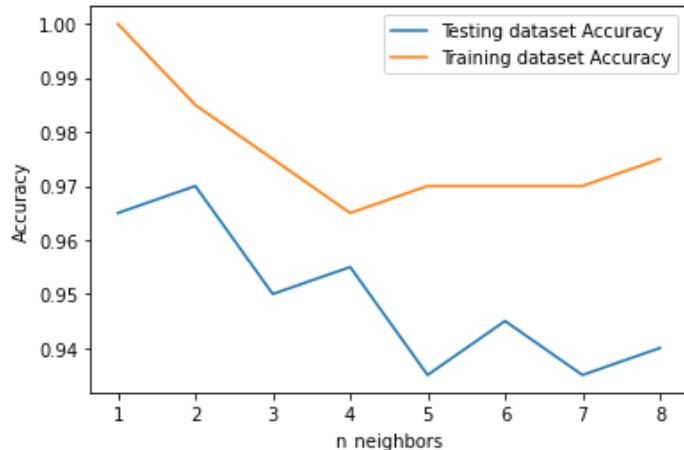
0.935

KNN MODEL accuracy for the test data set is 93.5 %



predicted value of the image is [1]

4.Graph between Training and test dataset Accuracy:



For SVM classifier:

1.test dataset when passed through Svm model:

```
[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

2.Predictions And Scores:

0.94

SVM MODEL accuracy for the test data set is 94.0 %

3.Visualisation of the Random images:



predicted value of the image is [0]

Assumptions:

1. MNIST DATASET is very large ,so here we have used only 100 images of 0s and 1s .not all the digits and also not all the images ,so that our model can run fast.
2. There is compromise in the accuracy little bit due to limited and low dataset ,training would not be so efficient.
3. MOUNT the gdrive in the colab and also upload the two zipped folder as TRAIN. Zip and TEST.zip which consist of 0s and 1s images of train and test part respectively.

Limitations:

1. We have used Library of Sklearn for training the model.
2. If the folder name or path is changed then our code cannot import datas from the folder.

References:Geeksfor geeks model implementation of KNN and SVM

Colab

Link:<https://colab.research.google.com/drive/1Co9e0IT47CNsevuww3wLuZYIVzgODTiC?usp=sharing>

Q7.Given a word image find out if the word is bright text on a dark background or dark text on bright background.

Ans:

1. Imported required libraries .
2. Here in order to check the dark or light background we have used to find gradient of image in x and y direction
3. Using sobel filter we have calculated the gradient over x and y direction.
4. It has been observed that for the light background grad is decreasing as we move from left to right in x direction.
5. similarly,It has been observed that for the dark background grad is increasing as we move from left to right in x direction.
6. Hence here we have calculated the gradient of both the images and shown the results by comparing the particular row gradients whether it is increasing or decreasing.

Results:



Bank

Displayed image in the Code.



WHO

Bank

Displayed image in Gray Scale.

For Img1:



img1 WHO is :Dark background

For IMG2:



img2 BANK IS Dark background

Assumptions:

1. Here we have taken only single row of the gradient array obtained using sobel filter.
2. Using 1 row we have noticed the change in gradient and decided whether its a light or dark background.

References: geeksforgeeks referred for sovel filter code .

Colab

link:<https://colab.research.google.com/drive/1HRnKUmmUlVKaBMaeEcq9ZZ51AoH5Jtan?usp=sharing>

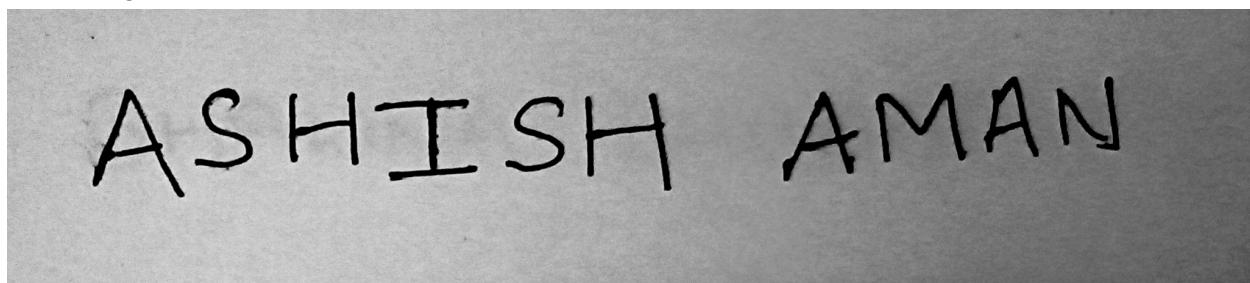
Q.8. Write your name in capital letters on a piece of white paper and a random letter from your name. Click photographs of these. Implement the Template Matching algorithm and discuss your observation.

Ans:

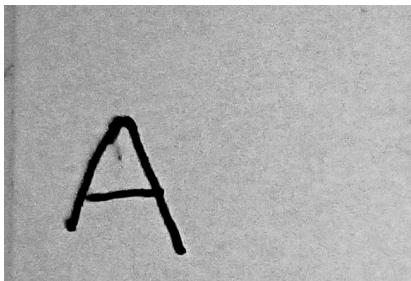
1. Imported the required libraries.
2. Wrote the name 'ASHISH AMAN' and template character 'A' separately for performing template matching.
3. Upload both image into colab
4. Read and displayed the both image.
5. Crop the template to [150:450,90:300]
6. Resize the template to resize(template,(120,250))
7. After performing all the image processings to the template we finally use TM_CCOEFF_NORMED method of template matching to the grayscale image of name.
8. Set the threshold according to the output.
9. Finally after hit and trial threshold=0.4626
10. We have seen that it has detected two As out of 3s As.

Results:

Main image of name'ASHISH AMAN'



Template:'A'

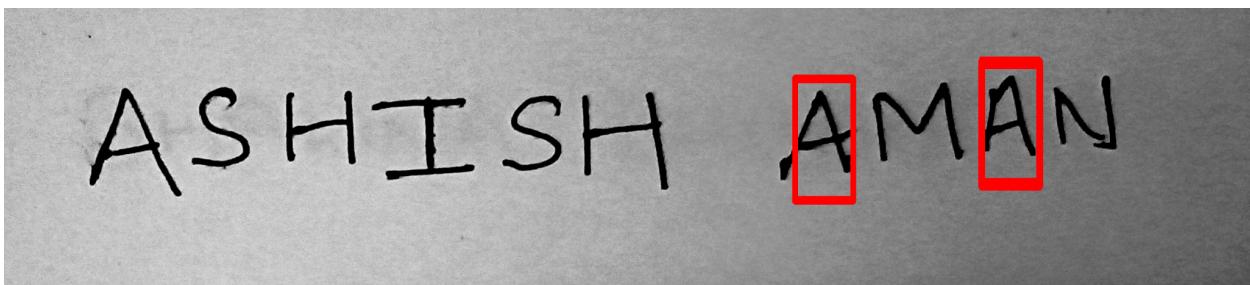


649 441

Resized Template:



The output of the Template Matching:



Assumptions & Observations:

1. We have taken the font style and writing style same of image and template .
2. Normally template should have all the image parameter same as of original image matching part.but here we have written A again and again processed it so that it can perform well.
3. As we can see 2 As are detected at the threshold value-0.4626
4. If we decrease the threshold then it will start detecting little bit of M I and then 1st A .
5. So maintain that particular threshold value and used TM_CCOEFF_NORMED for template matching.

Limitations :

1. Use the mentioned threshold for this image,
2. If the image imported will change then the result will be different at the particular threshold.
3. Hence our code is partially static for this image only as we have to adjust the shape size of the template .
4. For better template matching results , we can crop the A from the main image but it doesn't make any sense .

References:geeks for geeks -implementation of template matching.

Colab

Link:<https://colab.research.google.com/drive/1kSDu7uyIx3SBWENWjUOPUKDMGZcG75Kk?usp=sharing>

Q9.Choose one image from Problem 5. Show histogram of pixel values with bin size 10.

Perform histogram equalization and

Show the output image.

Ans:

1. Chosen image is AKSHARDHAM TEMPLE from the landmark dataset.
2. Import the required libraries and read and shown the image of temple.
3. Histogram equalization is performed on the black and white image in which it counts the frequency of the grayness on the graph in the form of histogram.
4. We have plotted the histogram using calchist fn of cv2 library with the bin size 10..
5. After equalizing the histogram and hstacking we get the output image.

Results:

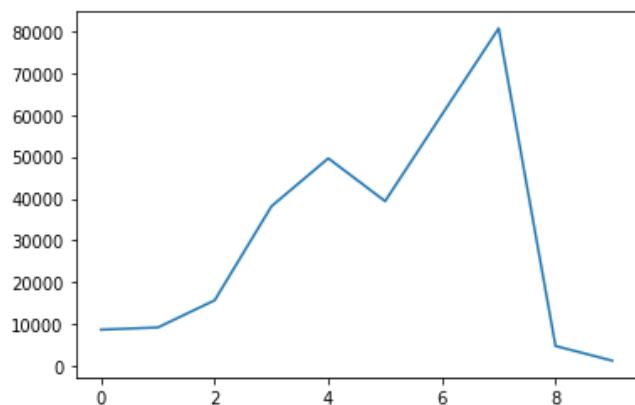
Image of AKSHARDHAM



Gray Scale image of it and Output after performing Histogram Equalization to it:



Histogram Plot:



Assumption:

1. We have taken grayscale image for the histogram plot.
2. Bin size=10 as we can see the plot as it is mentioned in the question.

References: Gfg blog for referring.

Colab

Link:<https://colab.research.google.com/drive/1LUpUAjl1qHYM3bAYhyUY7y66hGixZAh1?usp=sharing>

Q10. You will be given image of a mobile number. Use off-the-shelf OCR and find out the last three digits of the mobile number.

Ans.

1. Installed easeocr in the colab
2. Imported the library from the package
3. Set the language to read as EN
4. And import the image of both mobile no.s and displayed .
5. Applied the easeocr fn to both the image
6. And finally list of texts and coordinates is returned.
7. From it we have extracted texts part i.e. mobile no. digits in two string variables.
8. After it we have displayed last three digit of both the strings (mob no.s) using indexing

Results:

Images of Mobile No.s:

9160450925

9160450815



EXTRACTED mobile no is:-

1st mobile no is 9160450925

2nd mobile no is 9160450815

FINAL RESULTS:

last three digits of the 1st mobile no. 925

last three digits of the 2nd mobile no. 815

Colab

link:-https://colab.research.google.com/drive/1NLK9EfCQWURq_JXSb8bDEB2RklwSPgiy?usp=sharing