COMPUTER VISION

ASSIGNMENT-2A REPORT

-K VAMSI(M22RM002)

Q-1: METHOD-1

Procedure:

1.First after loading images in to variable they are converted to gray scale image.

2.Then color space into bins, which represent the range of possible color or intensity values. The number of bins depends on the complexity of the images and the desired level of detail in the similarity measurement.

3. Histograms of the images are computed by counting the number of pixels in each bin. This gives us a vector representation of the color or intensity distribution of each image.

4.Normalize the histograms to account for differences in image size or brightness. This can be done by L1 norm here. And I have appended labels and histograms into a array, which is then used to train K nearest neighbour model.

5.Then test image is given to the trained model to Calculate a similarity metric between the normalized histograms. Model will predict the possible output, then I have put text on the image that corresponds to predicted label.

6. The process is repeated for a set of reference images to find the closest match or matches to the query image.

Observations:

1.I have observed that levis containing image is classified as kfc, because the histogram of both levis and kfc are matching as they have same amount color intensity distribution.

2.Starbucks image is correctly predicted by the model as the color and intensity distribution between target and query image matched.

METHOD-2

Procedure:

1. The first I have done prepossessing the data. As the data set had only 10 images, I have done data Augmentation that includes different variations of the same logo to train the model. The logos that were pre-processes had resizing, normalization, and standardization of the image data.

2. The next step I have designed the neural network architecture. This involves deciding on the number of layers, the number of neurons in each layer, and the activation functions to be used.

3. The neural network model is then trained using the training data. During training, the model learns to recognize and match logos by adjusting its weights and biases.

4.Once the model has been trained, the performance of the model is measured using metrics such as accuracy.

5.Finally I have done testing on given two reference images.

Observations:

1.I have observed that levis containing image is classified as taco bell. Model was not able generalize on test data even after data augmentation.

2.Starbucks image is correctly predicted by the model as Starbucks.

METHOD-3

Procedure:

1.The logo images are pre-processed to remove any noise, and then they were converted to gray scale image.

2.Then SIFT object is used to detect important features in the logos. SIFT detects key points in the image that are scale and rotation-invariant, making it robust to changes in image scale and orientation. A descriptor is created that describes the local appearance of the image at the key point location.

3.Flann Matcher is used to find corresponding key points between two images.

4. RANSAC (Random Sample Consensus) is used to estimate the homography matrix between the matched key points. Homography is a transformation matrix that maps points in one image to their corresponding points in another image.

5. Finally match between the logos is given based on the degree of alignment and similarity of the transformed images using a metric called inlier ratio. The image with large inlier ratio was labelled as output.

Collab Link:

https://colab.research.google.com/drive/1jVaxQ_eIr5aTKxdcXh6nG7_DiJUZNi1V?usp=share_link

Observations:

1.I have observed that levis containing image was correctly classified as levis. The inlier ratio between image given and logo was the highest when compared to other given images.

2. I have observed that Starbucks containing image was correctly classified as Starbucks. The inlier ratio between image given and logo was the highest when compared to other given images.

Q-2:

Method-1(From scratch implementation)

Procedure:

1.First image is loaded in to a variable. Then the image is converted to Gray scale as Hough line is performed on Gray scale image.

2.Canny edge detector algorithm is applied in order find the binary image of the edges in the given image.

3.Accumulator array of two dimensions is created for both rho and theta. For each pixel in the edge image the possible lines are found. And for each computed parameter value corresponding element in the accumulator array is increased. It is done by voting for every possible line along the accumulator array.

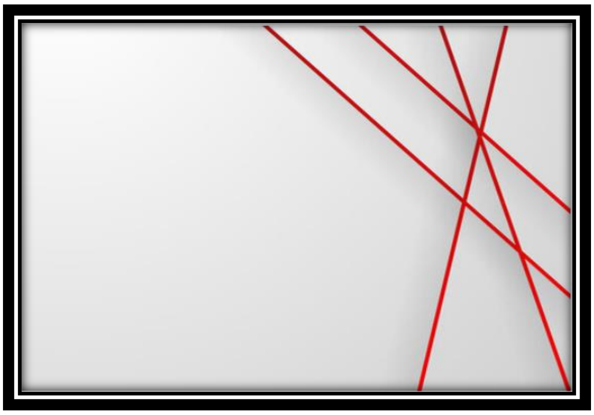
4.After iterating over all pixels in the edge image, accumulator array will contain set of points, the highly concentrated points correspond to the original lines in the image.

5.I have plotted high concentrated parameter points on the graph as shown below.

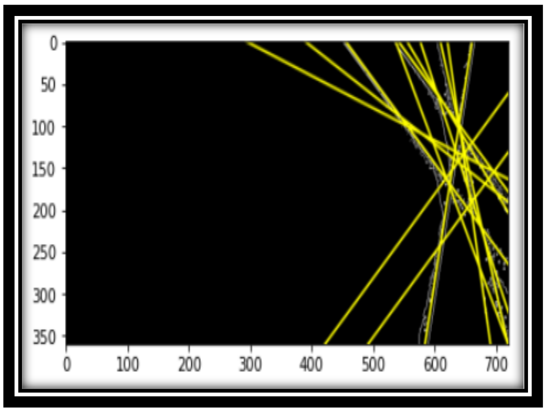
6.To extract lines from accumulator array I have iterated through parameter values and found out corresponding theta and rho values. And then the line is drawn on the original image using those values. Finally, I have plotted the image with detected lines.

Results:

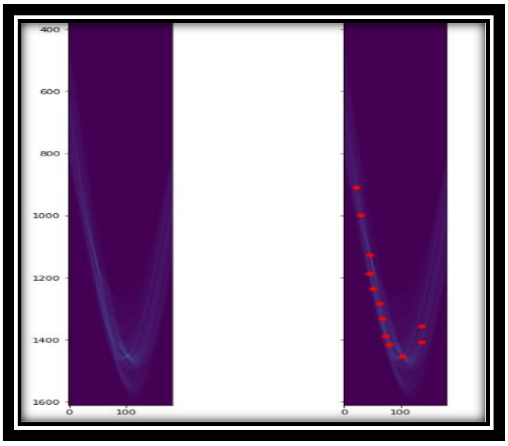
Original Image



Output Image



Rho and theta values:



Method-2(Inbuilt implementation)

Procedure:

1.First image is loaded in to a variable. Then the image is converted to Gray scale as Hough line is performed on Gray scale image.

2.Canny edge detector algorithm is applied in order find the binary image of the edges in the given image.

3.Then using inbuilt Hough I have found the Hough line transform and then plotted the lines on them.

Results:

Collab Link:

https://colab.research.google.com/drive/16fHK2f8kzJU_V-pOv0T1Q7RXp0KVYI9c?usp=share_link

Observations:

In Hough line method each line is represented by pair of rho and theta values. This method is sensitive to noise and discontinuous in the image. Performance of Hough Transform is affected by choice of parameters for accumulator array, such as range of rho and theta values.

1.It is observed that the code implemented from scratch has taken more time than inbuilt function code i.e. (Execution time: 12.193171739578247 seconds and Execution time: 0.08428287506103516 seconds).

2.From performance point of view both the methods were performed equally well and detected eleven lines.

Q-3:

Procedure:

1. Given images are pre-processed to remove any noise, scale or rotate them, and resize them to a consistent size.

2.Then the images are converted to gray scale and ORB is used to detect features in the images. ORB detects key points in the image that are scale and rotation-invariant, making it robust to changes in image scale and orientation.

3. Descriptors are computed for each key point. A descriptor is a binary string that describes the local appearance of the image at the key point location.

4. "DESCRIPTOR_MATCHER_BRUTEFORCE_HAMMING" is the Match the descriptors used to find corresponding key points between two images.

5.RANSAC is used to estimate the homography matrix between the matched key points. Homography is a transformation matrix that maps points in one image to their corresponding points in another image.

6.The similarity between the images based on the degree of alignment and similarity of the transformed images is calculated. Based on the similarity image is classified as faulty or perfect.

7.If the image is found faulty using canny edge detection and thresholding method the contours are drawn over the faulty region. If the image is found perfect nothing is done.

Collab Link:

https://colab.research.google.com/drive/1GtcQfbjB688uf4EJo5q4nQcJME5QPxoy?usp=share_link

Observations:

Example-1

1.For two Perfect images given one image was misclassified as the similarity value became low when compared to other images.

2. For two Faulty images given one image was misclassified as the similarity value became higher than perfect image showing maximum similarity with reference image when compared to other images.

3. Differences are occurring because of scale variances between two images.

4. Images contain noise, which can cause the ORB algorithm to detect false key points or miss important ones. This is one of the reasons in low-quality images or those captured in low-light conditions.

Example-2

1.For two Perfect images given one image was misclassified as the similarity value became low when compared to other images.

2. For two Faulty images both were classified correctly as the similarity value became higher thus showing maximum similarity with reference image when compared to other images.

3. The two images being compared have undergone different transformations, such as scaling, rotation, or distortion, this clead to differences in the key points detected by the ORB algorithm.

COMPUTER VISION

ASSIGNMENT-2B REPORT

-K VAMSI(M22RM002)

Q-1:

A homography matrix is a transformation matrix that maps points from one plane to another.

-It allows us to shift from one view to another view of the same scene by multiplying the Homography matrix with the points in one view to find their corresponding locations in another view.

-It describes the geometric relationship between two planar images. It can be used to align two images and create a panoramic view, image registration, or 3D reconstruction.

-The homography matrix is a 3x3 matrix that maps points in one image to corresponding points in the other image. The matrix is derived using corresponding points in both images and solving a set of linear equations.

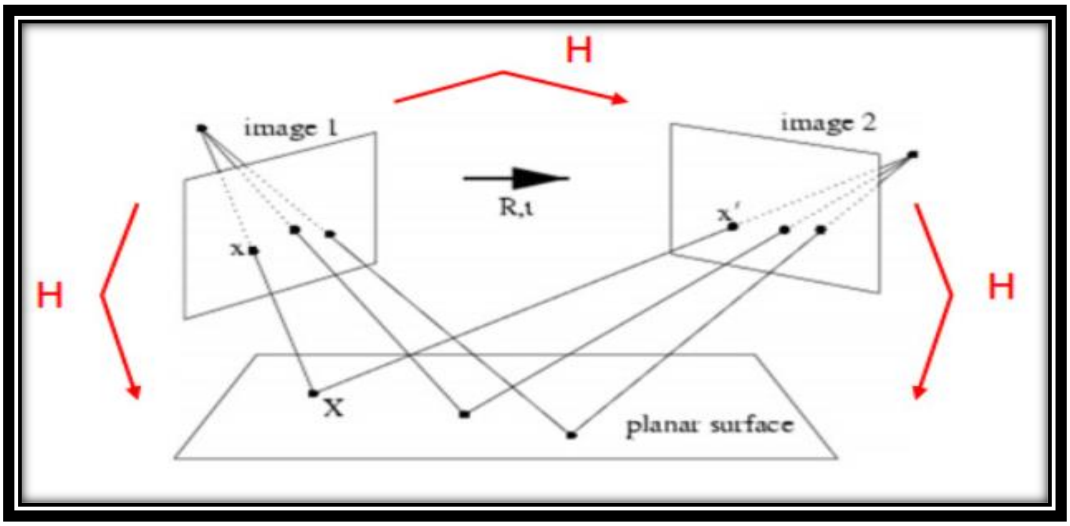
Here are the steps to compute a homography matrix:

1.Feature Detection: The first step is to detect features in both images. Common feature detection methods include SIFT, SURF, or ORB.

2.Feature Matching: Once features are detected, the next step is to match features between the two images. This can be done using techniques such as K nearest-neighbour matching or RANSAC.

3.Calculate Homography Matrix: Once corresponding feature points are identified, the homography matrix can be computed. This can be done using various methods such as Direct Linear Transformation (DLT) or normalized DLT.

4.Refine Homography Matrix: Once the homography matrix is computed, it may be necessary to refine it to improve the accuracy of the transformation. This can be done using techniques such as Levenberg-Marquardt optimization or RANSAC. Once the homography matrix is refined, it can be applied to one of the images to align it with the other image. This can be done using techniques such as bilinear interpolation or warping.



Mathematically finding homography matrix(H):

Computing Homography:

Given a set of matching features / points b/n images 1 & 2, find homography H that best "agrees" with matches. Let (x_d, y_d) be destination image points corresponding to (x_s, y_s) in the source image.

Then
$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_{H} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix}$$

In homogeneous coordinate s/m.

- Here H is homography matrix

- In order to find 'H' we require min^n 4 pairs of matching points. But in general we take all matching points to compute 'H' in order to make estimate of homography more robust.

For a given pair i of corresponding points:

$$x_d^{(i)} = \frac{\tilde{x}_d^{(i)}}{\tilde{z}_d^{(i)}} = \frac{h_{11} x_s^{(i)} + h_{12} y_s^{(i)} + h_{13}}{h_{31} x_s^{(i)} + h_{32} y_s^{(i)} + h_{33}}$$

$$y_d^{(i)} = \frac{\tilde{y}_d^{(i)}}{\tilde{z}_d^{(i)}} = \frac{h_{21} x_s^{(i)} + h_{22} y_s^{(i)} + h_{23}}{h_{31} x_s^{(i)} + h_{32} y_s^{(i)} + h_{33}}$$

Rearranging:
$$x_d^{(i)}\left(h_{31} x_s^{(i)} + h_{32} y_s^{(i)} + h_{33}\right) = h_{11} x_s^{(i)} + h_{12} y_s^{(i)} + h_{13}$$
$$y_d^{(i)}\left(h_{31} x_s^{(i)} + h_{32} y_s^{(i)} + h_{33}\right) = h_{21} x_s^{(i)} + h_{22} y_s^{(i)} + h_{23}$$

Rearranging the terms & writing as linear eq^n:

$$\underbrace{\begin{bmatrix} x_s^{(i)} & y_s^{(i)} & 1 & 0 & 0 & 0 & -x_d^{(i)}x_s^{(i)} & -x_d^{(i)}y_s^{(i)} & -x_d^{(i)} \\ 0 & 0 & 0 & x_s^{(i)} & y_s^{(i)} & 1 & -y_d^{(i)}x_s^{(i)} & -y_d^{(i)}y_s^{(i)} & -y_d^{(i)} \end{bmatrix}}_{\text{known}} \underbrace{\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}}_{\text{unknown}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Combining eqn for all corresponding points:

$$\underbrace{\begin{bmatrix} x_s^{(1)} & y_s^{(1)} & 1 & 0 & 0 & 0 & -x_d^{(1)}x_s^{(1)} & -x_d^{(1)}y_s^{(1)} & -x_d^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_s^{(i)} & y_s^{(i)} & 1 & 0 & 0 & 0 & -x_d^{(i)}x_s^{(i)} & -x_d^{(i)}y_s^{(i)} & -x_d^{(i)} \\ \vdots & & & & & & & & \\ x_s^{(n)} & y_s^{(n)} & 1 & 0 & 0 & 0 & -x_d^{(n)}x_s^{(n)} & -x_d^{(n)}y_s^{(n)} & -x_d^{(n)} \\ 0 & 0 & 0 & x_s^{(n)} & y_s^{(n)} & 1 & -y_d^{(n)}x_s^{(n)} & -y_d^{(n)}y_s^{(n)} & -y_d^{(n)} \end{bmatrix}}_{A\ (\text{known})} \underbrace{\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}}_{h\ (\text{unknown})} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Now we need to solve for $Ah = 0$ such that $\|h\|^2 = 1$.

It can be solved using constrained least squares.

Define least squares problem:

$$\min_h \|Ah\|^2 \text{ such that } \|h\|^2 = 1$$

$\|Ah\|^2$ can be written as $\|Ah\|^2 = (Ah)^T(Ah) = h^TA^TAh$ & $\|h\|^2 = h^Th = 1$

$$\therefore \min_h (h^TA^TAh) \text{ such that } h^Th = 1$$

Define Loss fn $L(h,\lambda)$

$$L(h,\lambda) = h^TA^TAh - \lambda(h^Th - 1)$$

differentiating $L(h,\lambda)$ w.r.t h: $2A^TAh - 2\lambda h = 0$

$\boxed{A^TAh = \lambda h}$ eigen value problem.

$\therefore$ Eigen vector h with smallest eigenvalue λ of matrix A^TA minimises loss fn $L(h)$. Now we rearrange 'h' vector into 3x3 matrix which gives us homography matrix.

References:

https://www.youtube.com/watch?v=J1DwQzab6Jg&list=PL2zRqk16wsdp8KbDfHKvPYNGF2L-zQASc

Q-2:

The process of stereo matching involves several steps:

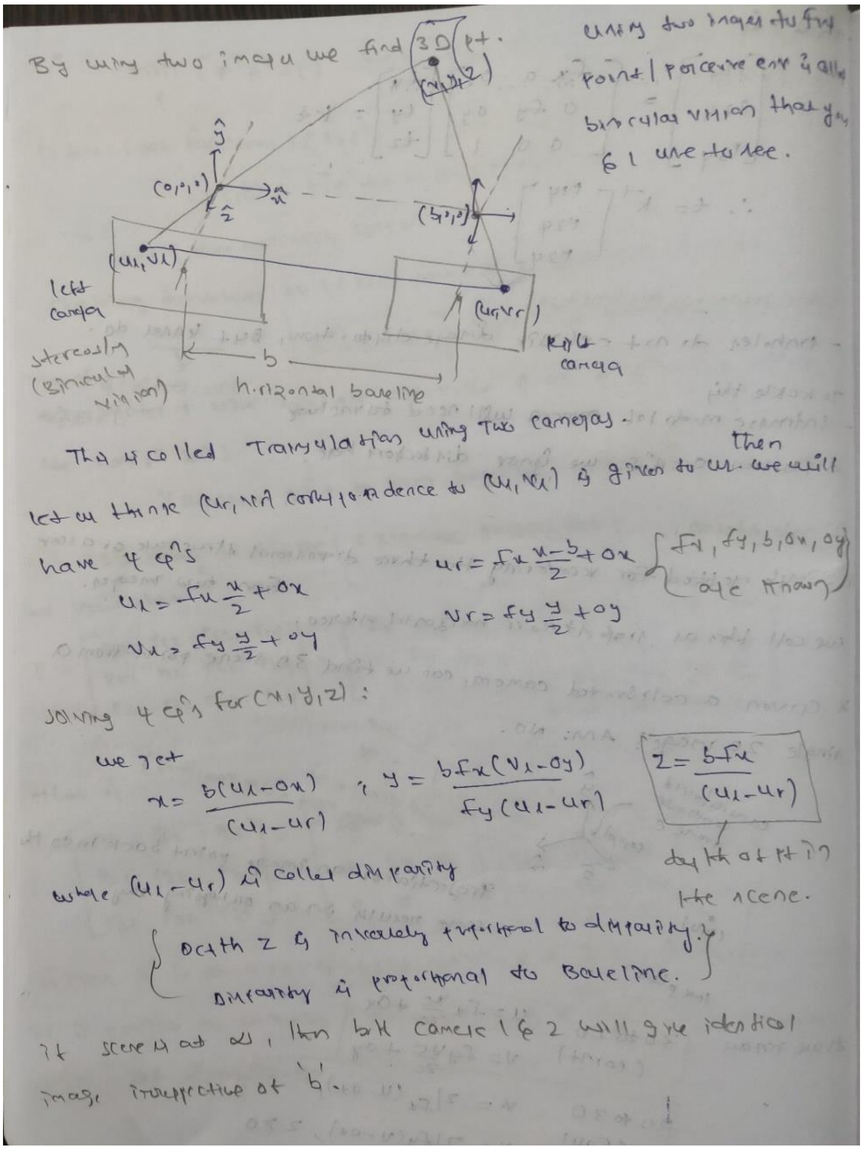
1.Image Rectification: The first step in stereo matching is to rectify the images so that corresponding points lie on the same horizontal scan line. This is done to simplify the matching process and reduce the search space.

2.Feature Extraction: The next step is to extract features from the rectified images. Common features used for stereo matching include corners, edges, and SIFT/SURF features. These features are then matched between the two views to find corresponding points.

3.Disparity Computation: Once corresponding points have been found, the disparities between them are computed. This can be done using various techniques such as block matching, correlation-based matching, or optical flow.

4.Depth Estimation: Finally, the computed disparities are used to estimate the depth of each point in the scene. This can be done by triangulation, which involves computing the intersection of rays originating from the camera centres and passing through the corresponding points as shown below.

Mathematically depth can be found as:


By using two images we find 3D pt. (x, y, z)

using two images to find point / perceive env is also binocular vision that you & I use to see.

$(0,0,0)$ $\hat{y}$ $\hat{x}$ $\hat{z}$ $(b,0,0)$ (u_l, v_l) (u_r, v_r)

left camera

right camera

stereoscopy (binocular vision)

b

horizontal baseline

This is called Triangulation using two cameras.

let us think (u_r, v_r) correspondence to (u_l, v_l) is given to us. Then we will have 4 eq^n's

$u_l = f_x \frac{x}{z} + o_x$

$v_l = f_y \frac{y}{z} + o_y$

$u_r = f_x \frac{x-b}{z} + o_x$

$v_r = f_y \frac{y}{z} + o_y$

$\{f_x, f_y, b, o_x, o_y\}$ are known

solving 4 eq^n's for (x, y, z):

we get

$x = \frac{b(u_l - o_x)}{(u_l - u_r)}$; $y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)}$

$\boxed{z = \frac{b f_x}{(u_l - u_r)}}$

depth of pt in the scene.

where $(u_l - u_r)$ is called disparity

Depth z is inversely proportional to disparity.

Disparity is proportional to Baseline.

If scene is at ∞, then both cameras 1 & 2 will give identical image irrespective of 'b'.

Stereo matching has many applications in computer vision, including 3D reconstruction, object tracking, and depth sensing. It is widely used in fields such as robotics, autonomous driving, and augmented reality.

Applications:

1.3D Reconstruction: Stereo matching can be used to estimate the 3D geometry of a scene by analysing two or more images taken from different viewpoints. This can be used for applications such as building 3D models of objects or environments, creating virtual reality experiences, or assisting in medical imaging.

2.Depth Sensing: Stereo matching can be used to estimate the depth of objects in a scene, which can be useful in applications such as autonomous vehicles, robotics, or gesture recognition.

3.Object Tracking: Stereo matching can be used to track the movement of objects in a scene by computing the disparities between the object's features in different frames.

4.Medical Imaging: Stereo matching can be used in medical imaging to create 3D models of organs or other structures, which can be used in diagnosis or treatment planning.

References:

https://www.youtube.com/watch?v=hUVyDabn1Mg

Q-3:

Image Stitching Process:

1.Image Acquisition: The first step is to acquire a set of images of the scene. The images should be taken from different viewpoints but should overlap to some extent to allow for stitching.

2.Feature Detection and Matching: The next step is to detect and match features between the images. Common feature detection methods include SIFT, SURF, or ORB. Once features are detected, they need to be matched between the images to determine which features correspond to the same point in the scene.

3.Image Alignment: After feature detection and matching, the next step is to align the images. This can be done using various techniques such as homography estimation. The goal is to find a transformation that aligns the images, taking into account any differences in camera position or orientation.

4.Image Blending: Once the images are aligned, the next step is to blend them together to create a seamless panorama. This can be done using techniques such as multi-band blending or feathering.

5.Post-Processing: After the panorama is created, some post-processing may be necessary to improve the quality of the result. This can include adjusting brightness and contrast, removing lens distortion, or correcting for colour balance.

Collab Link:

https://colab.research.google.com/drive/1adk7pj4dHzRc-2jCDA5qjWGaQ_m4ScOj?usp=share_link

Result:

Method-1: Using various functions as defined in the process above. Implementation from scratch.

Observations:

-Here I have observed that blending of the image has not taken well and brightness variance is not mitigated.

Method-2: Using inbuilt function and post processing.

Observations:

-Here I have observed that stitching of the images was done quite good after doing post processing of the stitched image.