Big Dipper 22

Mentor - Manasvi Vaidyula

Friend Blend

Sreenya Chitluri	2020102065	ECE
Tejah S S	2020112028	ECD
Sankeerthana Venugopal	2020102008	ECE
Maulesh Gandhi	2020112009	ECD

Link to the repo

Structure

Colour Normalisation

Histogram Equalisation

Face And Body Detection

Haar Cascade Classifier

Homography Estimation

Key point detection, Key point matching, Computing the homography

Image Blending

Alpha blending, GrabCut

Color normalisation

- Histogram Equalisation

Histogram Equalisation

- After capturing the input photos, they are color adjusted to produce better mixing results. If the lighting circumstances are sufficiently varied between the two seeds, merging one on top of the other results in a picture that appears to have been manipulated.
- Since lighting is the most prominent factor, we convert images to LAB space and apply histogram equalization on their L values. The histogram of the first image is equalized with the histogram of the second image.

Histogram Equalisation Example

Input Image 1

Input Image 2

Histogram Equalised Image 1

Face and Body Detection

- Haar Cascade Classifier

Haar Cascade Classifier

- Locating the subjects in the pictures is important for determining which image-merging process to use.
- To find where the humans are located in the images, we use face detection by Haar feature-based cascade classifiers.
- Using the location and size of the face, we determine a bounding box for the body of the human. We use a cascade of boosted classifiers based on Haar-like features to extract a bounding box for the face of each person.

Homography Estimation

- Key point detection
- Key point matching
- Computing the homography

Keypoint Detection

using openCV functions orb.detect and orb.compute

- FriendBlend uses Oriented FAST and Rotated BRIEF (ORB) keypoint detection.
- ORB keypoint detection accounts for rotation invariance, and the keypoint descriptors are essentially BRIEF descriptors for rotated patches around the key points.

Keypoint Detection from Scratch

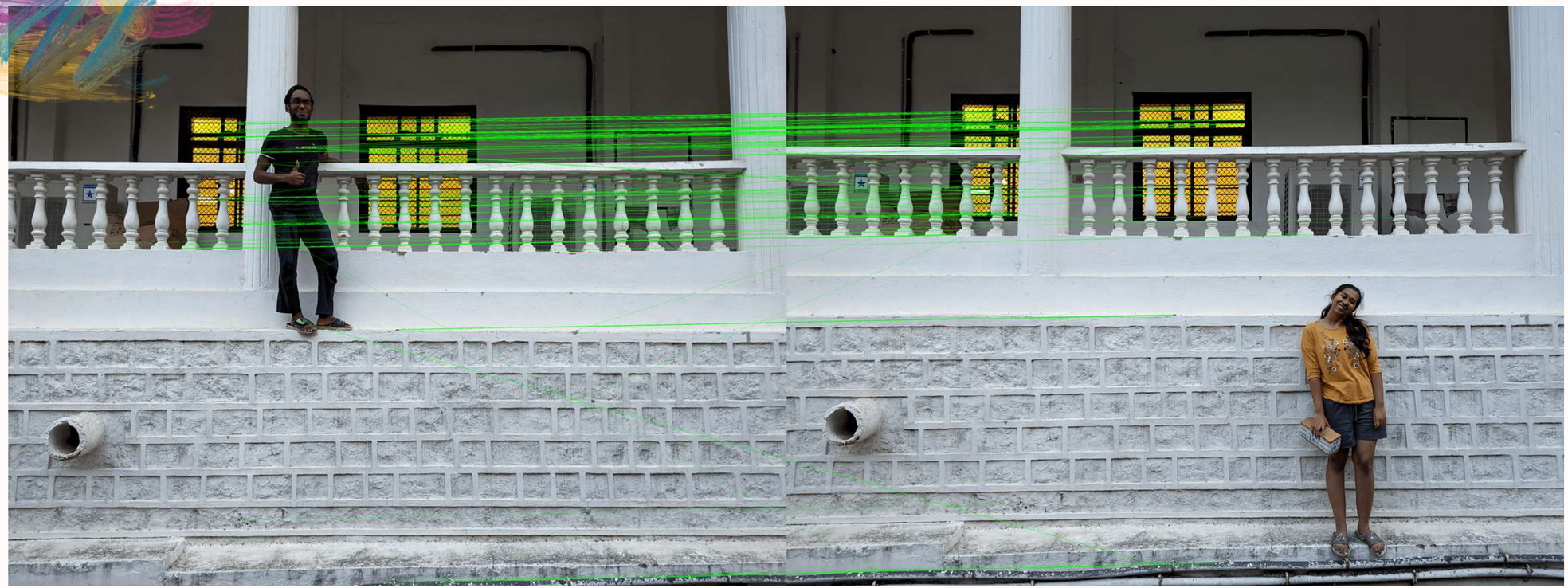
Using Harris Corner Detection

- The grayscale of the original image is used to find the corners.
- The Sobel operator is used to find the x and y gradient values for every pixel in the grayscale image.
- For each pixel in the grayscale image, we consider a 3x3 window around it and compute the corner strength response or the Harris value.
- Image pixels with Harris response crossing the threshold are counted as valid key points. The x and y coordinates of the key points and a small radius of 3 (to include the neighborhood of the key point) are sent as inputs to orb.detect to calculate the corresponding descriptors.

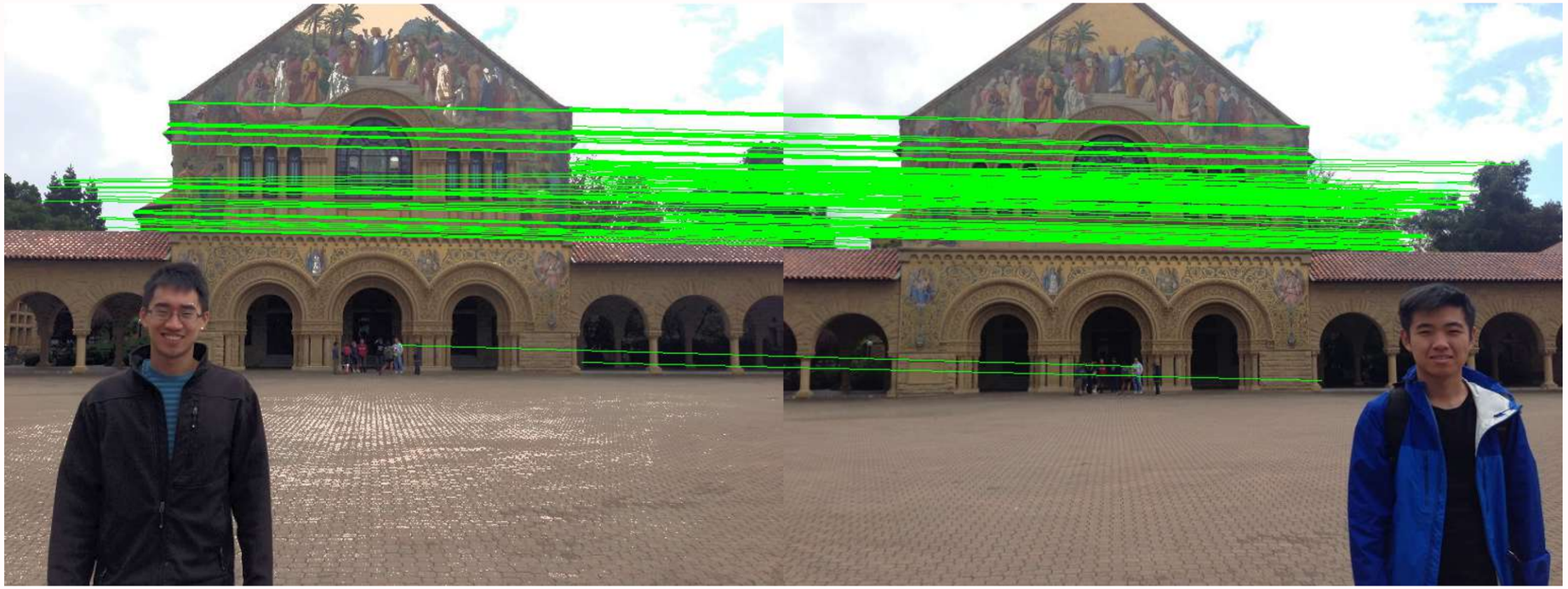
Keypoint Matching

The key points in the two images are matched using hamming distance between the descriptors. A pair of key points, one from each image, are said to be matched if the key points are within a threshold hamming distance from each other (10 times the distance between the closest key-point match in our implementation)

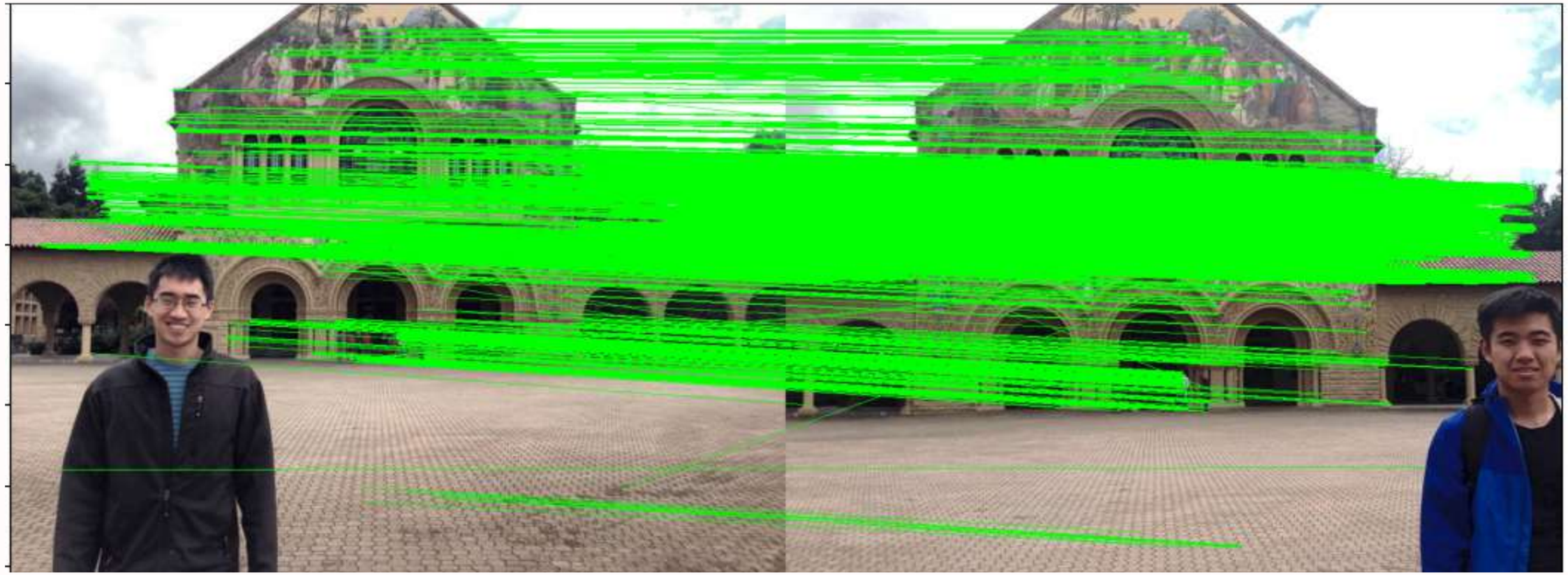
The matches are pruned based on the key-point location. Since each person is in each image once, any key-points on the person should be rejected. Key-points from the face or body in one image should not be found in the other image. Therefore, we prune a key-point match if any of the key-points are within the bounding box of a person.



Match results after ORB

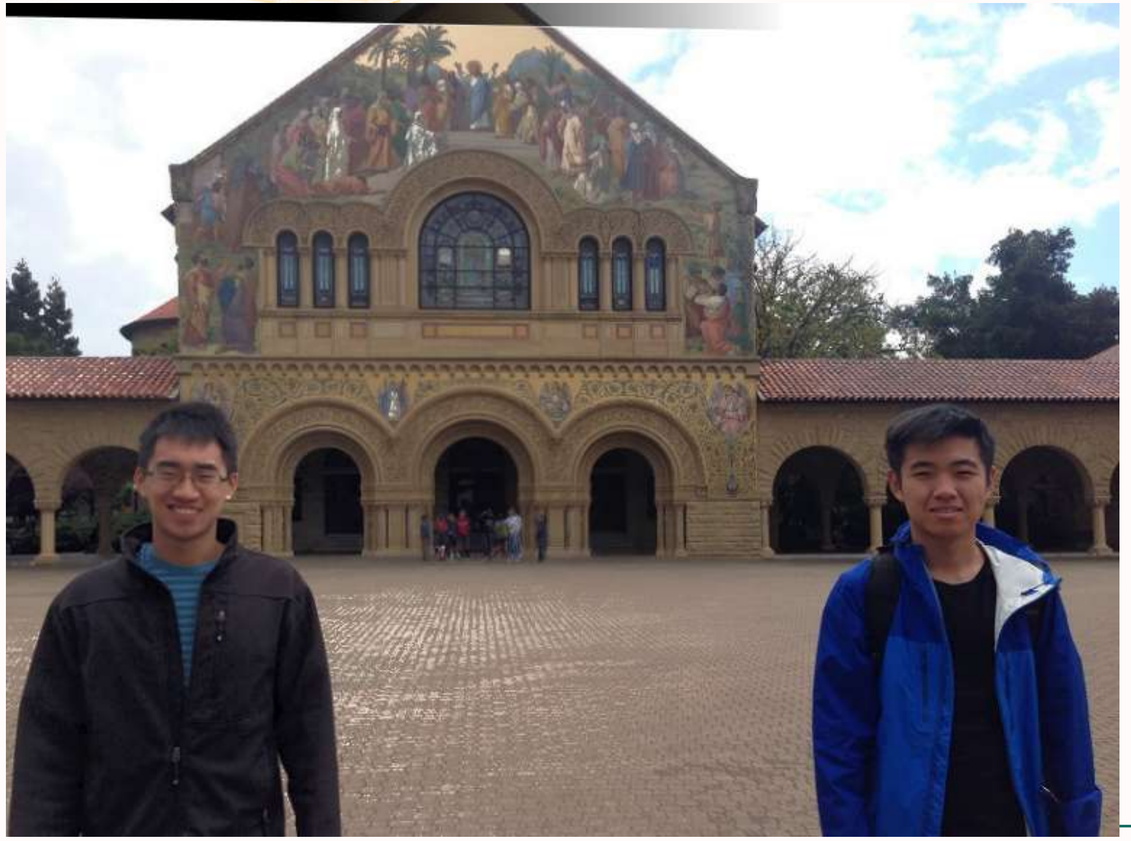


Match Results after Harris Corner Detection

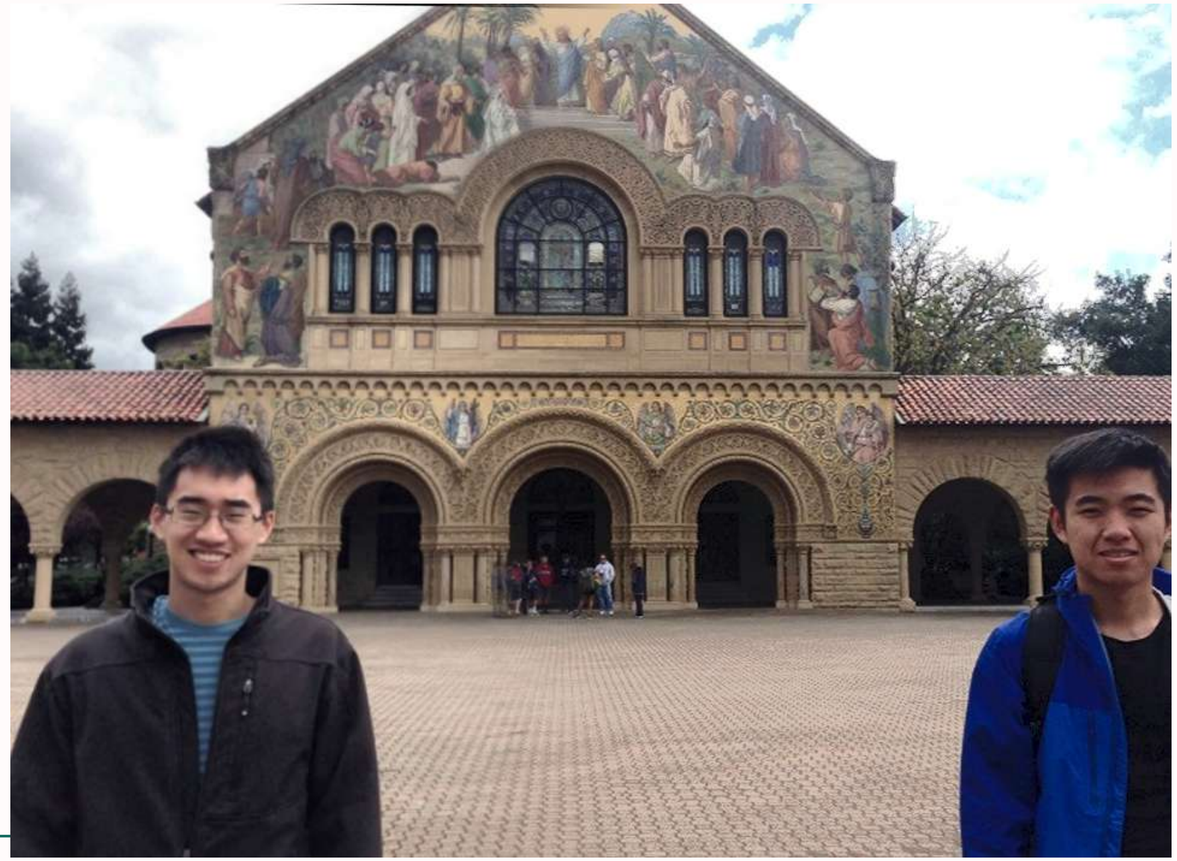


Final blended image comparison

With ORB



With Harris Corner Detection



Computing Homography

- We utilise RANSAC to determine the homography that best warps the pictures into the same viewpoint once we've located a solid collection of keypoint matches.
- Once we get the homography matrix that best maps the first image to the second, we warp the first image using the homography matrix to align it with the second one.

Warped Image

Image Blending

- Check the overlap amongst the subjects
- Alpha-Blending (if subjects are far apart)
- GrabCut (if the subjects overlap)

Overlap Checking

- Now that the subjects in the two images have been detected, the next step is to check how much they overlap.
- If it is found that the overlap is –
    a. High – we use Grab-Cut to proceed.
    b. Low – we use Alpha Blending to proceed.
- If the body boxes of the images are overlapping, then we classify them "near", else they are "far".
- We have a function to check which subject is on the right and left. Using this, we then check for the overlap.

Subject 1

Subject 2

Far Subjects Example

Subject 1

Subject 2

Overlapping Subjects Example

Alpha-Blending

The distance between the subjects allows us to easily merge that area and then crop the final image. Alpha blending is done in each color channel independently. The rightmost edge of the bounding box for the left subject is where the blending starts, and the leftmost edge of the right subject is where it finishes .

Alpha-Blending Results

GrabCut

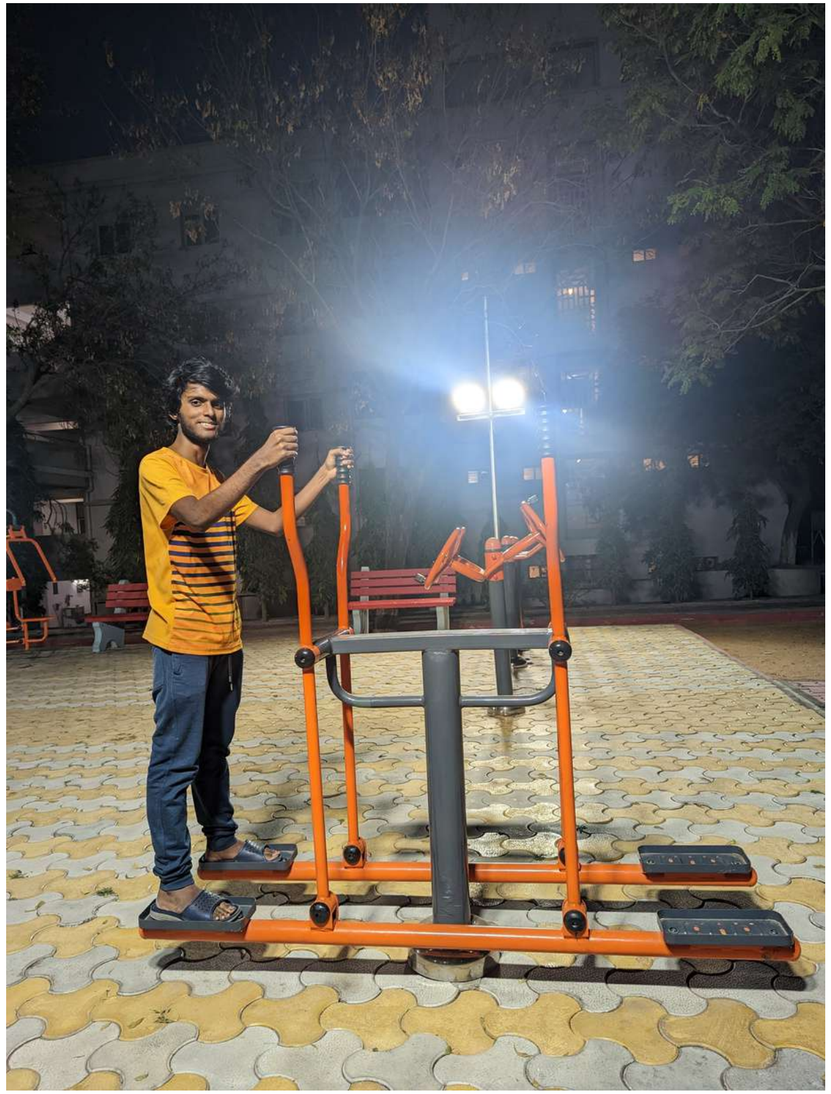
- In order to make it appear as though one subject is standing in front of the other when the topics are close together, one subject must be segmented on top of the other subject.
- We determine the foreground subject based on the size of the face bounding boxes of our subjects.
- We obtain a mask to crop the foreground subject using GrabCut. The mask is eroded using a 3x3 ellipse-shaped structuring element to remove artifacts at the edges and placed on top of the background image to get the blended image.

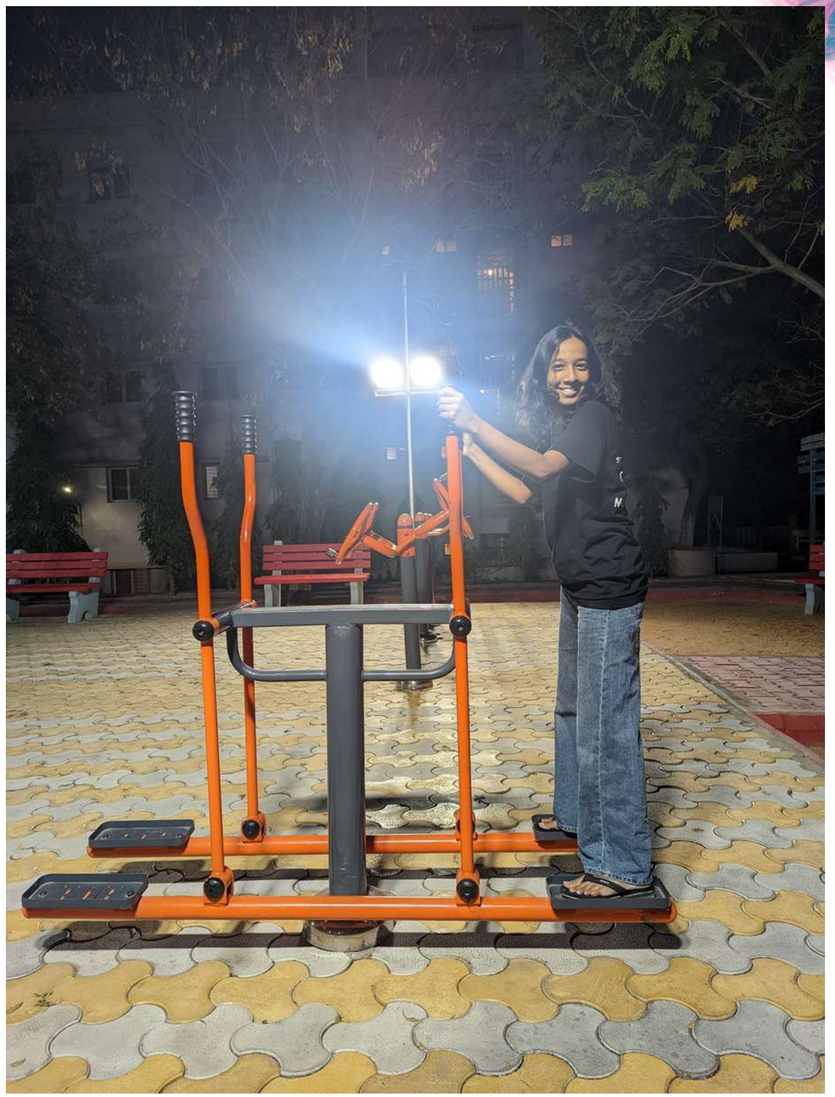
GrabCut Results

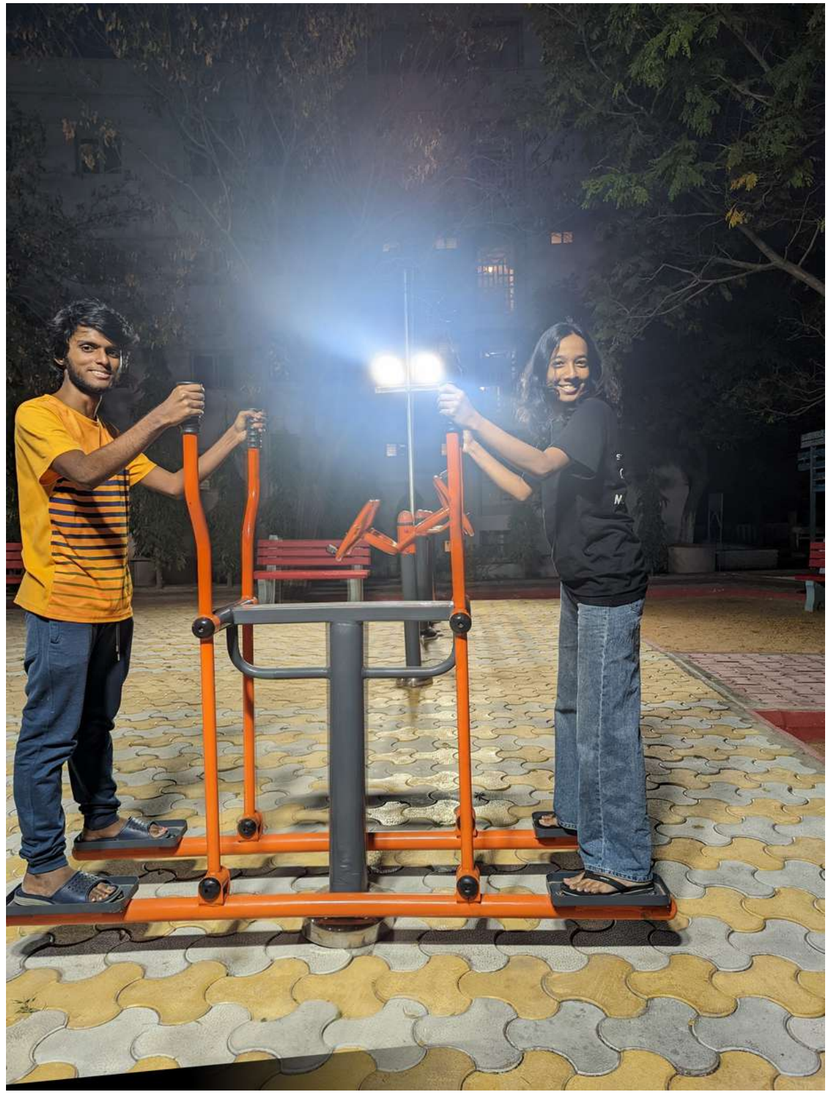
GrabCut Results

More Results

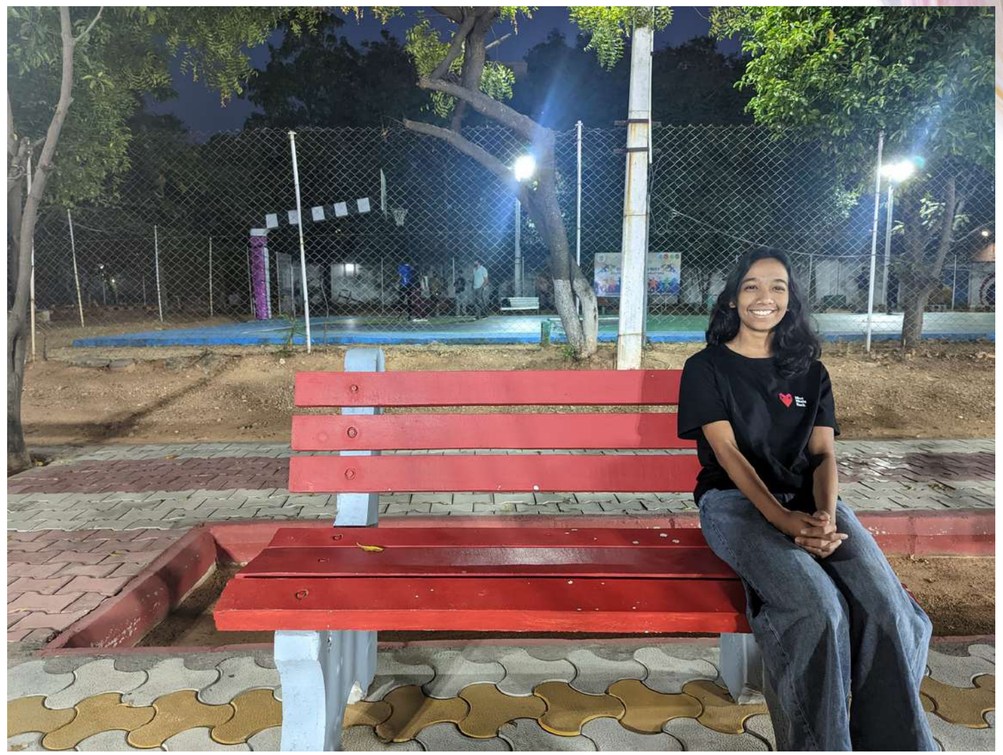
Failed due to too much overlap amongst the subjects. GrabCut can't move the subject around in our implementation.



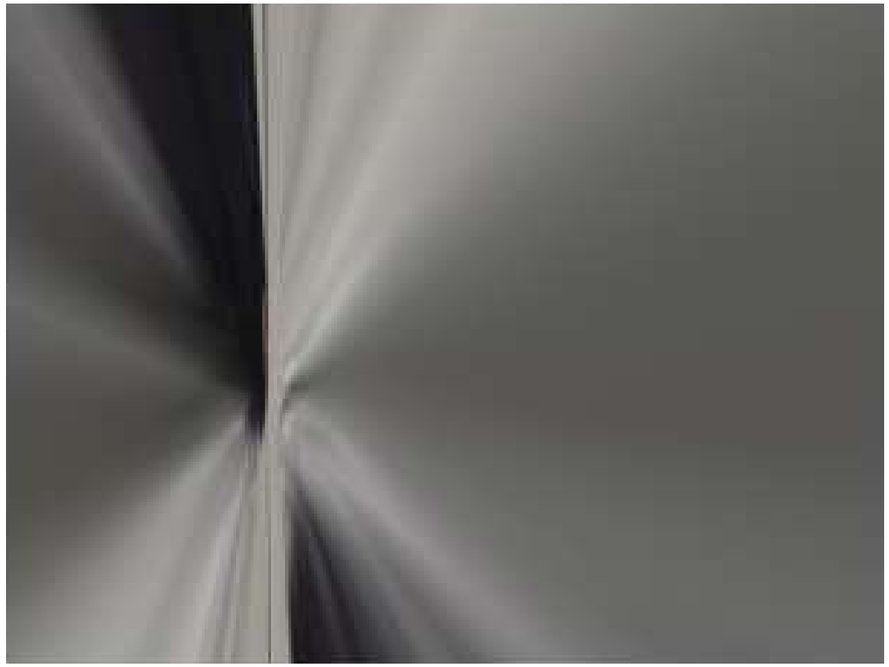




Works well due to sharp corners leading to more dense key point and accurate mappings.



Warped first image



Second image

Fails because the background is plain and hence no key points can be detected. Warping fails.


FACULTY LOUNGE


FACULTY LOUNGE


FACULTY LOUNGE