REPORT

Introduction:

In this assignment, I have implemented a mosaic creating program using python and openCV. In this, I have created a mosaic/panorama of two images using image stitching and blending techniques. In this, I have used the SIFT descriptor for finding the features and used knn matching for matching the points using the RANSAC algorithm. Then, I aligned the two images according to the homography and used the graph cut technique for finding the optimal seam and blended the images along the seam using pyramid blending.

The implementation consist of 3 main steps:-

1. Pre-Processing/Registration
2. Graph Cut Stitching
3. Pyramid Blending

Pre-Processing/Registration:

This function is used to preprocess in which the two images get aligned with each other. The function finds the position of two images and takes care of the cases when the first image is the left image,right image, upper image or lower image. Hence no ordering is required previously and the images can be passed in any order. In this function, various choices are made according to the performance obtained. The steps are:-

1. I have first read the two images from the folder passed in the argument and resized them dynamically so that the size of both the images comes within a fixed range.
2. After that, I have found the features in both the images using the SIFT feature descriptor. I have compared it with other feature descriptors like

BRISK and ORB but the speed and performance of SIFT outperforms other descriptors.

3. After that, I have matched the features using k nearest neighbours algorithm with value of k = 2 since there is a possibility of matching a wrong point with less distance. Hence kept the value of K=2 which ensures performance and speed.
4. After that, I have calculated the homography of one image with respect to another using the RANSAC algorithm and align the two images using the homographies.
5. After that, I found the intersection of two aligned images and sent the intersection area of both images into a graph cut function to find the optimal seam.

GraphCut-Stitching:

This function takes two images as input and finds the optimal seam using the graph-cut algorithm as mentioned in the paper. In this, we have used the inbuilt maxflow library to compute maxflow. So the only task remaining is to form the graph, calculate weights and to form edges. The edge weight is calculated by taking the square of difference in pixel values of source and sink so that the weight at pixels having same value at source and sink will be less and hence mincut will incorporate those values where the change in intensity is minimum.

The graph is constructed containing the same number of nodes as the size of the image. Then the edge weights are calculated between the adjacent nodes in the same row and same column by the method described above. The difference is calculated using the vectorised approach for fast calculation. Then the edges are added with the corresponding edge weights into the graph. At last, the terminal edges are added connected with either source or sink and max flow is calculated. Then the edges that remained connected are returned and a mask is created that tells the portion coming from source and sink. After that the Pyramid blending is called.

Pyramid Blending:

This function is used to blend two images along the mask provided by using the techniques of pyramid blending. I have implemented this blending as it allows creation of seamless panorama which is not in the case of direct blending. The level of blending (upto what factor we downscale or upscale the images i.e. 16 times scaling for level 3) is taken as an argument in the function. In this, we first calculated the gaussian pyramids for source image, target image and mask. Then we calculated the laplacian pyramids of the gaussian pyramids of each source, sink and mask. The pyramids are calculated by taking the difference of upscaled images and actual images. After that, we combine the image according to the mask and the image is constructed by adding the combined images of previous level to current level and the last image is returned.

Observation:

- The seams and quality of mosaics generated improves as we increase the scaling. The optimal cuts become better as image pixels increase.

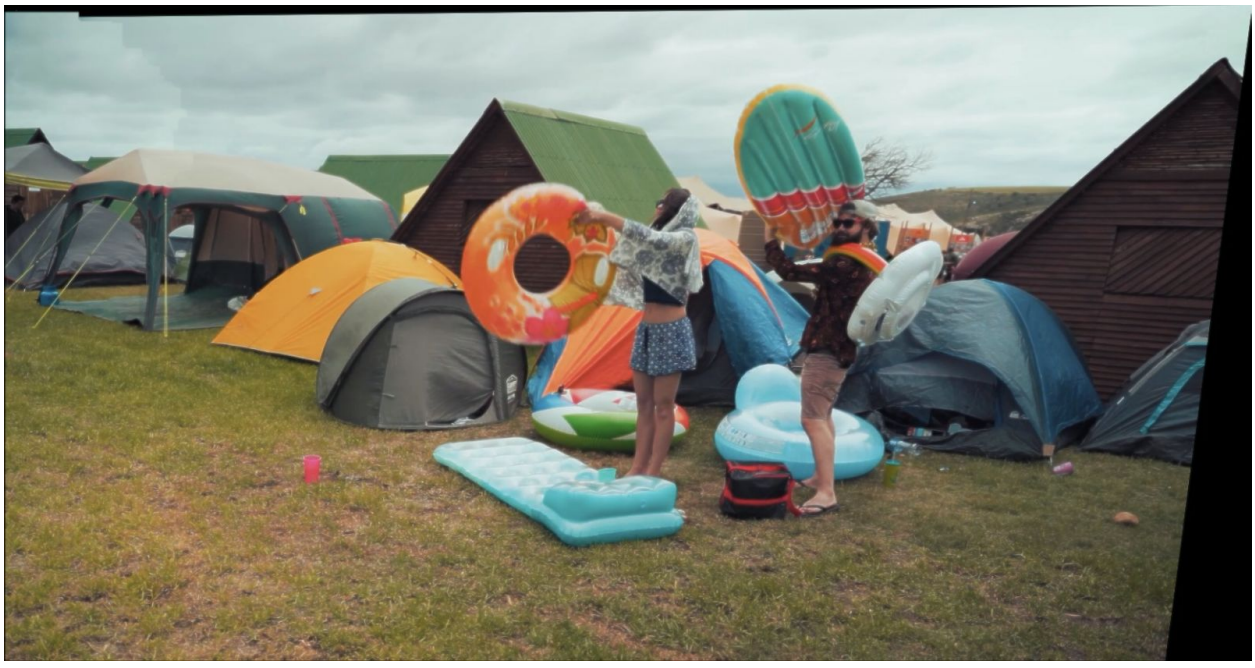
- The K nearest neighbour matching works better than normal matching as wrong points with less distance can be matched. Hence chosen k points for matching and it works fast and accurate with k=2.

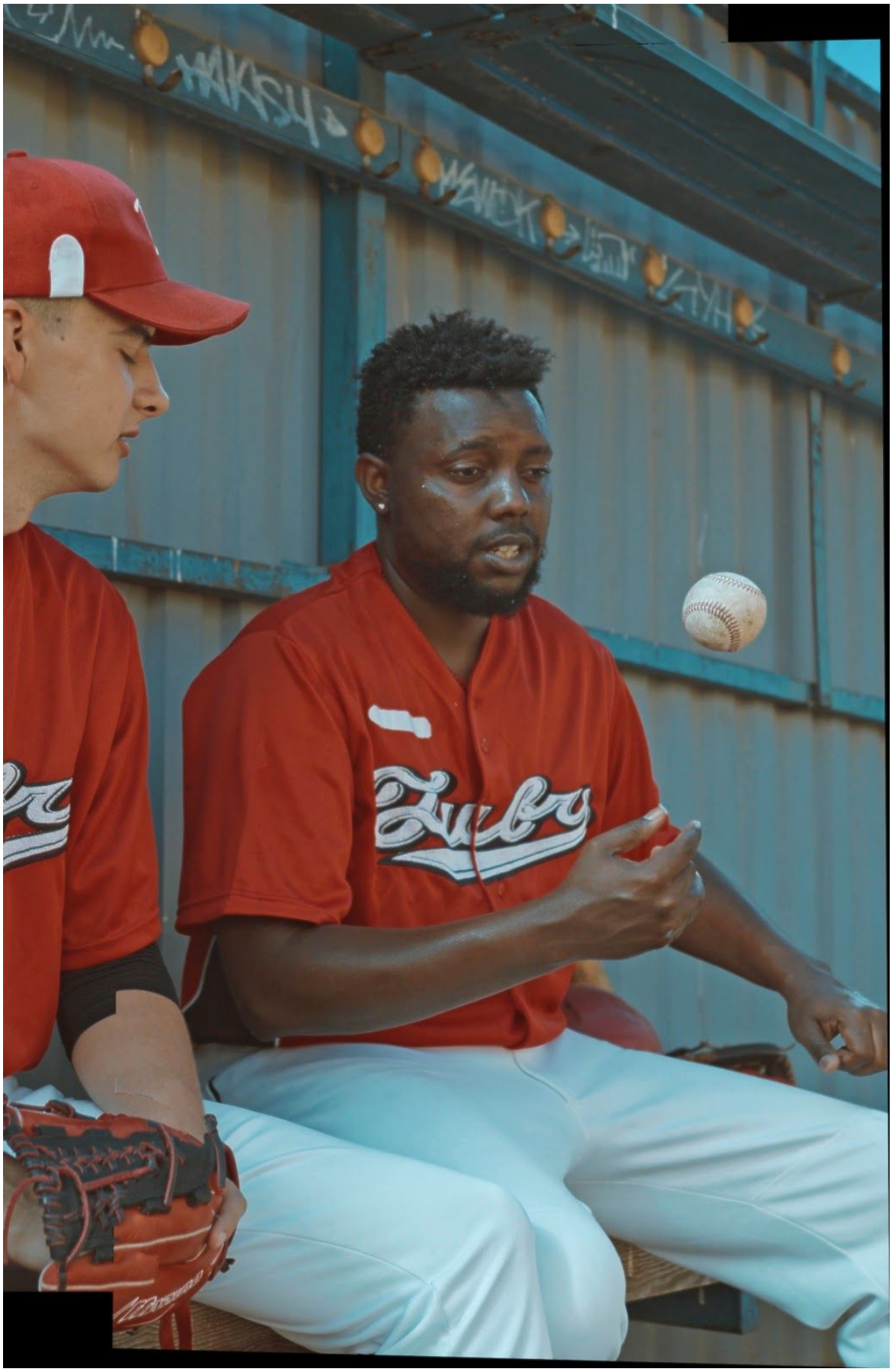
- I have used the grayscale images for graph cut calculation as I observed that the seams found are practically similar in both coloured and gray images but the time taken on gray images is much less than that of coloured images.

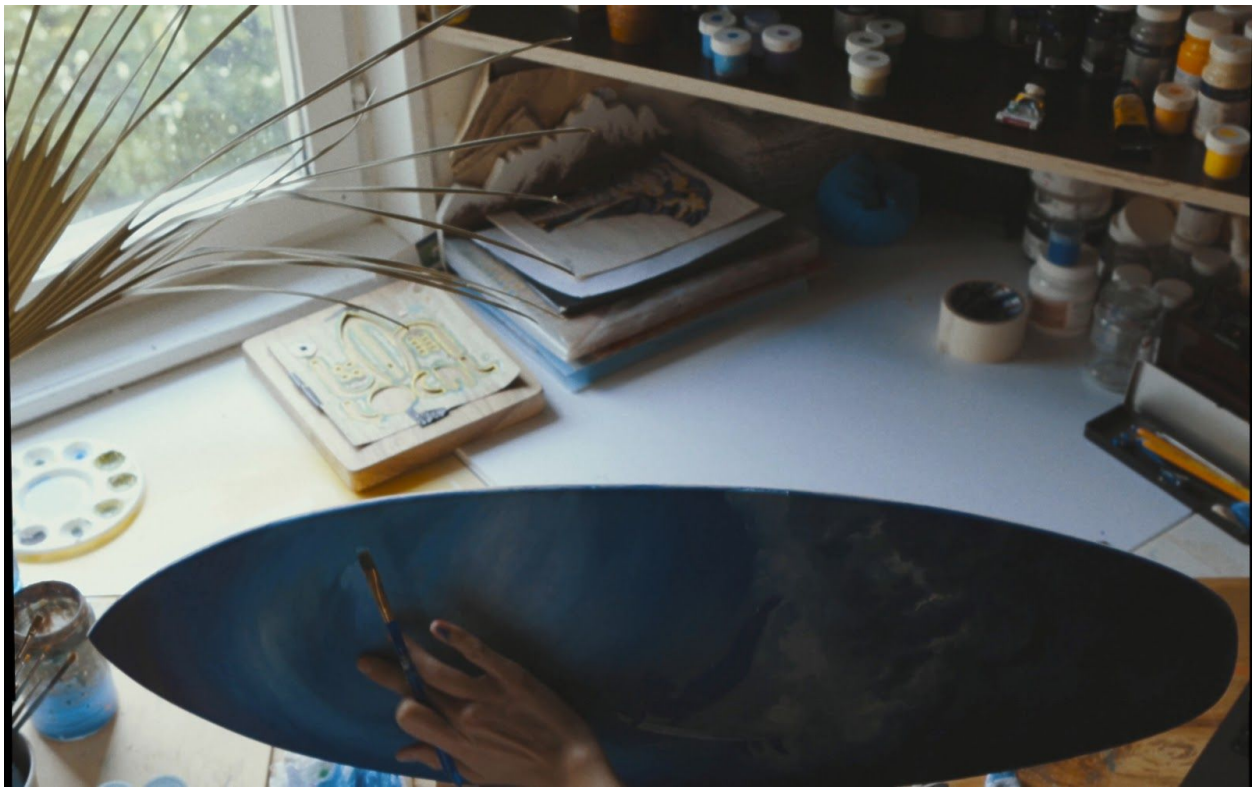
- I have observed that the seams in the case of direct blending is observable and we can distinguish the two images but in the case of pyramid blending, the region around the seam gets smoothed and the seam becomes nearly invisible.

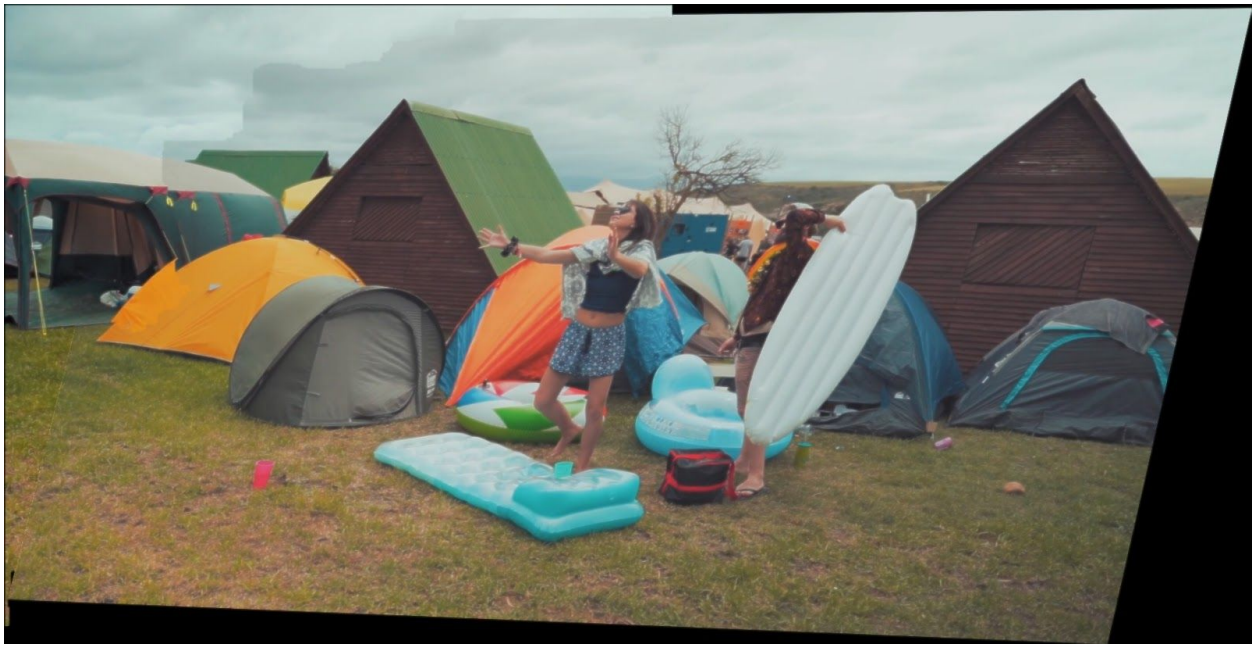
Mosaics generated on validation-set:

As we can observe from the images that the seam/cut is nearly invisible in both the cases. Also it is handling the moving objects carefully as in both the images, there are moving objects present




Zuber

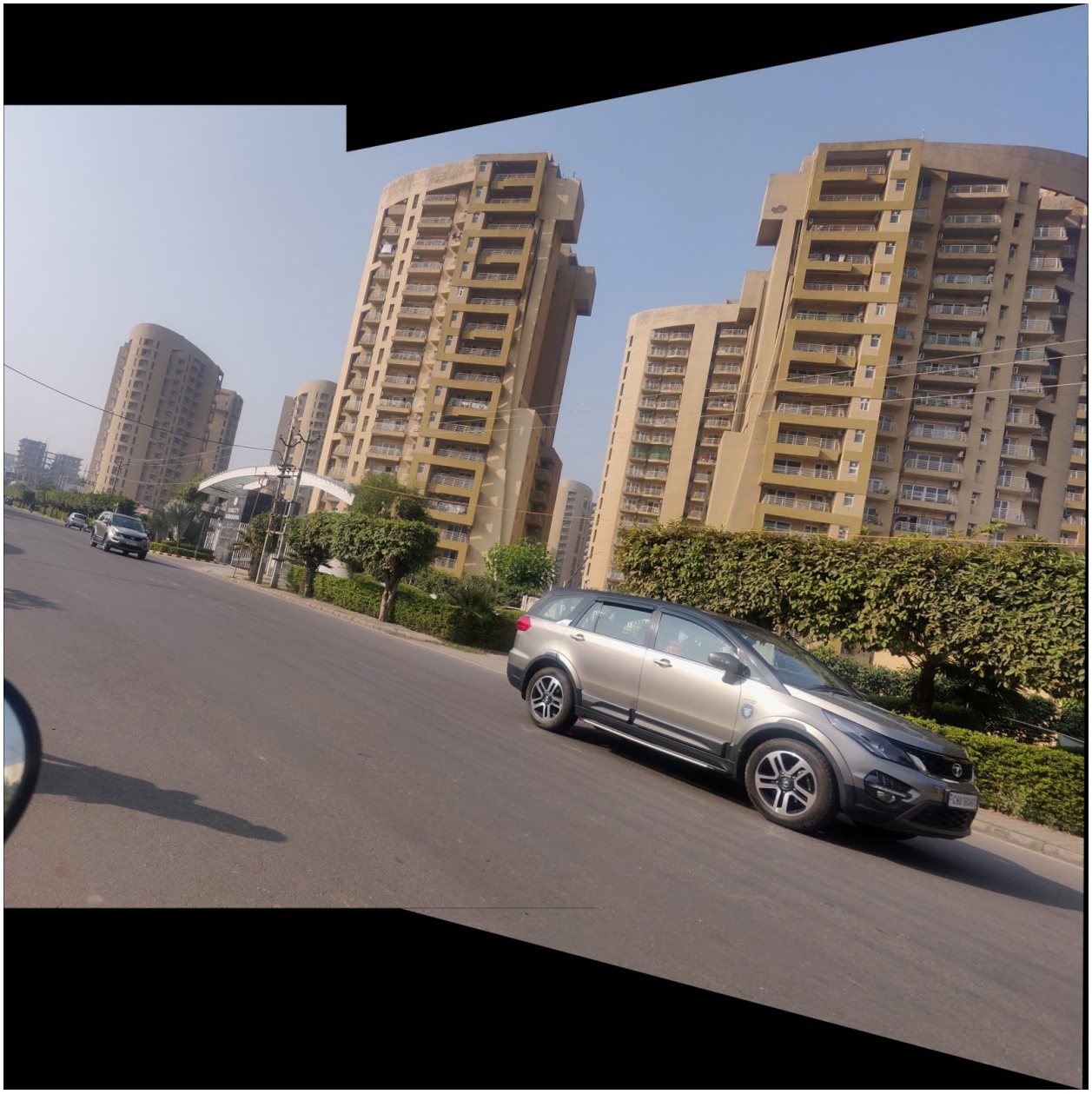




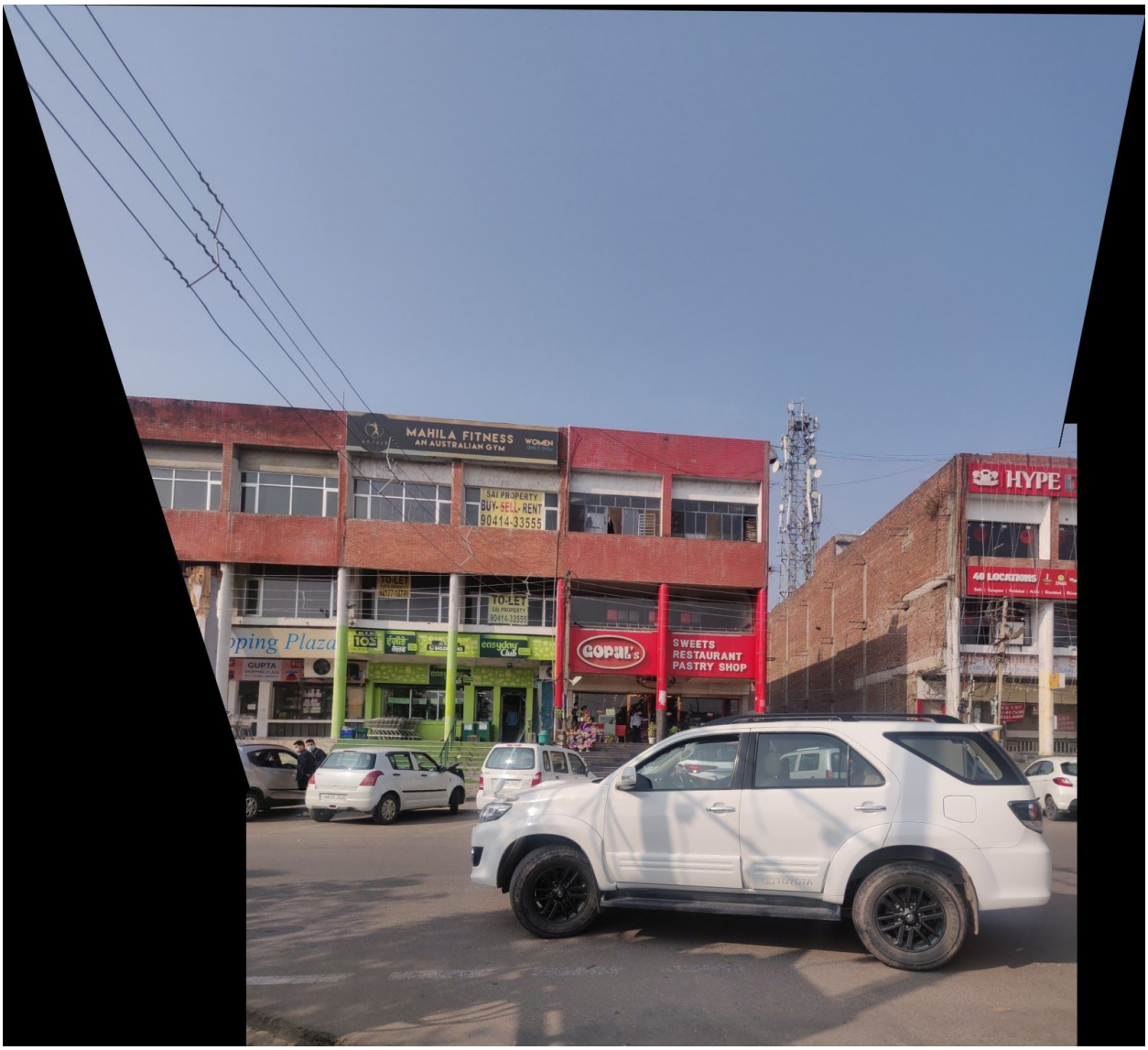

WISSES RÖSSLI
RESTAURANT

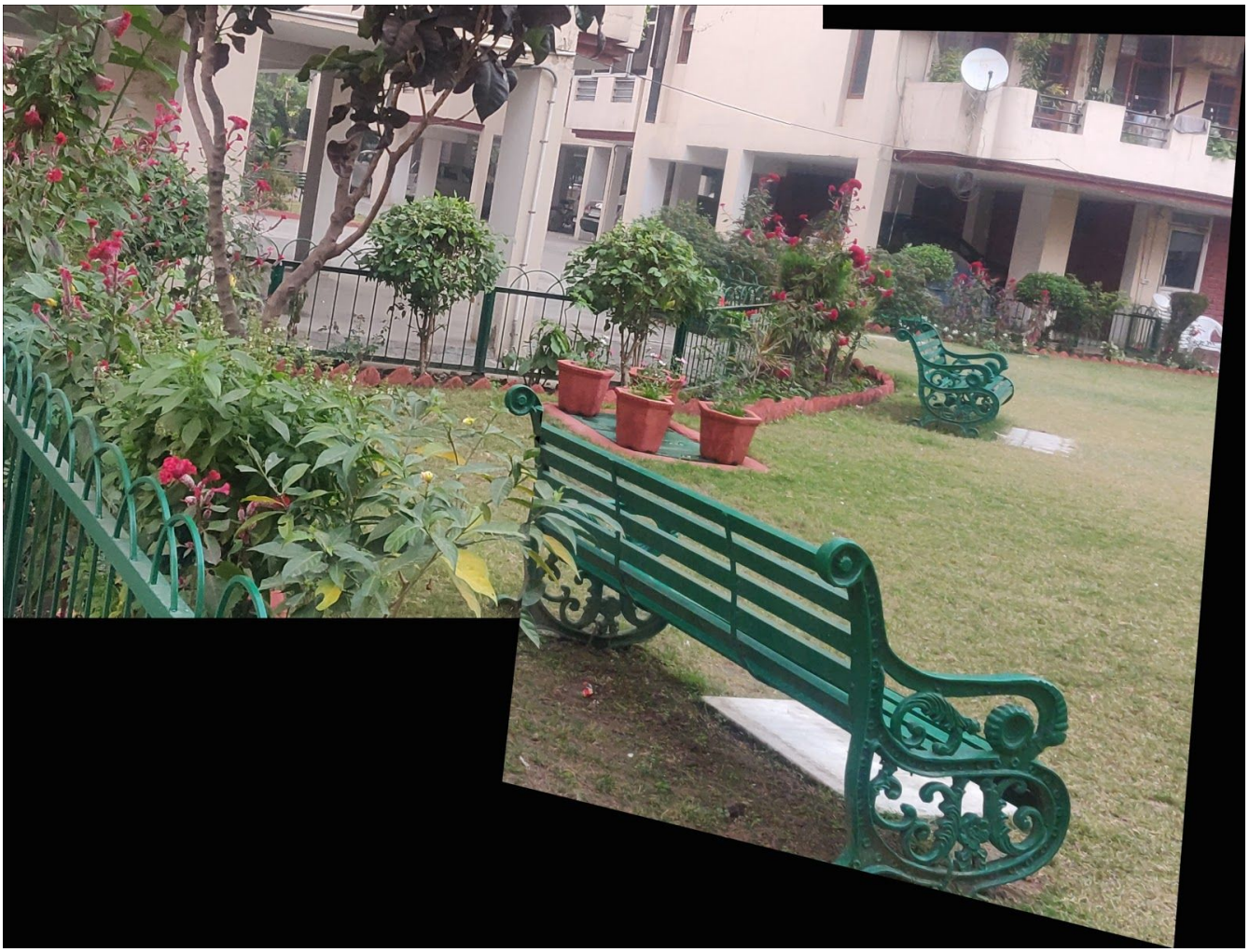
Mosaics generated on self clicked images (best):

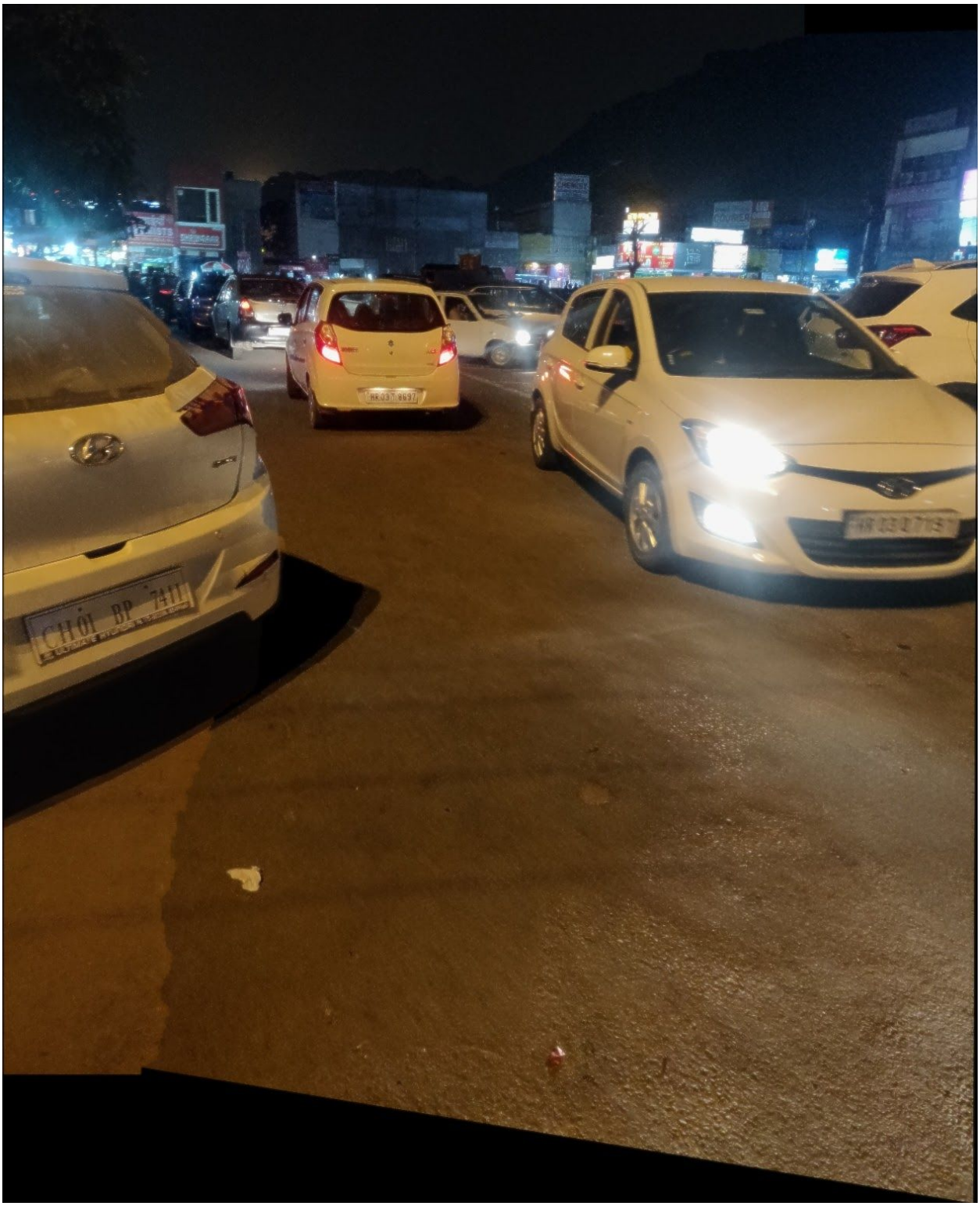
The mosaics are created for images of all types like having moving objects(car) in the 1st image and 4th image. The cars were moving when the photo was taken and the mosaics handled them quite well. The images are stitched both in horizontal direction and vertical directions and are taken in both the daylight and night light and it performs excellent in all the cases.



The photo portraying moving car (1st image)

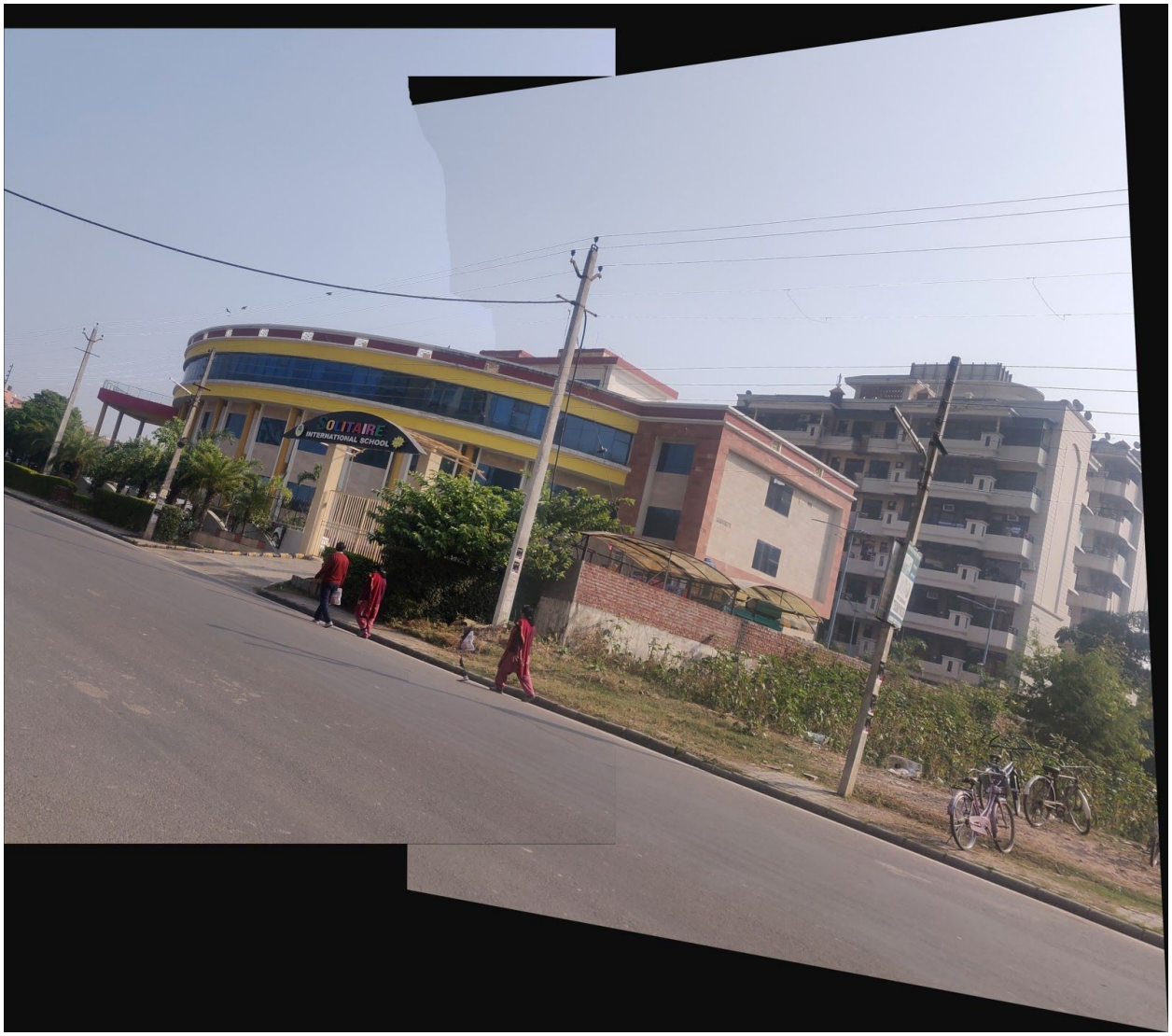

MAHILA FITNESS
AN AUSTRALIAN GYM
WOMEN
SAI PROPERTY
BUY - SELL - RENT
90414-33555
TO-LET
GOPAL'S
SWEETS
RESTAURANT
PASTRY SHOP
HYPE
46 LOCATIONS
easyday
Club
GUPTA





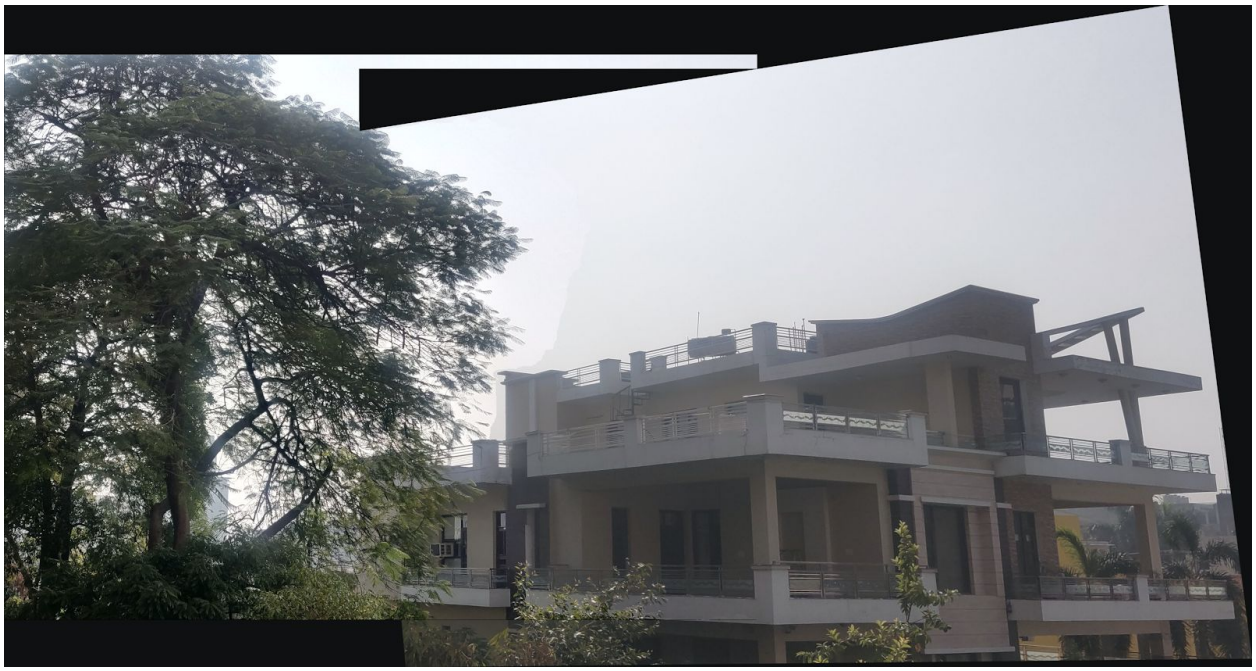
Mosaics generated on self clicked images (worst):

The mosaic creation failed in a few cases where the cuts found portray multiple objects for the same object (ghost), or cut the images from in between which makes the black background visible in the images.



In this image, the same lady was portrayed twice and the man at back was half cutted down.

In these images, black background is visible in the image in the centre which shows that the images are blended



In this image, the foot of the girl is visible below the black line and bushes near the tree shows that the images are stitched together and have some defects.

Conclusion:

I feel that the defects are due to the random min-cut picking. As the graph can have many min-cuts, it will pick the one and portray it. Hence it is possible that it will pick the cut in which the object gets cuts down in between. I have tried the code on various moving objects and in most cases, it gives the right result, but it also sometimes causes defects.

Drive link - https://drive.google.com/drive/folders/1QzYEiKepDZv5vkeuEDX7zge30ii-fhm_?usp=sharing