COMPUTER VISION
Assignment-2

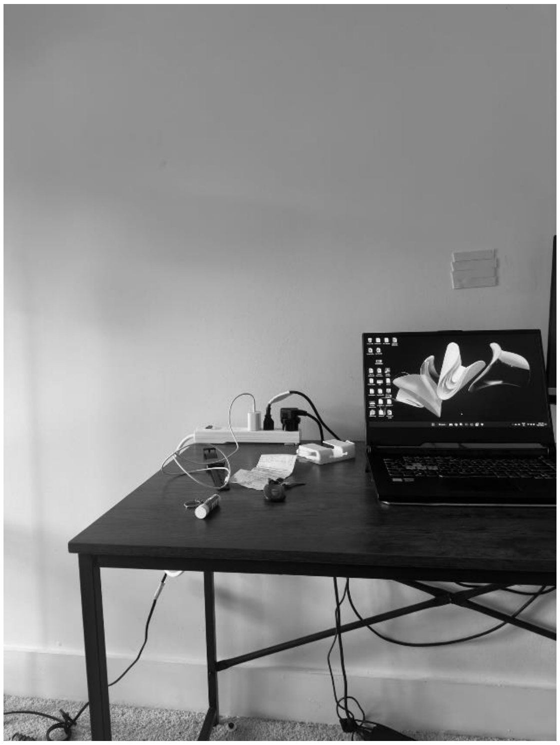
Name: Sai Harshith Kumar Akkenapelly **Panther ID:2704643**

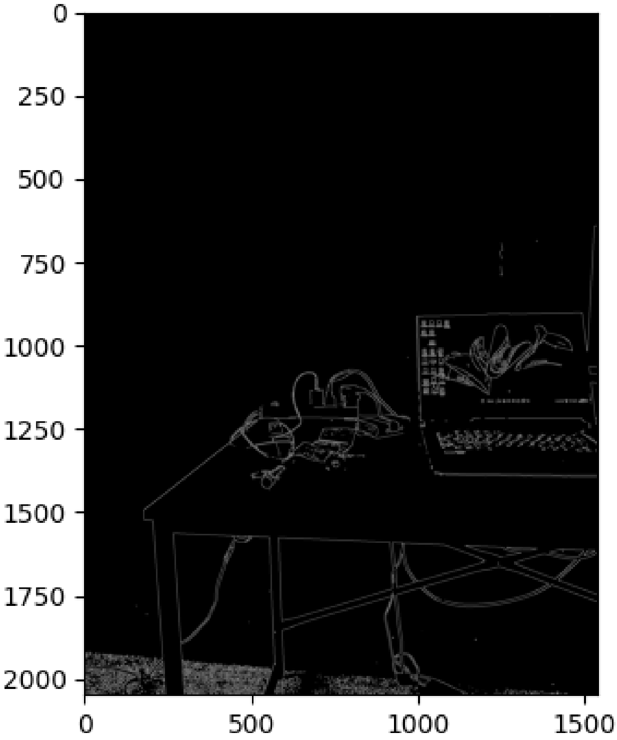
Question1:

Canny Detection:

With the use of the Canny edge detection technology, the amount of data that needs to be processed can be drastically reduced while still extracting meaningful structural information from various vision objects. It is frequently used in many computer vision systems.

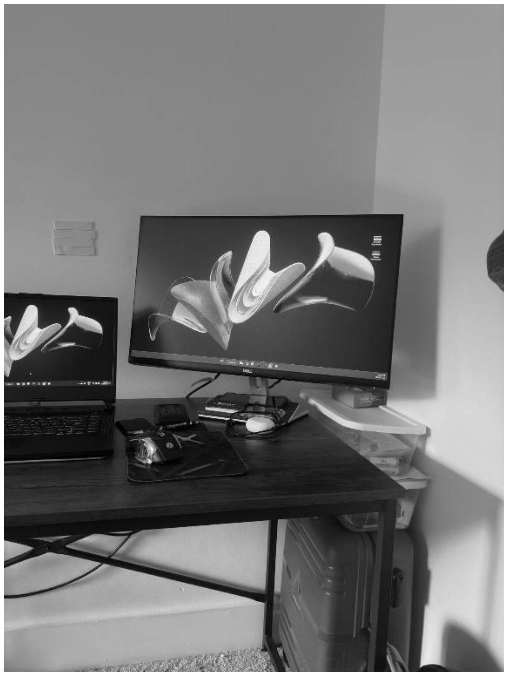
Consider the image given for canny edge detection





Harris Corner Detection:

The Harris corner detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image**.**
Taken a video and picked two images from the video and placed them in the folder. The harris.py script in the question2 folder and it takes the img1.jpg image, and initially it converts the image into float32 and then applies inbuilt
Harris corner detection. The output is written into img1c.png. Below is the output on execution.



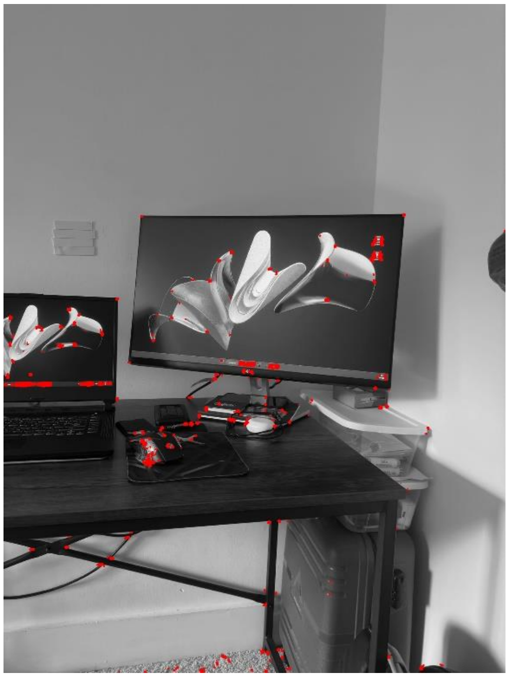


Image Coordinates Detection:

Another image img2C.png is taken and coordinators.py script is used to get 5 corners of the monitor object used in the image. The following is the output after execution.

```
[275,319],[796,313],[769,575],[251,535]
[651,446],[968,442],[996,591],[657,598]
```

Used this coordinates and run script homograpy.py to get output. One execution the below output is got.

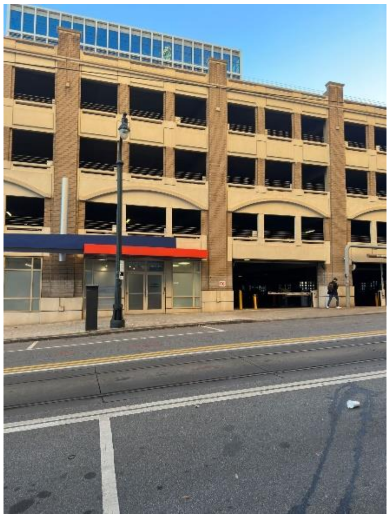
Homography Matrix:

```
[[ 1.08903895e+00 -5.10577799e-02  3.98168744e+02]
[ 2.05354431e-01  5.76273122e-01  2.26500848e+02]
[ 4.64837413e-04 -2.54493906e-04  1.00000000e+00]]
```

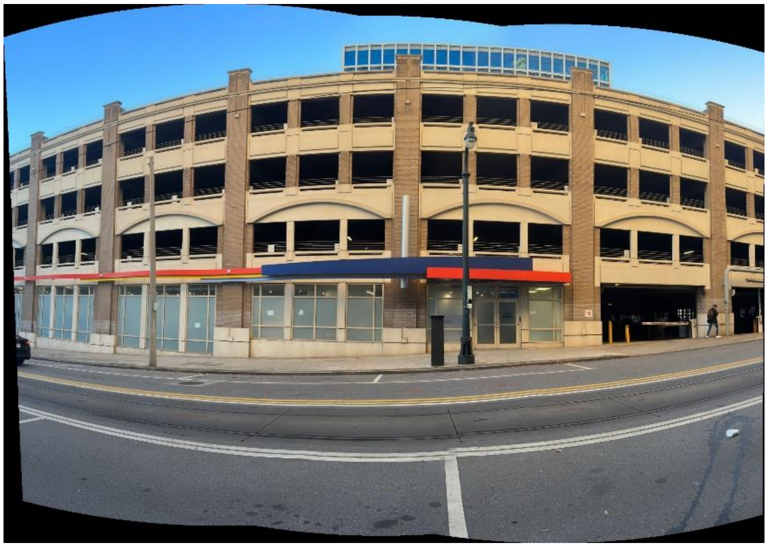
Question2:

Performed the image stitching operation on 5 sets of buildings/images with 3 images per set to form the stitching.

One of the dataset is as below



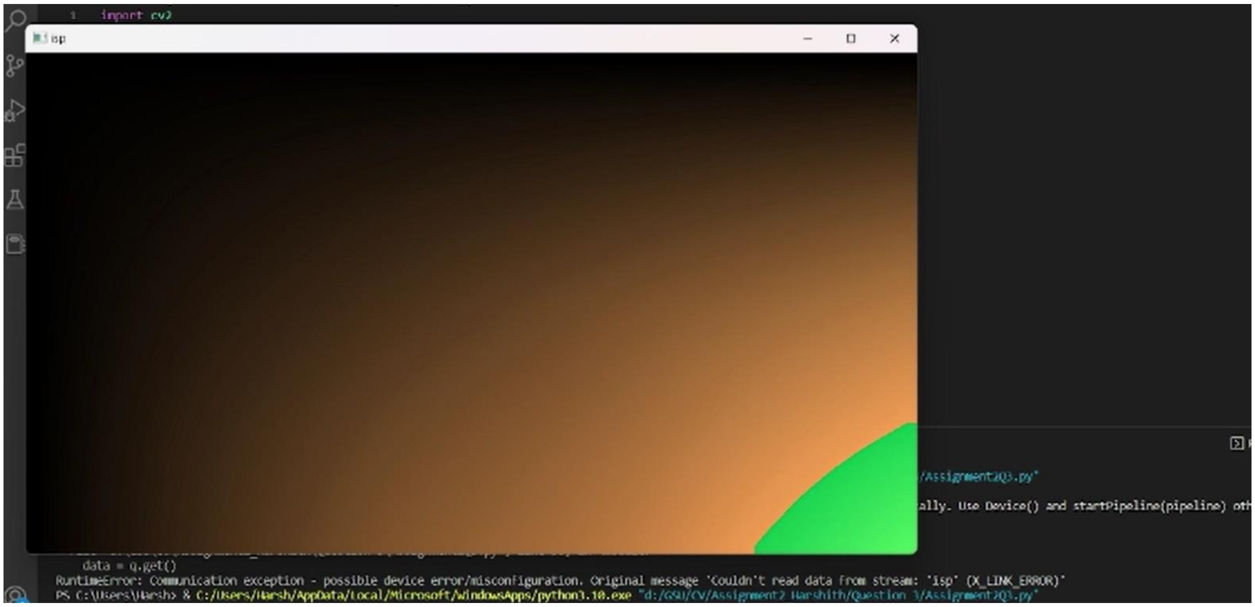
The result of image stitching for above is



Question3:

In question3 , the script **Assignment2Q3** file **integral image function** calculates the integral values of image which is custom written. The output has been saved and video recorded as a file imagefeed in the folder.

Sample output:



```
import cv2
```

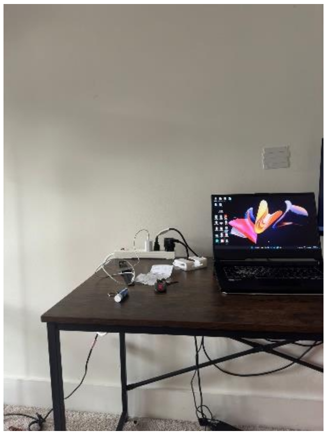
```
/Assignment2Q3.py"
ally. Use Device() and startPipeline(pipeline) oth
    data = q.get()
RuntimeError: Communication exception - possible device error/misconfiguration. Original message 'Couldn't read data from stream: 'isp' (X_LINK_ERROR)'
PS C:\Users\Harsh> & C:/Users/Harsh/AppData/Local/Microsoft/WindowsApps/python3.10.exe "d:/GSU/CV/Assignment2 Harshith/Question 3/Assignment2Q3.py"
```

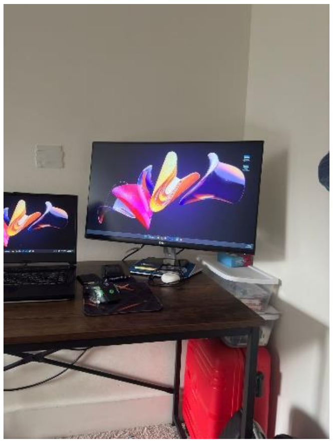
Question4:

Assignment2Q4 is the source code file used to execute task 4. Below are the images used as input to the file.

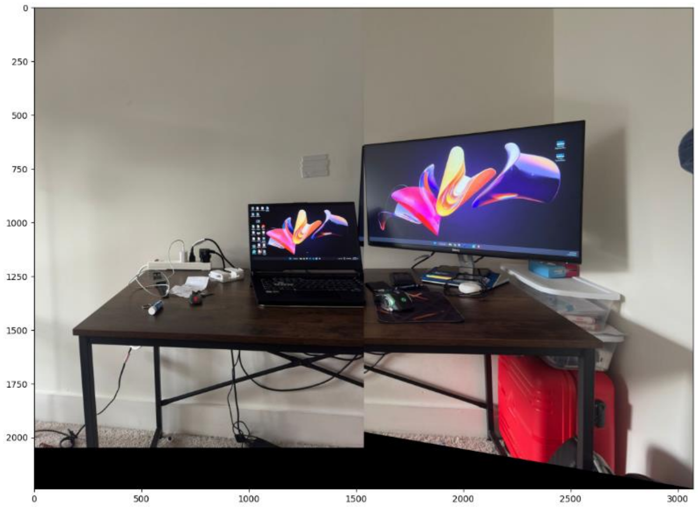
In the question4 folder, the script Assignment2Q4.py on execution gives the stitched image sift_Result.png. The main logic is to find the main feature points using the cv2 sift algorithm in Main_Ponts_Func(). Then, Match_The_Key_Points_Func() uses the KNN algorithm to find a nearest possible match between images. Then, use Find_Homography() method to find homography and use it to stitch the image using cv2.WarpPresepective() function to get the result image.

The output on execution is below.





The output file is as below:



Question5:

In question5 folderAssignment2Q5.py the script is same to sift script but the Main_Ponts_Func() uses cv2 orb feature detection.