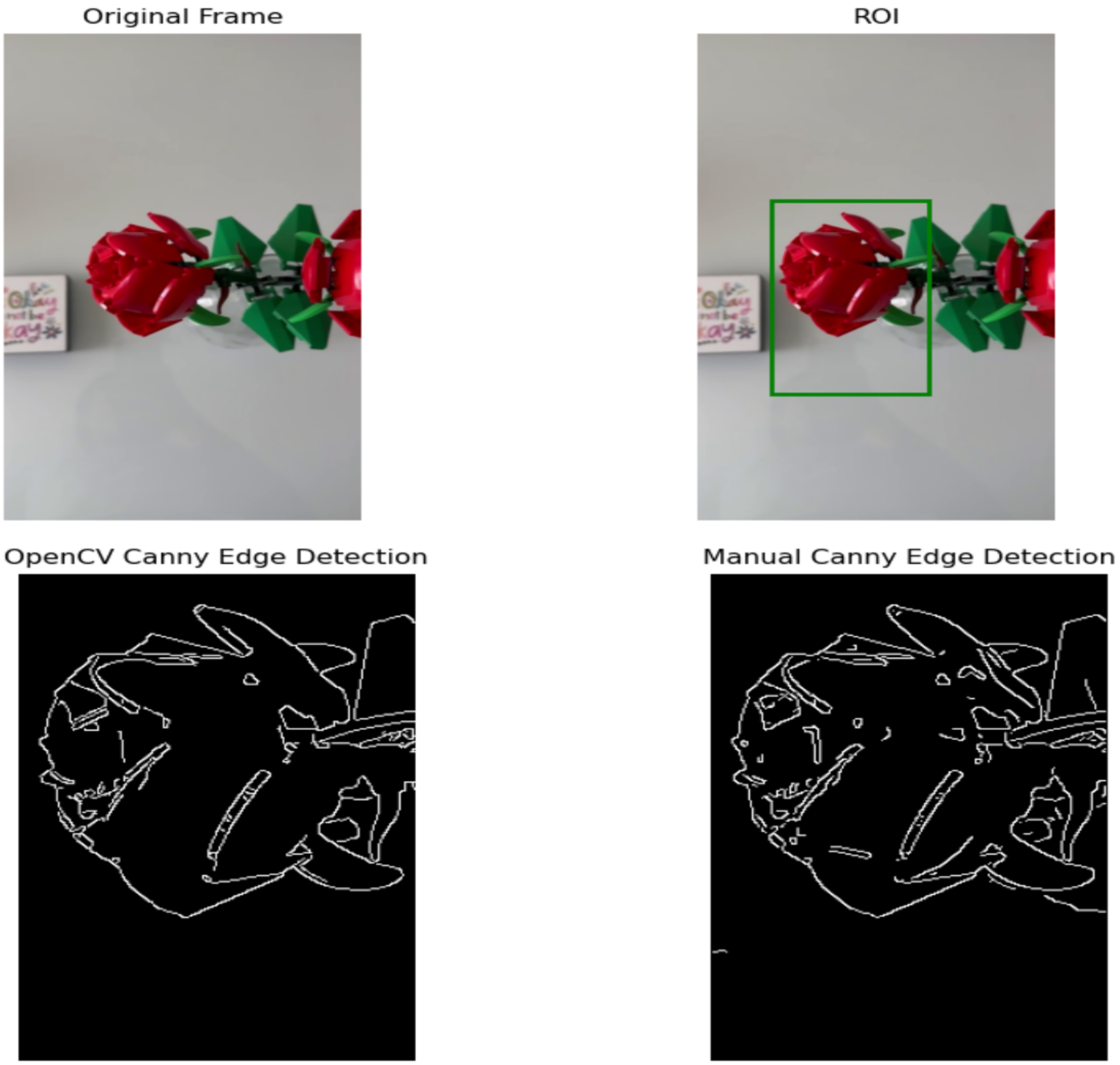
# CSC 8830: Computer Vision Assignment_2 Solutions

Capture a 10 sec video footage using a camera of your choice. The footage should be taken with the camera in hand, and you need to pan the camera slightly from left-right or right-left during the 10 sec duration. For all the images, operate at grayscale unless otherwise specified:

A 10 sec video was recorded from a mobile camera with 1080p resolution and 30FPS.

1.**Pick any image frame from the 10 sec video footage. Pick a region of interest in the image making sure there is an EDGE in that region. Pick a 5 x 5 image patch in that region that constitutes the edge. Perform the steps of CANNY EDGE DETECTION manually and note the pixels that correspond to the EDGE. Compare the outcome with MATLAB or OpenCV or DepthAI's Canny edge detection function.**



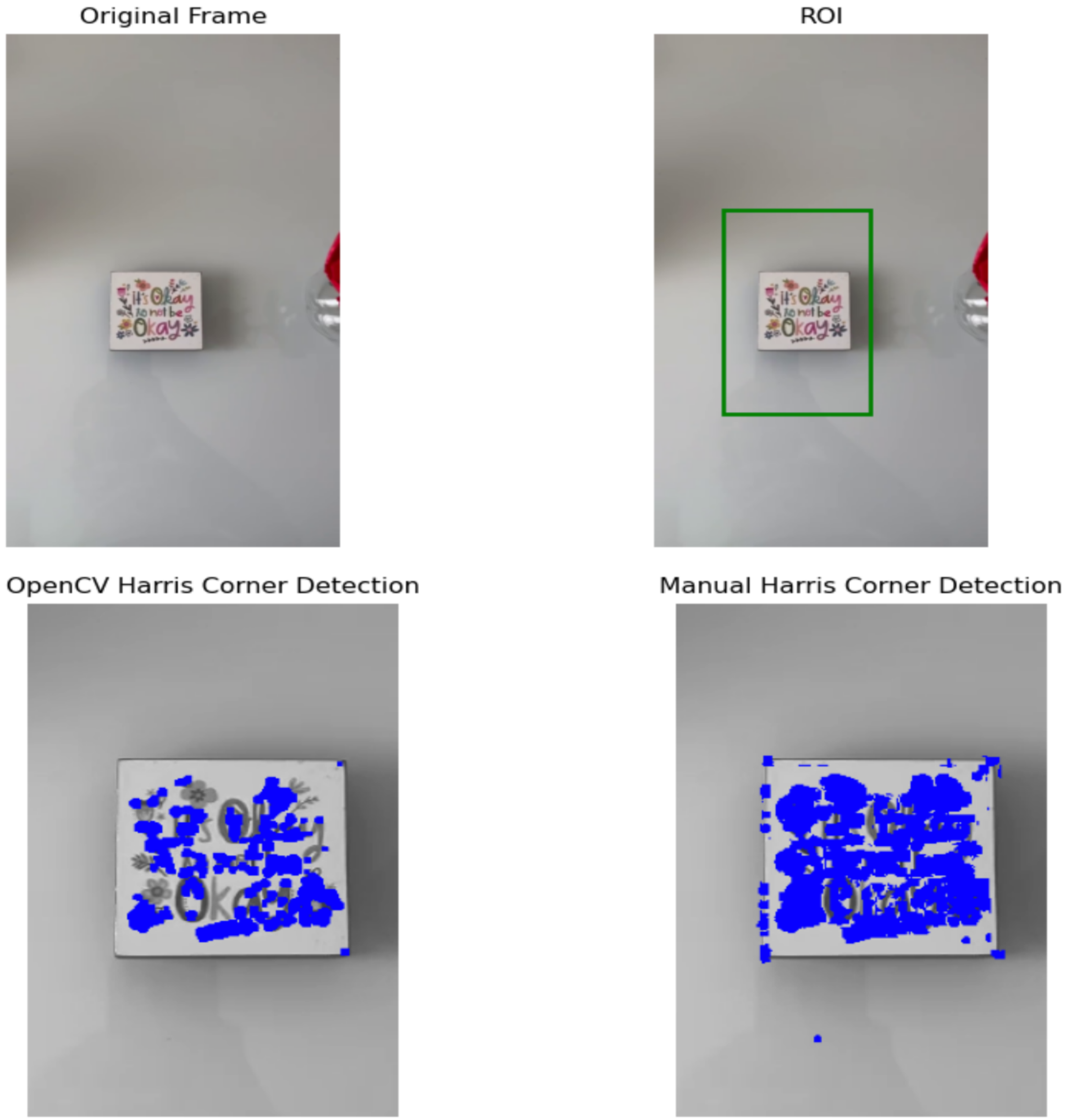
Original Frame

ROI

OpenCV Canny Edge Detection

Manual Canny Edge Detection

2. **Pick any image frame from the 10 sec video footage. Pick a region of interest in the image making sure there is a CORNER in that region. Pick a 5 x 5 image patch in that region that constitutes the edge. Perform the steps of HARRIS CORNER DETECTION manually and note the pixels that correspond to the CORNER. Compare the outcome with MATLAB or OpenCV or DepthAI's Harris corner detection function.**



Original Frame

ROI

OpenCV Harris Corner Detection

Manual Harris Corner Detection

**Consider an image pair from your footage where the images are separated by at least 2 seconds. Also ensure there is at least some overlap of scenes in the two images.**

3. **a. Pick a pixel (super-pixel patch as discussed in class) on image 1 and a corresponding pixel ((super-pixel patch as discussed in class)) on image 2 (the pixel on image 2 that corresponds to the same object area on image 1). Compute the SIFT feature for each of these 2 patches. Compute the sum of squared difference (SSD) value between the SIFT vector for these two pixels. Use MATLAB or Python or C++ implementation -- The MATLAB code for SIFT feature extraction and matching can be downloaded from here: https://www.cs.ubc.ca/~lowe/keypoints/ (Please first read the ReadMe document in the folder to find instructions to execute the code).**

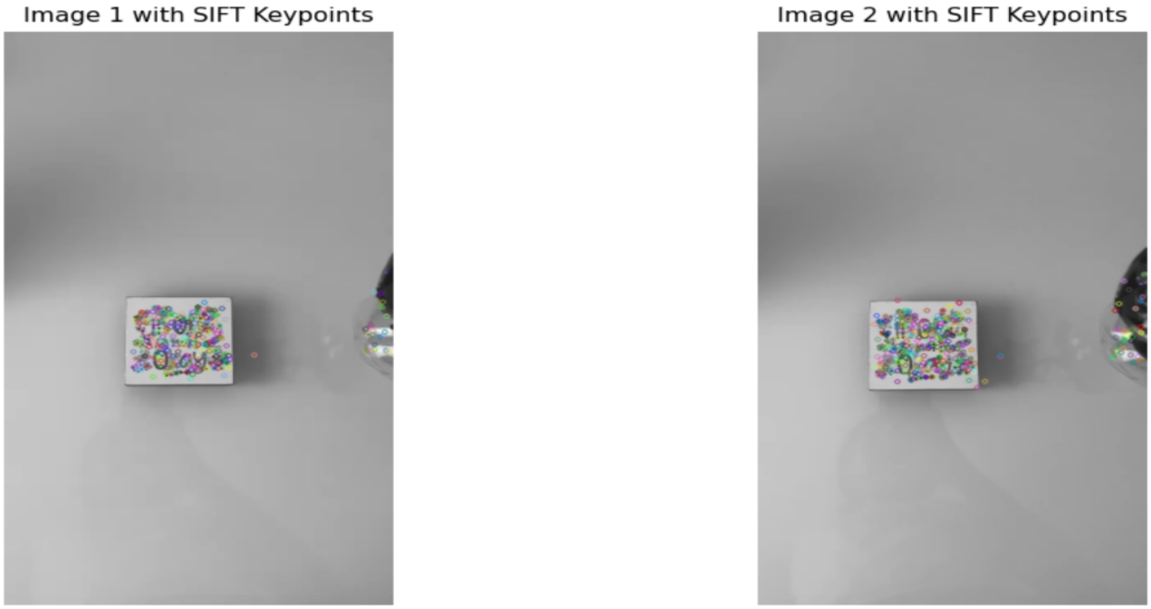
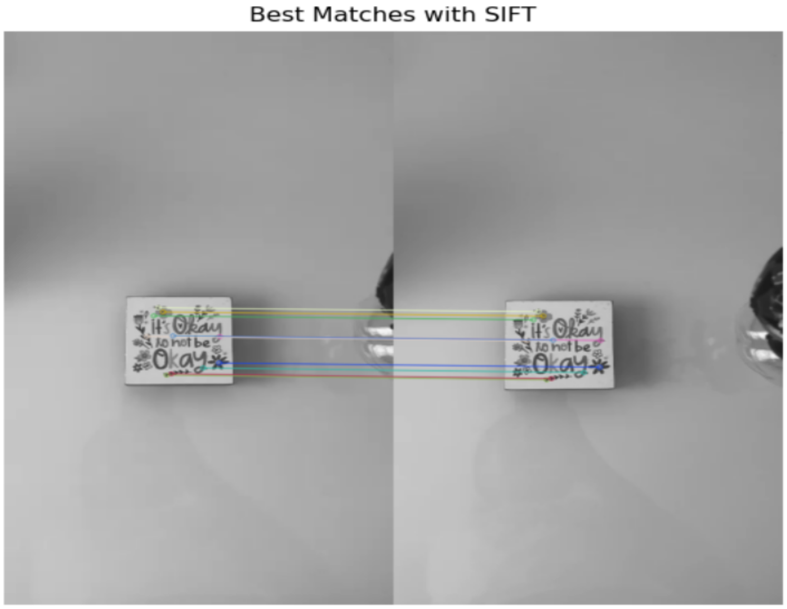


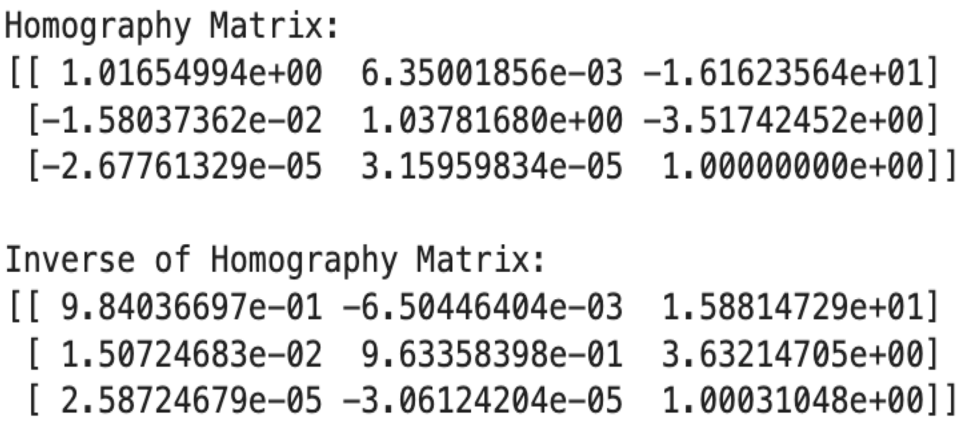
Image 1 with SIFT Keypoints

Image 2 with SIFT Keypoints

```
SSD between two SIFT vectors is  645.0
```

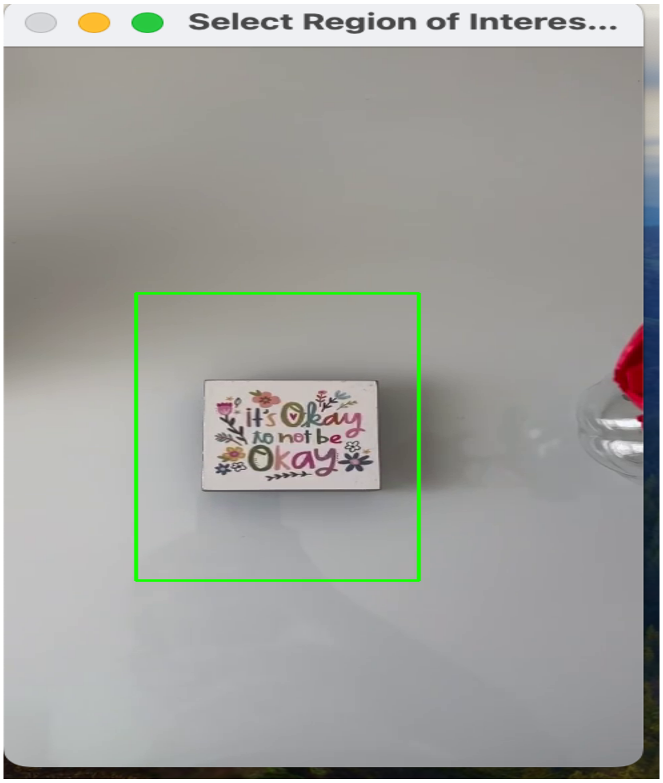

Best Matches with SIFT

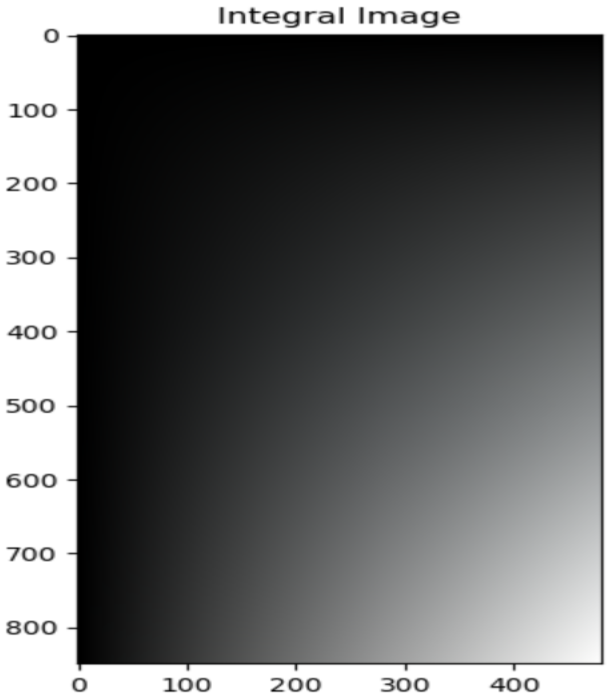
**b. Compute the Homography matrix between these two images using MATLAB or Python or C++ implementation. Compute its inverse.**



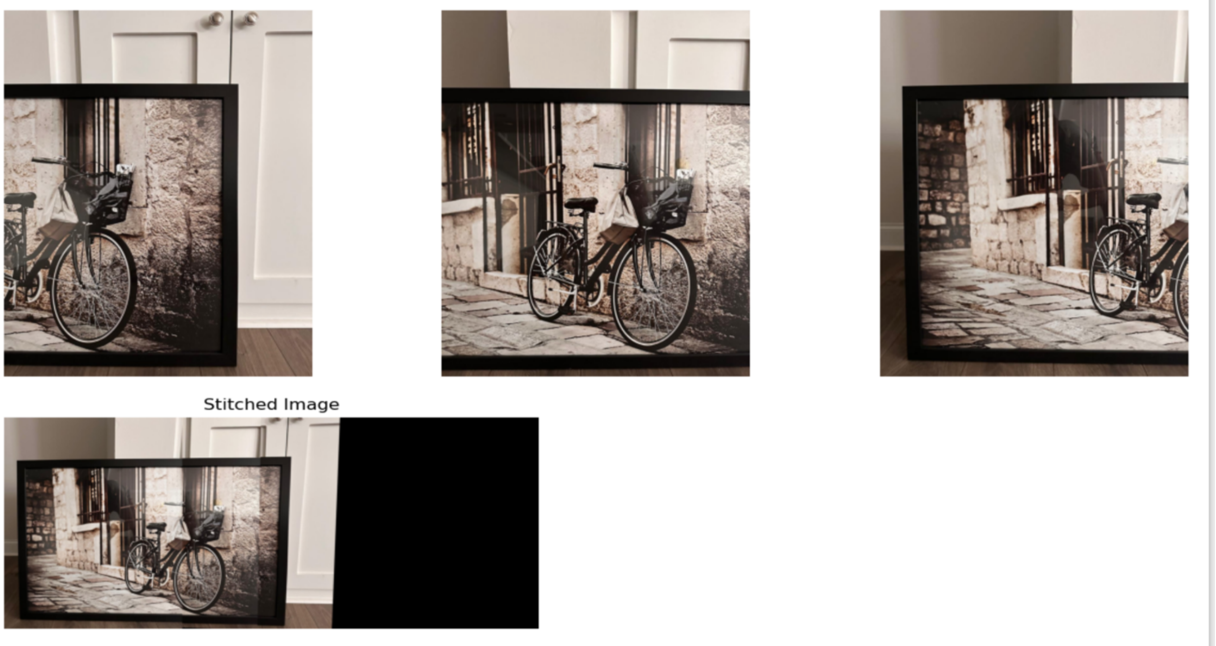
```
Homography Matrix:
[[ 1.01654994e+00  6.35001856e-03 -1.61623564e+01]
 [-1.58037362e-02  1.03781680e+00 -3.51742452e+00]
 [-2.67761329e-05  3.15959834e-05  1.00000000e+00]]

Inverse of Homography Matrix:
[[ 9.84036697e-01 -6.50446404e-03  1.58814729e+01]
 [ 1.50724683e-02  9.63358398e-01  3.63214705e+00]
 [ 2.58724679e-05 -3.06124204e-05  1.00031048e+00]]
```

4. **Implement an application that will compute and display the INTEGRAL image feed along with the RGB feed. You cannot use a built-in function such as "output = integral_image(input)"**

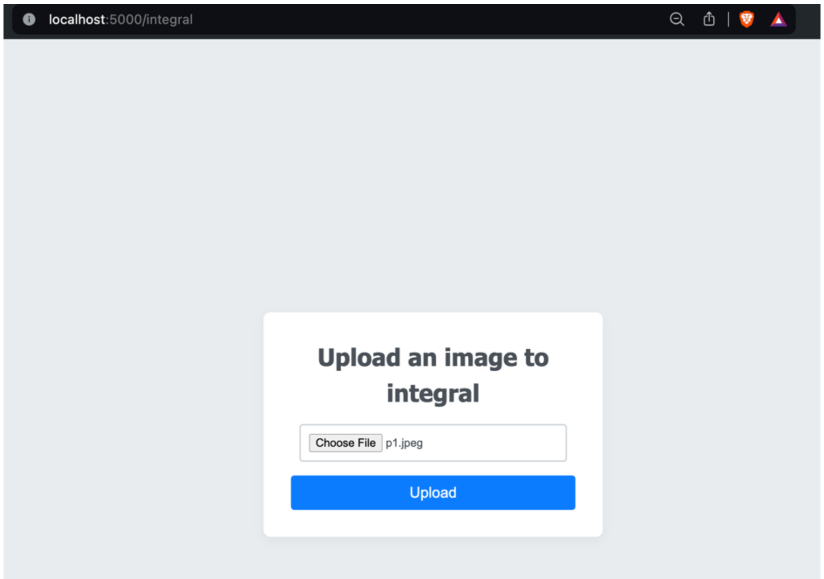


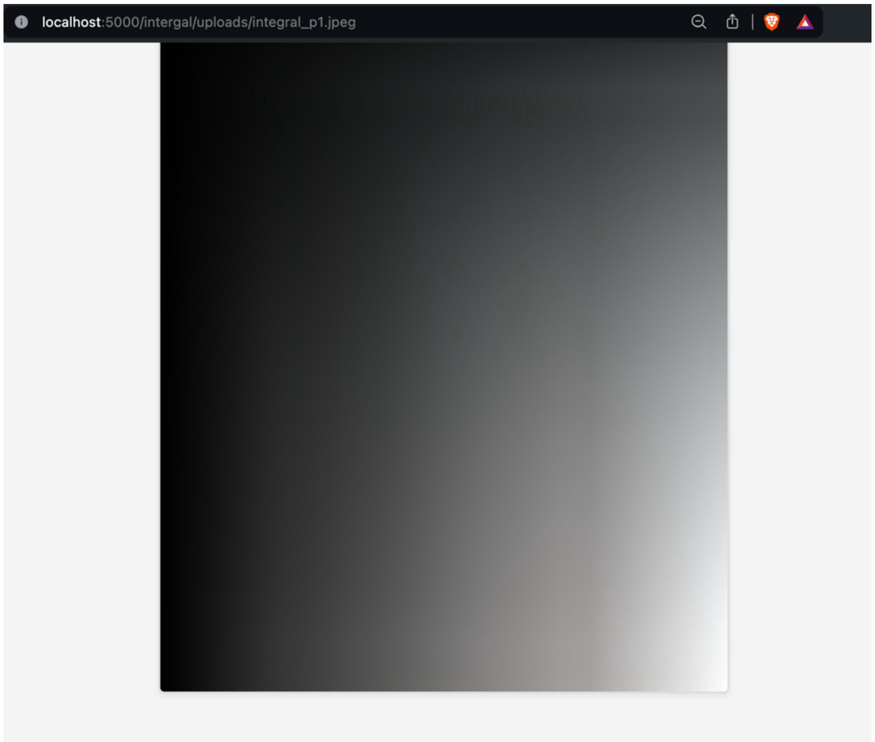


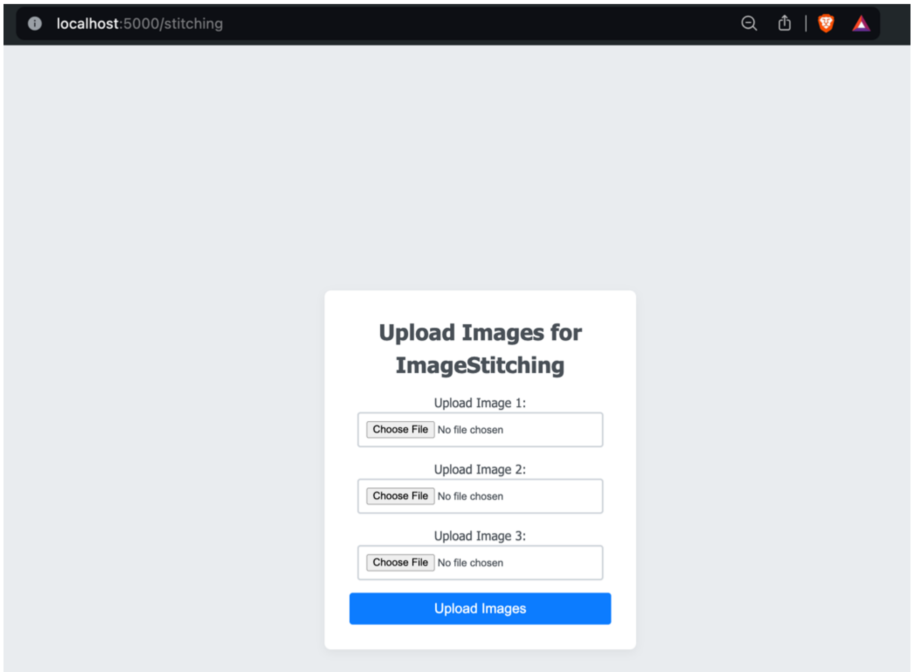
5. **Implement the image stitching for a 360 degree panoramic output. This should function in real-time. You can use any type of features. You can use built-in libraries/tools provided by OpenCV or DepthAI API. You cannot use any built-in function that does output = image_stitch(image1, image2). You are supposed to implement the image_stitch() function**



### 6. Integrate the applications developed for problems 4 and 5 with the web application developed in Assignment 1 problem 4*






localhost:5000/stitching
Upload Images for ImageStitching
Upload Image 1:
Choose File No file chosen
Upload Image 2:
Choose File No file chosen
Upload Image 3:
Choose File No file chosen
Upload Images

## Github Link:

[https://github.com/harshinijaini/ComputerVision-CSC8830/tree/main/Harsh_Assignment_2](https://github.com/harshinijaini/ComputerVision-CSC8830/tree/main/Harsh_Assignment_2)