CS5330 Project-4 Report

A short description of the overall project in your own words. (200 words or less)

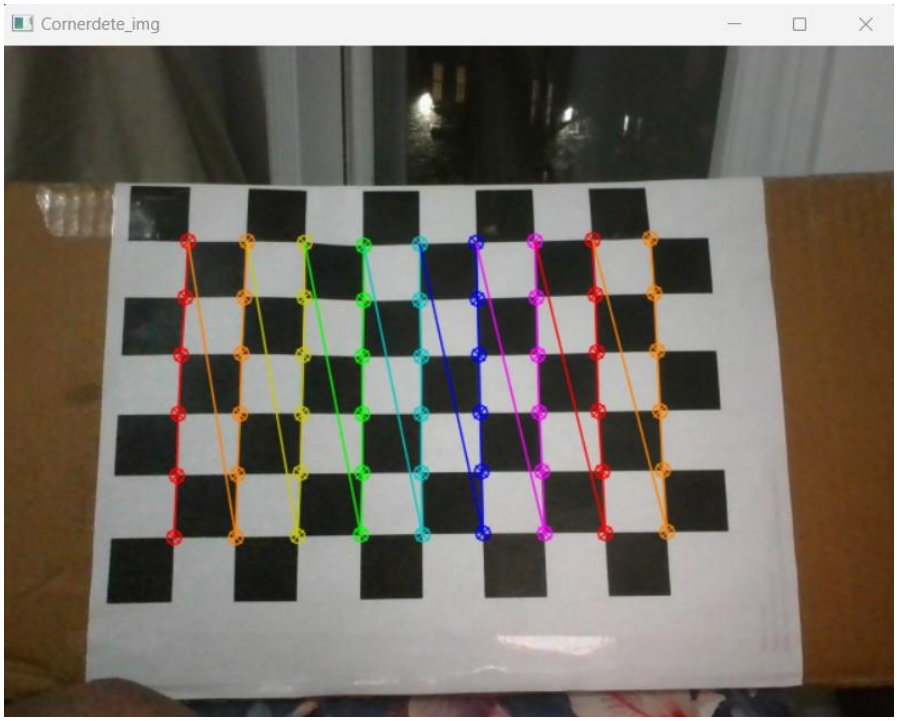
Project Description:

This project involved camera calibration and using it to place virtual objects. The target is a checkerboard pattern and the first task is to detect and extract target corners using the “findChessboardCorners" function. The next step is to let the user specify an image for calibration and save the corner locations and corresponding 3D world points. Using cv::calibrateCamera calibration,camera_matrix and distortion_coefficients are printed before and after the calibration, along with the final reprojection error. The virtual object is placed over the scene relative to the target(in this case checkerboard) that moves and orients itself based on the motion of the camera/target. The point_set of 3D positions will remain the same, regardless of the orientation of the checkerboard (target).

Any required images along with a short description of the meaning of the image.

Image Calibration:

Chessboard calibration:



Before and after Calibration Camera matrix, Distortion coefficients and Re-projection error:

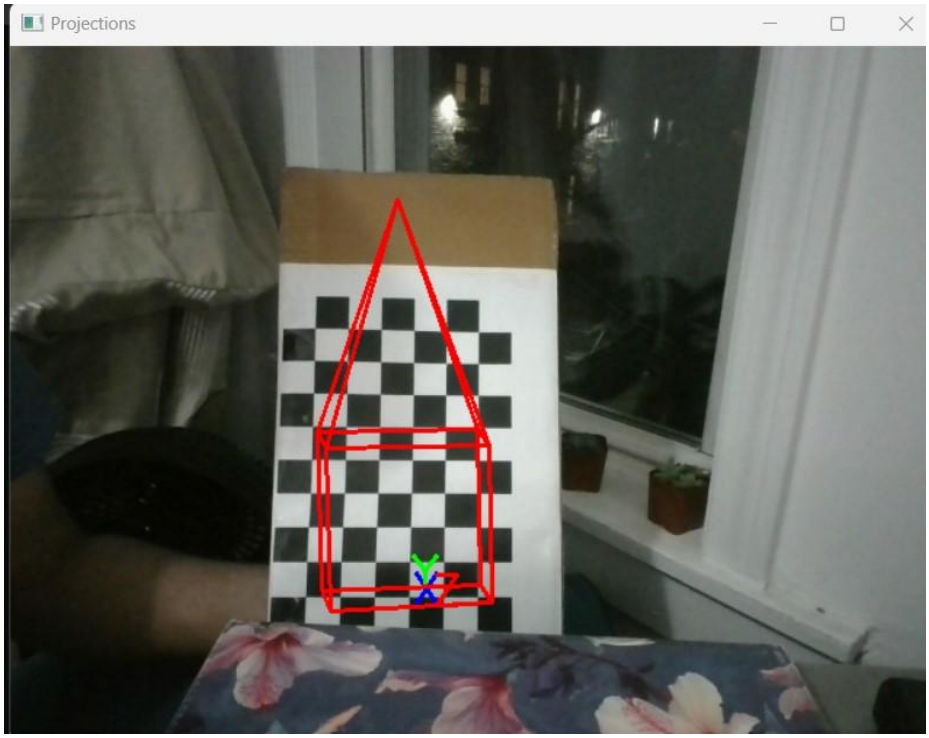
```
camerametr [1, 0, 640;
 0, 1, 480;
 0, 0, 1]

Camera matrix[6233.789202391408, 0, 319.5;
 0, 6233.789202391408, 239.5;
 0, 0, 1]
Distortion coefficients[-178.5602862988513, 146356.2511671353, 0, 0, 0]
Reprojection error: 0.431173
```

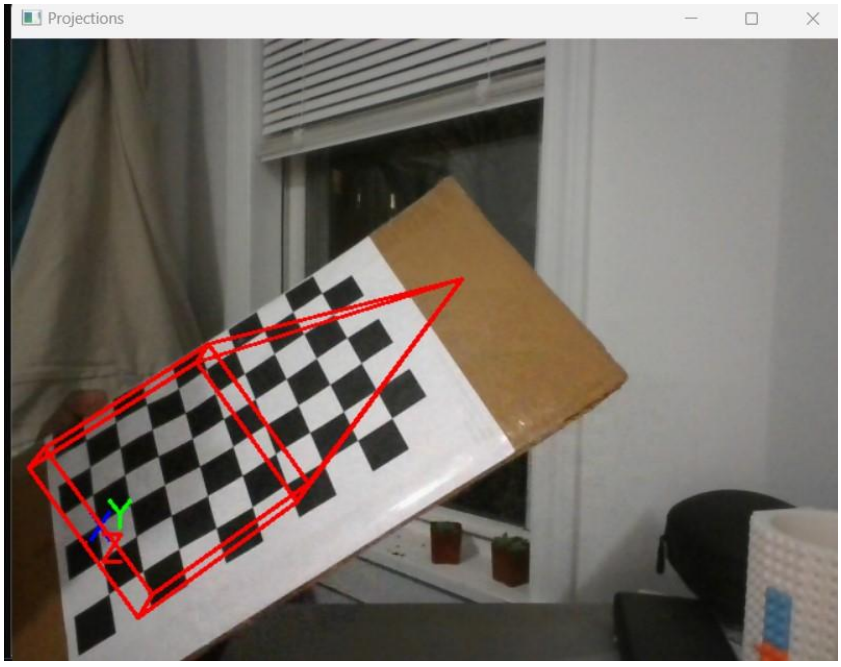
```
Translation vector: [-4.567945927614 87;
 11.29590172452086;
 1452.276605169491]
Rotation vector: [2.113699907687004;
 -0.1490270658584726;
 -2.242797538291622]
Translation vector: [-4.5739821045664 14;
 11.26459813246229;
 1452.439122929088]
Rotation vector: [2.113121181189188;
 -0.1482155222356482;
 -2.241936239283512]
Translation vector: [-4.572483096237 55;
 11.20352321074726;
 1451.160947486331]
Rotation vector: [2.110834260888668;
 -0.1390460269729248;
 -2.237288516013572]
Translation vector: [-4.599399412120476;
 11.12677442380689;
 1439.483691602292]
Rotation vector: [-2.096051704985547;
 0.1031094391057821;
 -2.23176808386319]
```

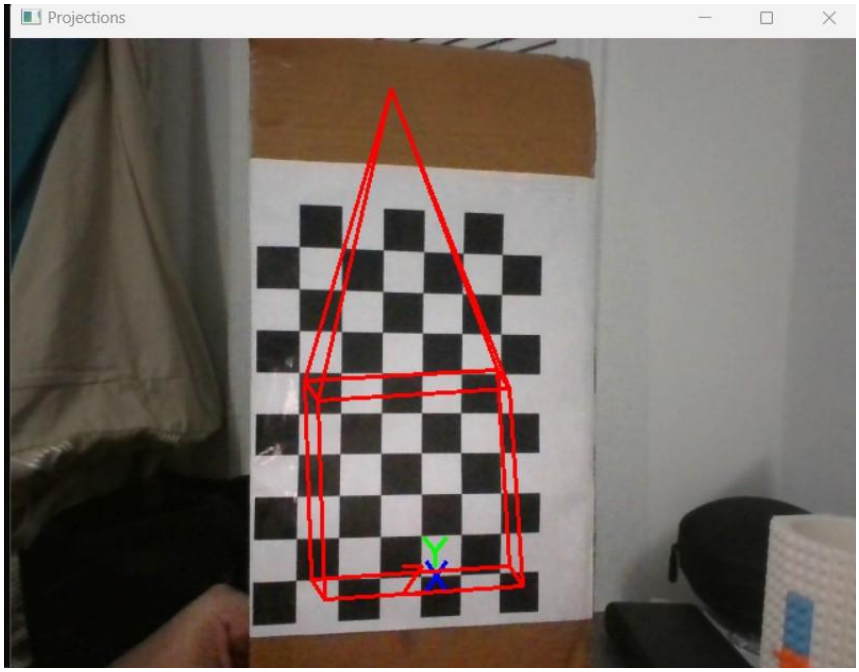
Virtual object projection:

We have created a pyramid over cube structure which is projected on a chess board



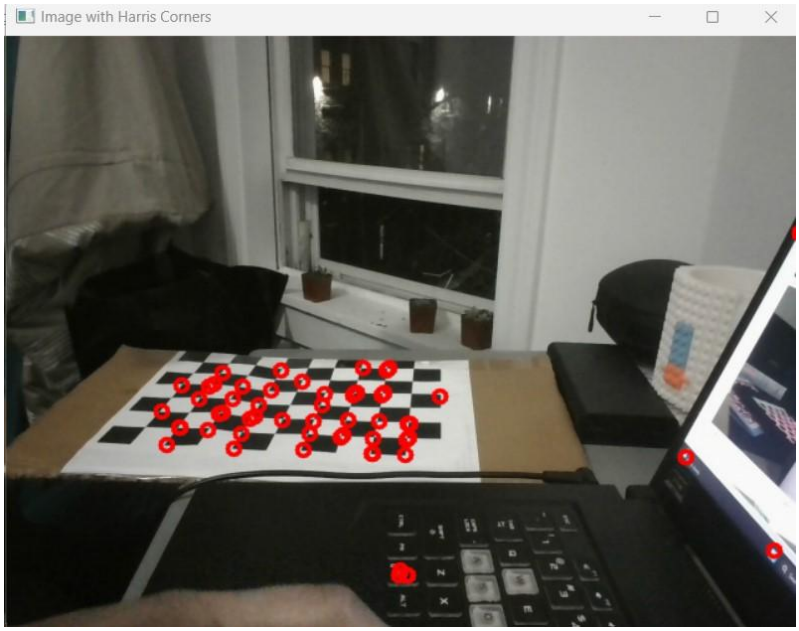
The target orienting according the chess board orientation:

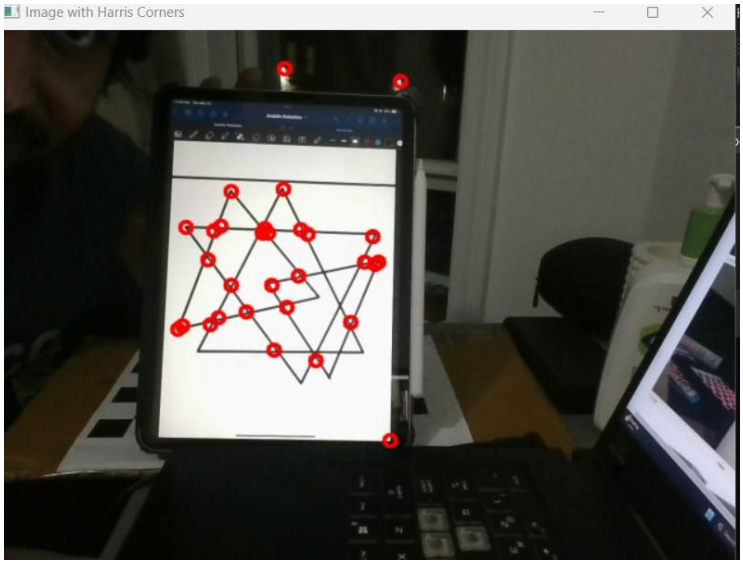




Robust Features:

Using Harris Corners Detector the corners are detected in the chess board and a geometrical pattern. Just the way we used chessboard corners as the basis for putting augmented reality, we first need to find the correlations between the 2D coordination in the image and the 3D coordination in the real world and that can be done once we have features extracted and the features extracted using harris features is a good example for it because lot of good number of features are obtained as observable from below phoots. Once we are able to project the points in the image to a 3D coordinate system, we can calibrate the camera and then get the camera position through which we can project any virtual object.

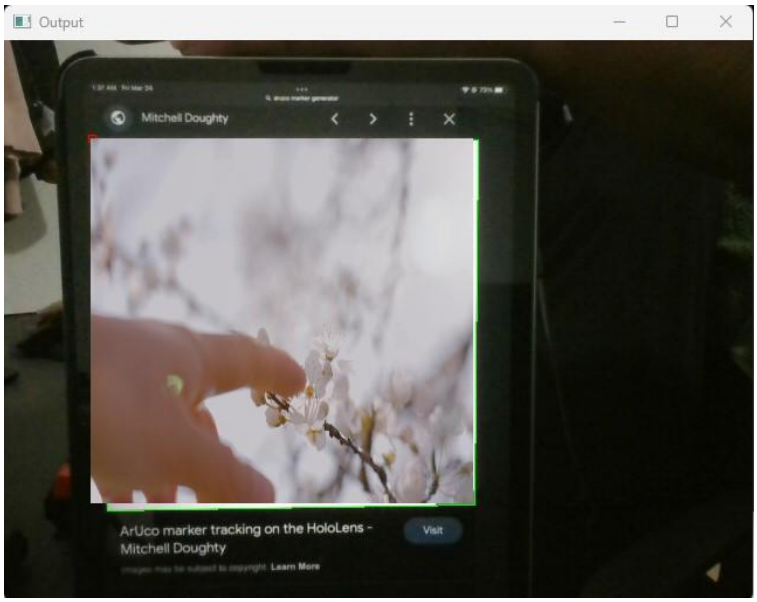


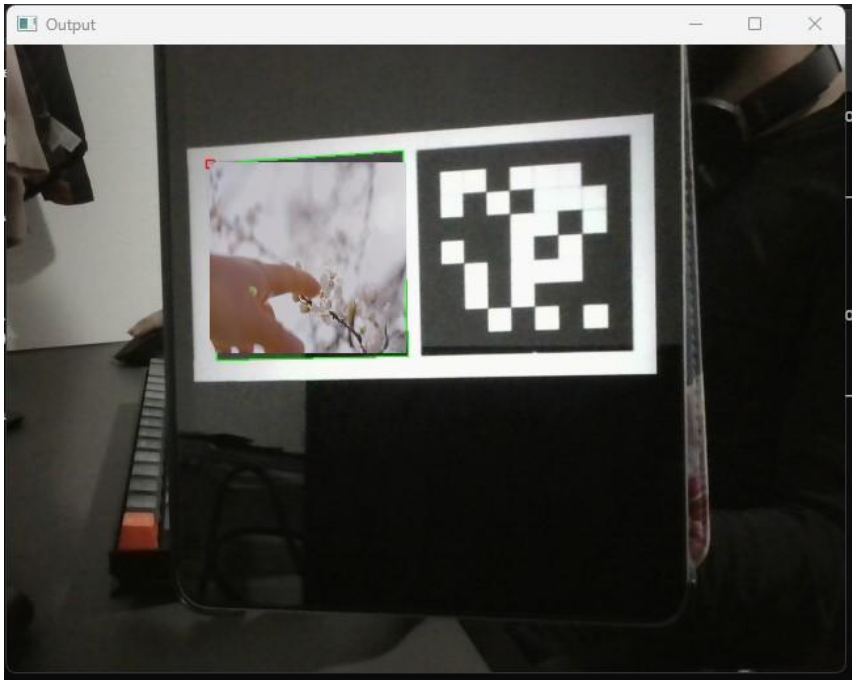


A description and example images of any extensions.

Extensions:

- **Check out the ArUco Links to an external site. markers in the aruco module of OpenCV. Integrate these into your system.**

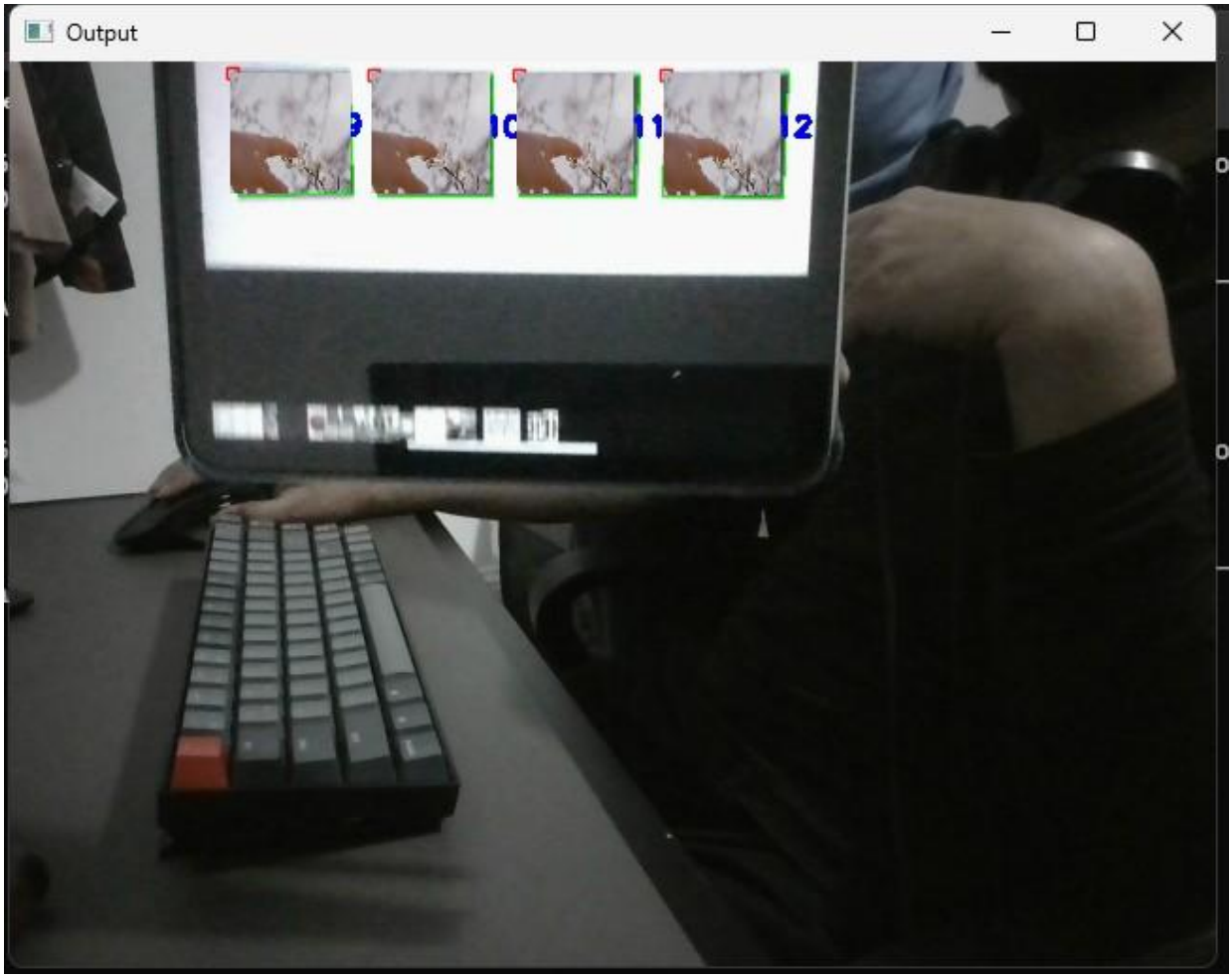




The system was designed to detect aruco targets and project the photo/pic on the Aruco marks.

- **Get your system working with multiple targets in the scene.**

  Not just a single target if multiple aruco marks are present the system can project the same picture on multiple aruco marks.
  The below is the aruco marks taken and the corresponding the picture is projected on all the aruco marks.



A short reflection of what we have learned:

We were able to understand various concepts related to camera calibration like intrinsic parameters, camera matrix, projection error and have had the opportunity to get hands-on with it. Got an idea on correlating the 2D coordination in images and 3D coordination in the real-world. Understood how Augmented reality works to an extent.

Acknowledgement of any materials or people you consulted for the assignment:

1. https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html
2. https://docs.opencv.org/4.x/d2/d1d/samples_2cpp_2lkdemo_8cpp-example.html#a21
3. https://learnopencv.com/camera-calibration-using-opencv/
4. https://www.uco.es/investiga/grupos/ava/node/69
5. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
6. https://learnopencv.com/homography-examples-using-opencv-python-c/