CS 408 - Individual Project

Project Report - Review on Harris Corner Detection

By T.S.N Dasa (S/18/335)

1.Introduction

Harris corner Detection method was first invented by Chris Harris and Mike Stephens in 1988 [1] by upgrading the existing corner detection method.This corner detection method can be used in many application areas such as object recognition, image stitching, and motion tracking.

If we can identify corners it is so useful in many cases of object detection.This method is based on the identify the corners **if there is corner then the image then there is an intensity difference for two sides .**

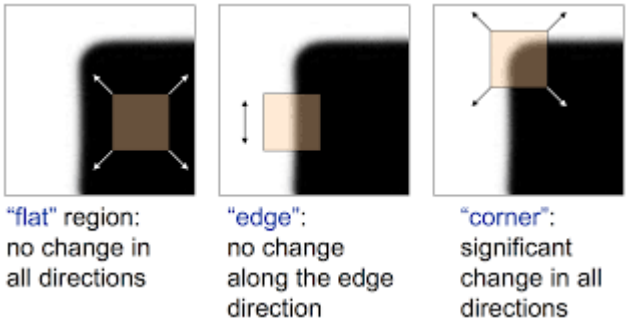


Figure 2.3: *Definition for corner of the images*

2.Dataset

As for the example cases we have decided to show it using 3 types of images which Can be

Figure 2.3: *Image1 ,image5, chessboard image , image 2,image 3,image 4(from upper left to bottom right)*

3.Methodology

We use these main factors in the methodology to the detection.

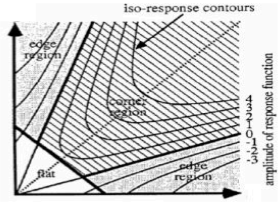
- Image gradients: Compute gradients using Sobel operators.
- Structure tensor: Construct the structure tensor matrix M from the gradients.
- Harris response: Derive the Harris response R using the eigenvalues of the matrix M.

3.1 Harris response function

Measure of corner response:

$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$



(k - empirical constant, k = 0.04-0.06)

where k is a sensitivity factor, usually between 0.04 and 0.06.

3.2 Corner criteria

Describe how corners are identified based on the Harris response function. Corners are regions where the response R is above a certain threshold and exhibits local maxima.

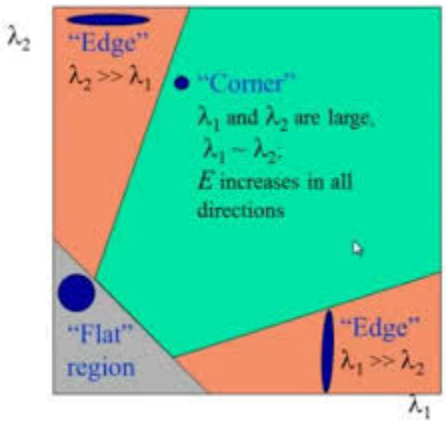


Figure 2.2: *Corner detection criteria*

4.Implementation

In the implementation of the code we can implement it by the the build the code from the scratch using the

4.1 Loading the Image

The image is loaded using `cv2.imread` and converted to grayscale using `cv2.cvtColor`.

4.2 Gradient Computation

Gradients in the x and y directions are computed using the Sobel operator.

4.3 Structure Tensor Components

The components of the structure tensor Ix2I_x^2Ix2, Iy2I_y^2Iy2, and IxIyI_x I_yIxIy are computed.

4.4Gaussian Filtering

These components are smoothed using a Gaussian filter to reduce noise.

4.5 Harris Response Calculation

The Harris response is calculated using the determinant and trace of the structure tensor.

4.6 Thresholding and Corner Detection

Corners are identified by applying a threshold to the Harris response and finding local maxima.

4.7 Displaying Results

The detected corners are plotted on the original image using `matplotlib`.

```python
def harris_corner_detection(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    Ix = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
    Iy = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
    Ix2 = Ix**2
    Iy2 = Iy**2
    Ixy = Ix*Iy
    Ix2 = cv2.GaussianBlur(Ix2, (3, 3), 1)
    Iy2 = cv2.GaussianBlur(Iy2, (3, 3), 1)
    Ixy = cv2.GaussianBlur(Ixy, (3, 3), 1)
    k = 0.04
    detM = Ix2 * Iy2 - Ixy**2
    traceM = Ix2 + Iy2
    R = detM - k * traceM**2
    R = cv2.dilate(R, None)
    threshold = 0.01 * R.max()
    corners = np.zeros_like(R)
    corners[R > threshold] = 255
    return corners
```

Figure 4.1: *code from the scratch*

But in here we used the simple implementation with the in build function in opencv

```python
# Apply Harris Corner Detection
dst = cv2.cornerHarris(gray, blockSize=2, ksize=5, k=0.04)
```

Figure 4.2: *opencv harris corner function*

Since with the in-built method we can get better results ,this is mostly used in getting the results.

Here is the google drive link for the implementation of the method with the different result that have been obtained
https://colab.research.google.com/drive/1UpUH_y0uy8M2gL_KHtS3C4G6pAolf2l4#scrollTo=J7oRfgeUBabj

5.Result

Here are the results obtained by the implementation

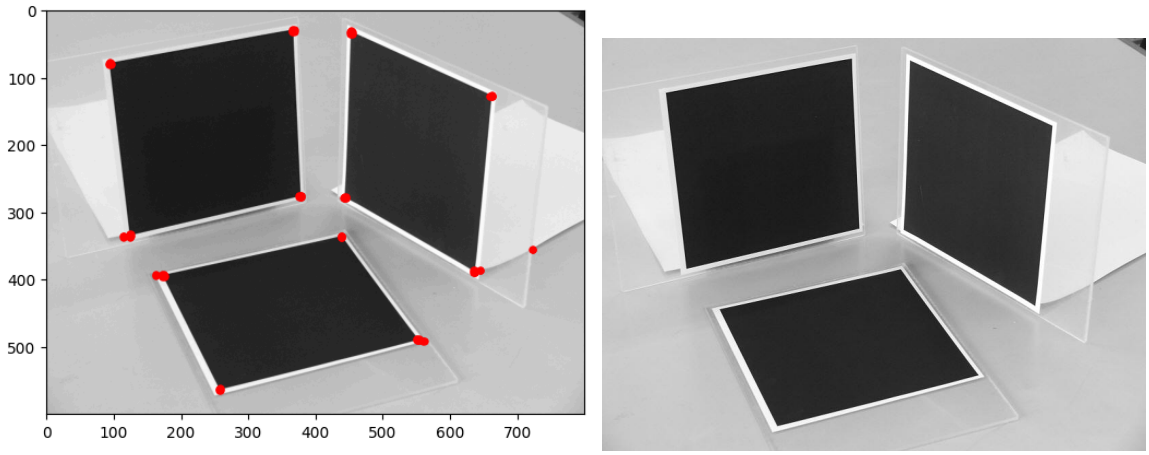


Figure 5.1: *Image corner detection*

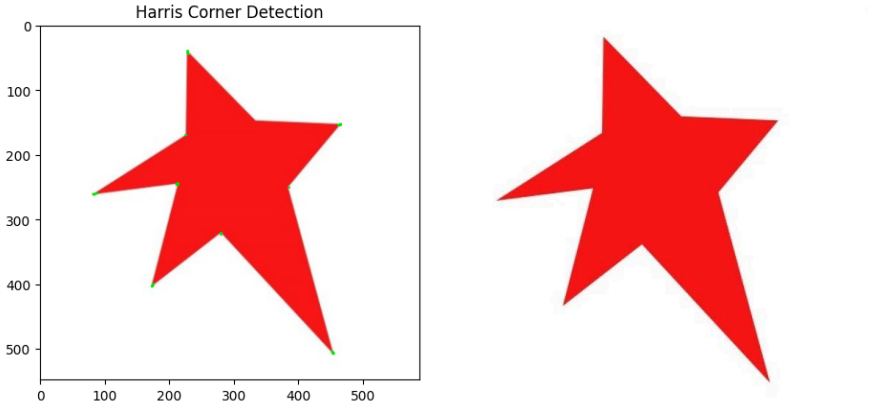


Figure 5.2: *Image corner detection* for star

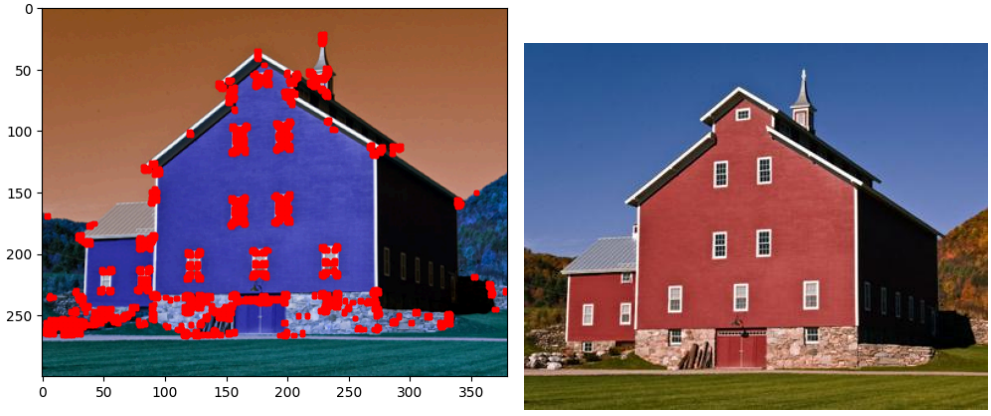


Figure 5.3: *Image corner detection* with K size = 3

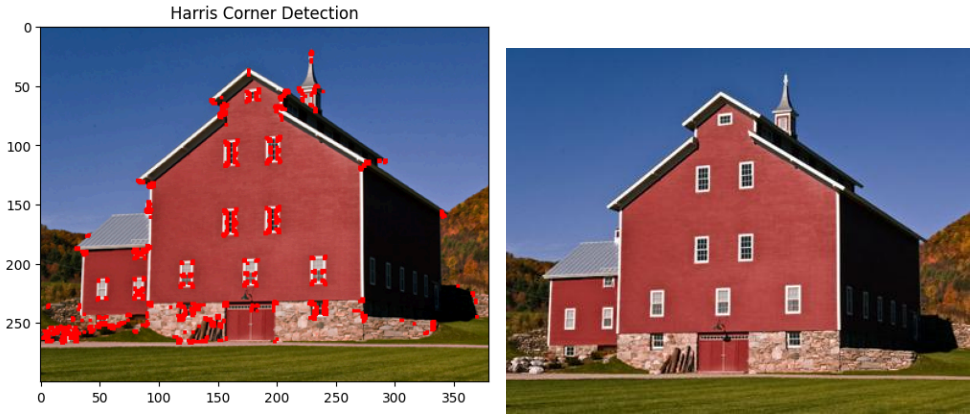


Figure 5.4: *Image corner detection* with Ksize = 5

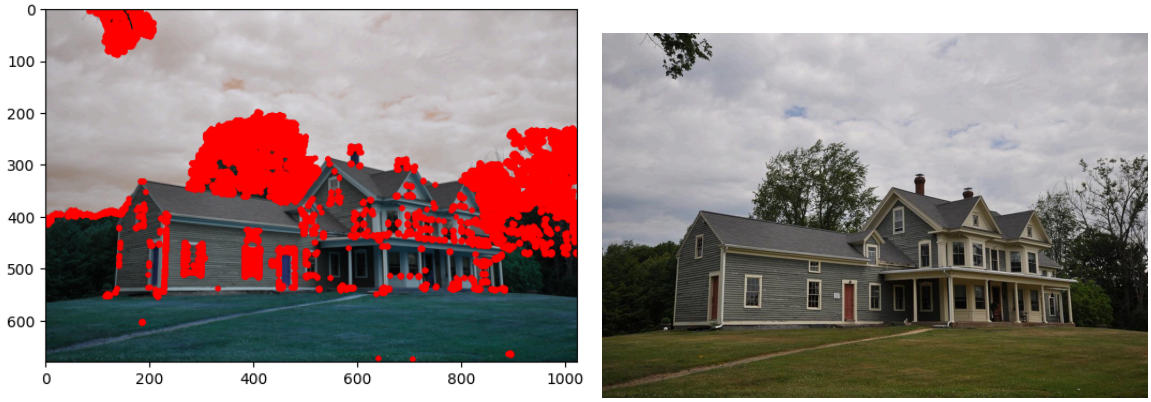


Figure 5.5: *Image corner detection* for Ksize =5

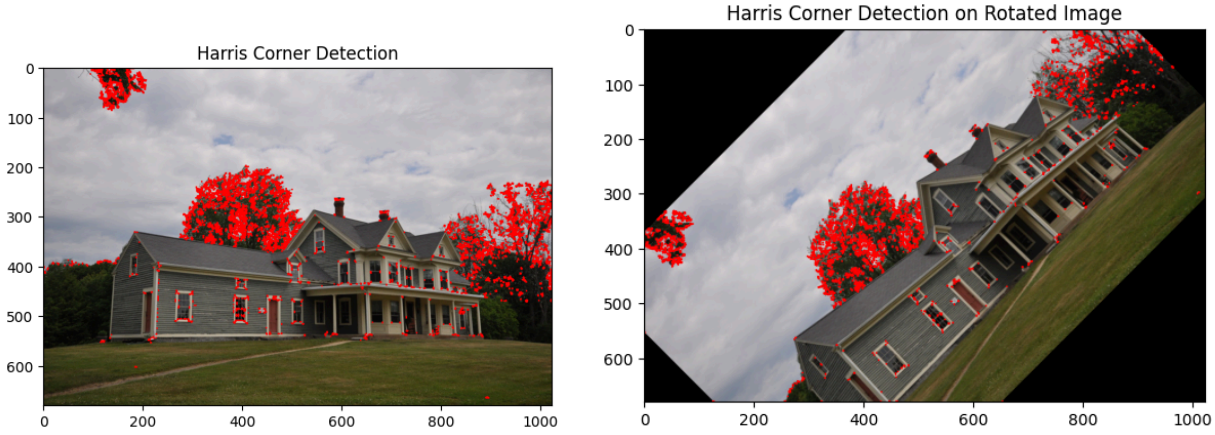


Figure 5.6: *Image corner detection comparison with rotation* K size =5

Figure 5.7: *Image corner detection* with K size =5

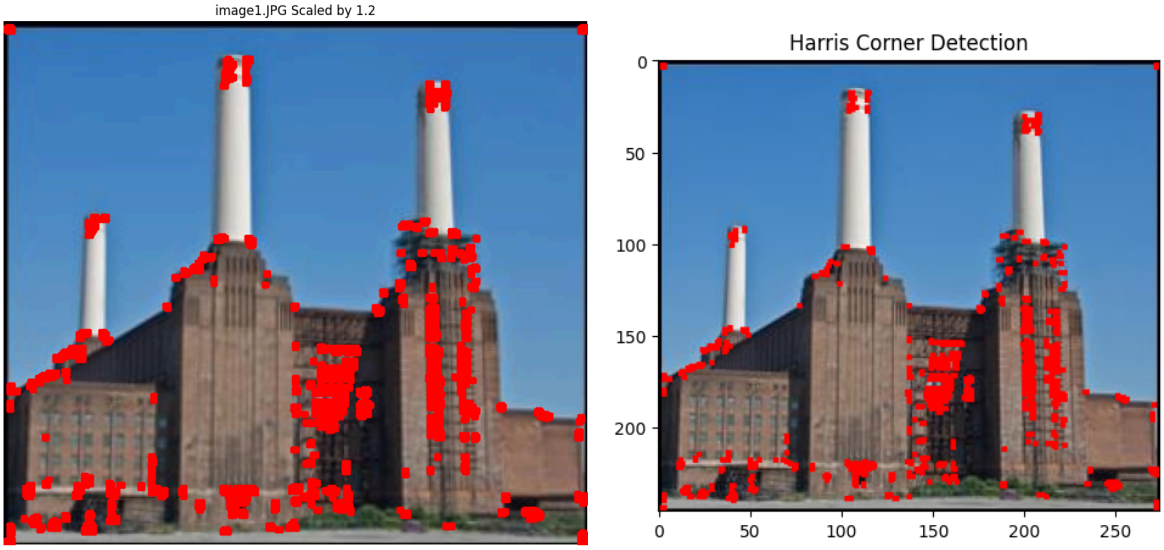


Figure 5.8: *Image corner detection* with K size =5 and the scaling by 1.2

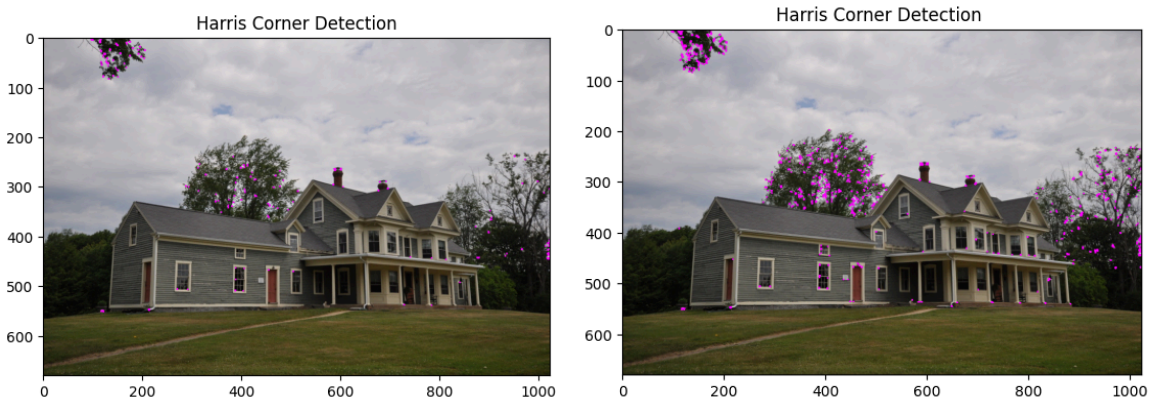


Figure 5.9: *Image corner detection* with K size =5 and parameter = 10 and 8

Figure 5.11: *Image corner detection* for Ksize 11 and 3 and threshold 8 and 2

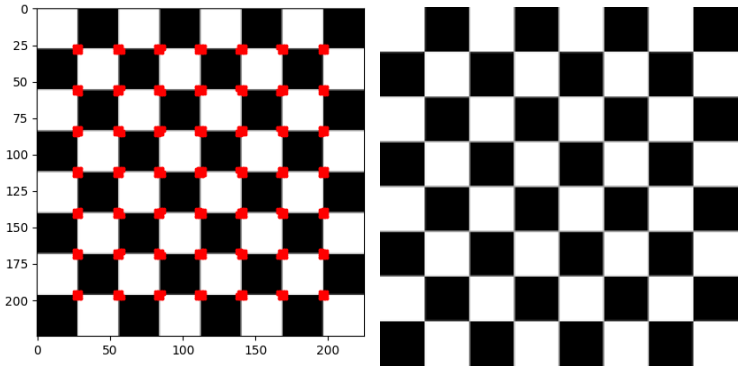


Figure 5.12: *Image corner detection* for chess board

6.Evaluation

Correctness and reliability

The correctness and the reliability of the the method have been observed by visual inspection this have been guaranteed with the best threshold values and has been used the trial and error method to obtain the best

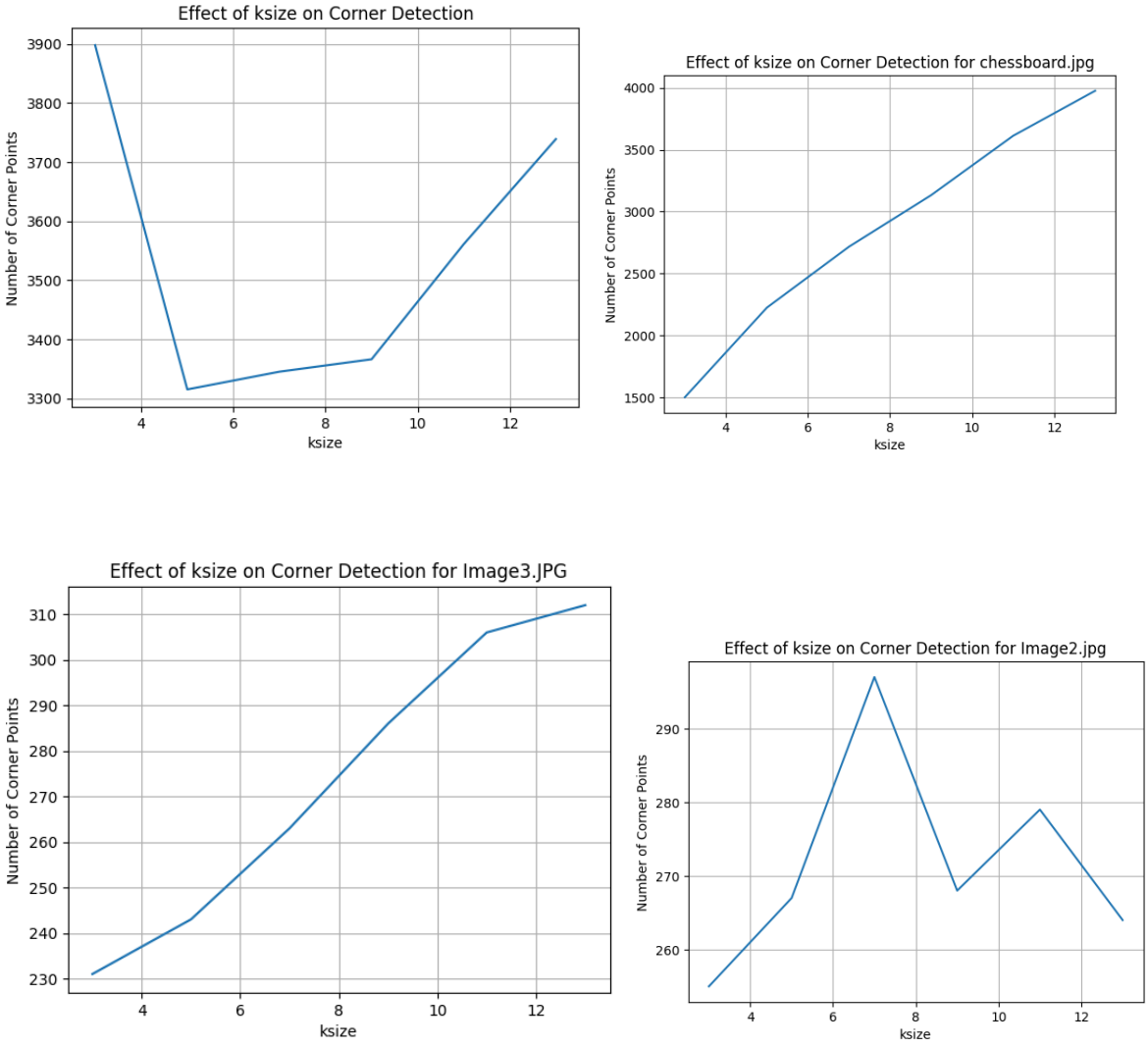


Figure 6.1: *Number of corner points differs when the k size differs*

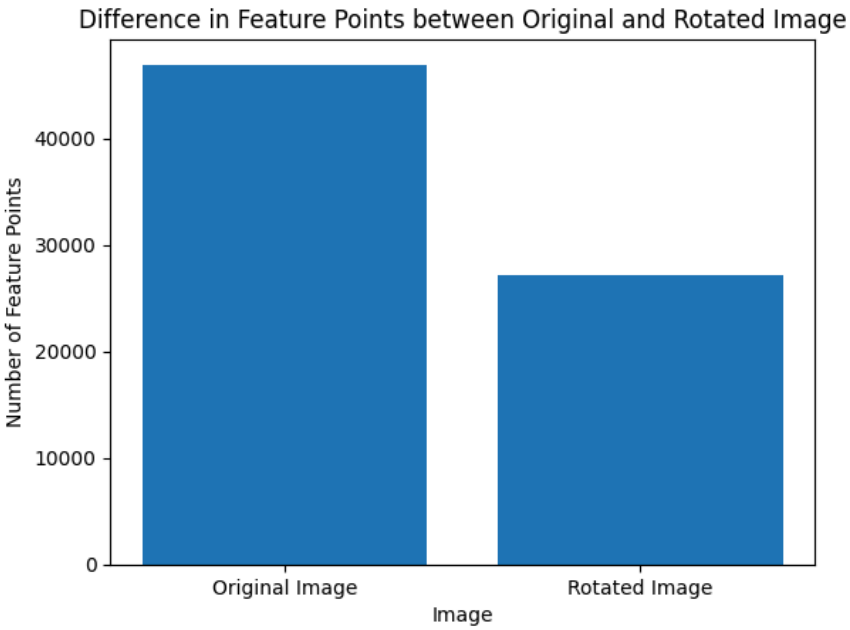


Figure 6.2: *Corner point difference in rotated image*

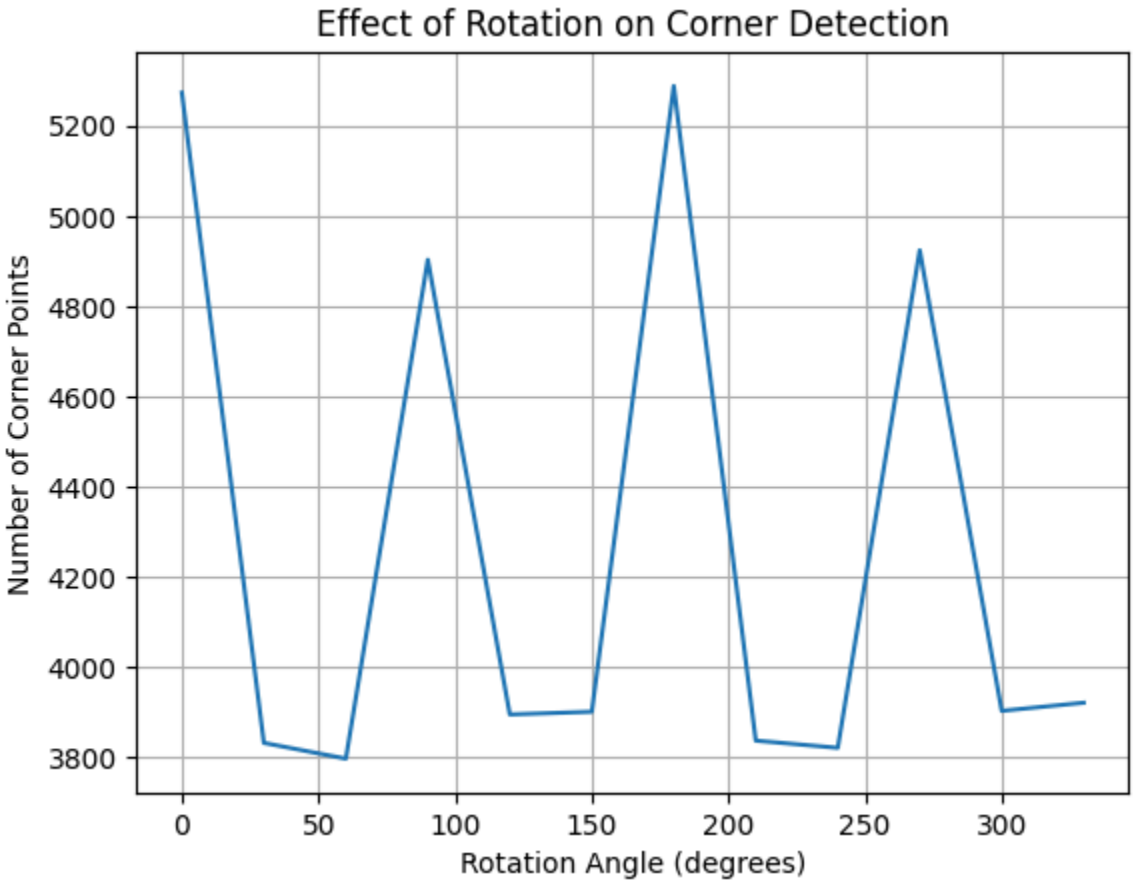


Figure 6.3 *: Rotation versus number of feature points*

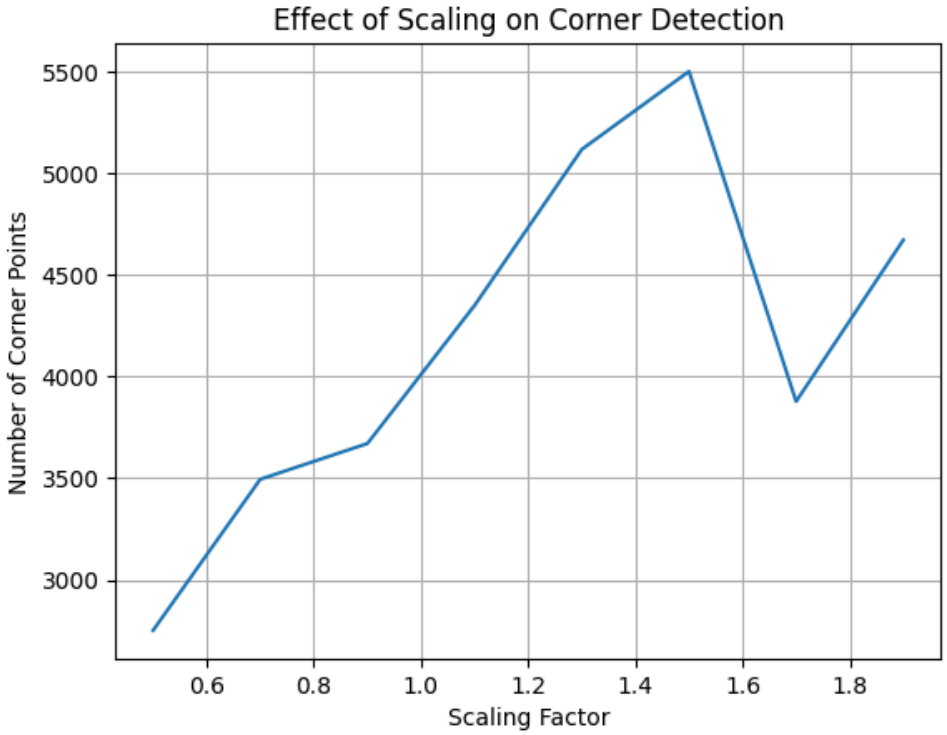


Figure 6.4: *When the scaling factor differs corner detection*

When identifying the difference of the corner points in the images we can clearly see that the Harris corner point is not suitable for the scaling factors in the images.

7.Discussion

When we see the result images we can see that this method works better on a such as a chessboard (figure 5.8) or the star (figure 5.2)or the simple object images (figure 5.1).

As when we discuss about this method this is very dependent on the k size that means the window of the corner (figure 5.3 and figure 5.4) as we can see in the graph which is shows the difference of the of the corner points have a bigger effect in that case.This has been studied by the graphs in the evaluation on the graph (figure 6.1).We can see that the bigger difference have been happened in some of the images which we earlier figure out to be more difficult to personal parameterized

But for the main case scenario this method is working fine in the case of the rotation even in the graph (figure 6.2) and as the image shown in the (figure 5.6) .It shows that the count of the corner points dropping low than the original image because it has been cutted in the different places but the presence of the cornes are same because of that case this works when the rotation happens.

But we can see as the scaling happens (figure 5.8)the points getting together in one's sports and the pixel difference cannot be identified.

And in addition to that we can optimize the results that we have been obtained by differentiating the threshold valued of the function.(figure 5.9)

We can clearly see as the case of the (figure 5.11) even though we have optimized all of the parameters, it still gives us the not corner features of the images which have small differences.

8.Conclusion

In conclusion harris corner detection method is a good use of the when the image has corners which can be easily identify by the gradient and then it can be used with the different parameter difference such as ksize or the threshold value of the function to get the best output images out there.This even give better results in the case of rotation of the image.

But as the image scaling happens this keeps giving less and less points than the actual points .So this is inevitable in scaling of the image and should be used carefully in those cases.

9.Reference

[1] J. Harris and M. Stephens, "A Combined Corner and Edge Detector," in Proc. Alvey Vision Conf., Manchester, UK, 1988, pp. 147-151.