

**The University of Azad Jammu and Kashmir**

**Department of Software Engineering**

**Computer Vision (SE – 4104)**

**Instructor: Engr. Ahmed Khawaja**

**Semester: Fall 2024**

**Session: 2021 – 25**

**Assignment 02**

**Name: Muneeba Shafiq**

**Roll No: 2021 – SE – 28**

# HARRIS CORNER DETECTION

## Objective

To implement the Harris Corner Detection algorithm for identifying corners in an image and analyzing its performance in detecting strong and distinctive corner points.

## Introduction

Corner detection is a fundamental task in computer vision, used to identify points in an image where the intensity changes sharply in multiple directions. These points are crucial for applications like feature matching, image stitching, object recognition, and 3D reconstruction.

The Harris Corner Detection algorithm identifies corners by computing the local changes in intensity within a window, quantifying these changes using eigenvalues of a gradient matrix. Points with two large eigenvalues are classified as corners.

**Key Concepts**:

1. **Gradient Matrix**: Represents intensity changes in horizontal (Ix) and vertical (Iy) directions.
2. **Corner Response Function**: Combines eigenvalues to classify regions as flat, edge-like, or corner-like.
3. **Thresholding**: Separates corners from non-corners using a predefined response threshold.

## Methodology

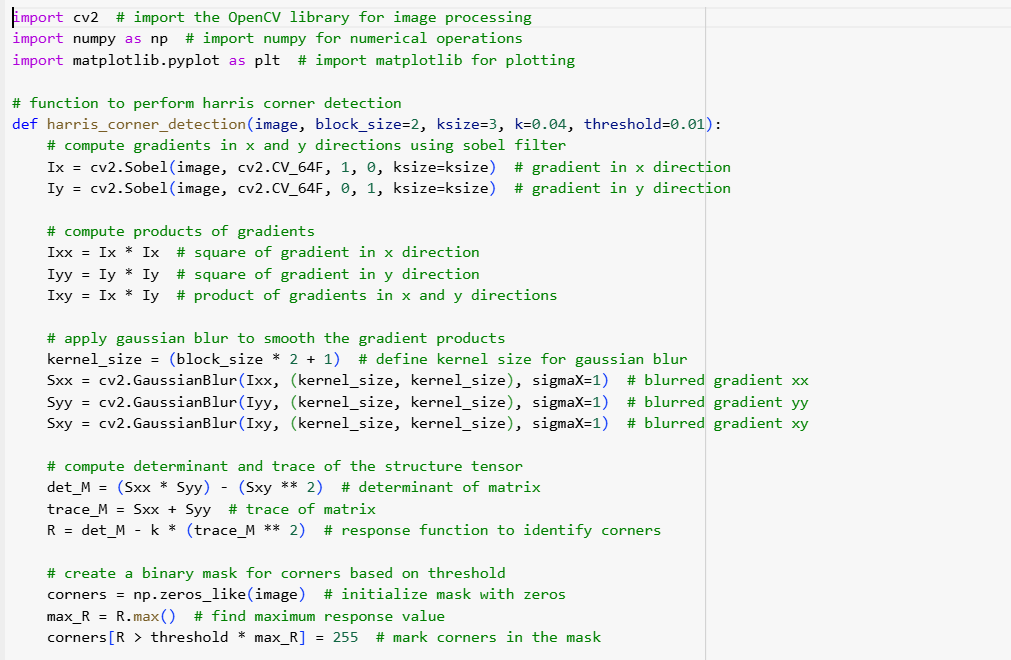
**Steps:**

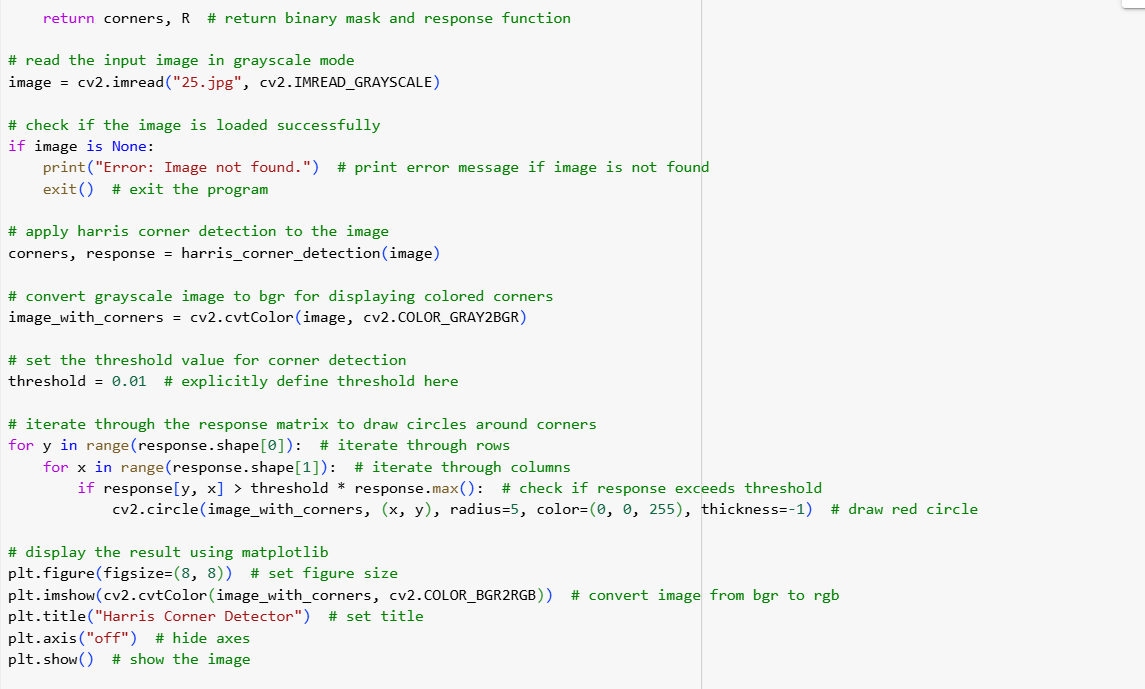
1. **Preprocessing**:
   * Convert the input image to grayscale.
   * Normalize the image to ensure consistent intensity values.
2. **Compute Image Gradients**:
   * Calculate horizontal (Ix) and vertical (Iy) gradients using the Sobel operator.
3. **Form Gradient Matrix**:
   * Compute components of the gradient matrix:

Ixx=Ix2,Iyy=Iy2,Ixy=Ix⋅Iy

1. **Apply Gaussian Smoothing**:
   * Smooth gradient components to reduce noise and emphasize significant regions.
2. **Calculate Corner Response**:
   * Compute the Harris response using: R=det(M)−k⋅(trace(M))2R where M is the gradient matrix, det(M)=IxxIyy−Ixy2 and trace(M)=Ixx+Iyy
3. **Thresholding**:
   * Define a threshold to retain only strong corner points.
4. **Visualization**:
   * Highlight the detected corners on the original image.

## Code Implementation





## Output

