AIM - Write a program that reads in an image, makes a determination about the location of the eyes, and output the positions of the eyes.
In order to proceed with the first assignment, you can use any algorithm that is mentioned in the entire course slides, but you **CANNOT** use any pre-defined functions/libraries (such as vision.cascadeobjectdetector) which will directly recognize eye/face.
Apart from this you can use any MATLAB functions,e.g. function to detect edges or to pre-process given images(e.g. - imgaussfilt etc.)

Objective - To help students learn basics of Matlab and computer vision. Matlab and computer vision features such as face detection and eye detection are used in a wide number of applications nowadays, with the help of this assignment , students will understand the use cases and applications of such programs.

Methods tried

1. Basic manipulations

Steps
1. Import image
2. Change to grayscale
3. Sharpen
4. Change contrast
5. Perform canny edge detection
6. Use “imfindcircles” to find circular objects in image

Reason for Failure
1. Imfindcircles requires to specify radius, every image has different eye radius size so it is difficult to set one general radius
2. Circles detected were not always eye, other non-circles visible to human ey were also picked up
3. Some eyes in picture were semi-circles and could not be detected
4. It was difficult to separate circles detected as eyes from no-eye circles

2. Feature matching

Operations
1. Import image
2. Change to grayscale
3. Sharpen
4. Change contrast
5. Here we compare two images together
6. Import image of generic eye. This image contains only picture of an eye
7. Extract features from eye image and target image
8. Match features between eye image and target image

9. Find points that match and output result

Reason for Failure

1. The eye image used is image specific. Ie only eyes cropped out of target image match
2. Highly ineffective as every eye image needs to be eyes from the target image
3. Defeats the entire purpose of eye detection if cropped image is used.

4. Face geometry

Operations

1. Import image
2. Change to grayscale
3. Sharpen
4. Change contrast
5. Find face in the image
6. Estimate average location of eye in a human being
7. Search for those coordinates in image and output

Reason for Failure

1. Highly inaccurate and ineffective
2. Cascade face detector was not efficient in finding faces for all images
3. Multiple faces were found even if image had only one face
4. It was difficult to estimate location of eyes based on geometry only

Method that works (Actually implemented)

BRISK feature extraction

Operations

1. Import image
2. Change to grayscale
3. Sharpen
4. Change contrast
5. Use BRISK feature detection to find strongest corner points in image
6. Verify accurate identification by plotting those points
7. Take top three feature points
8. Select one point and assign to one eye
9. Select second point such that it is not close to 1st selected point in order to avoid selecting both points on same eye
10. Output

Future improvements

I tried adding face detection such that all the brisk points detected lie within the face boundary and follow basic human face symmetry
This was not possible as faces are not detected accurately for all image, In some image the face is not detected at all, in some images multiple faces are detected even when they are not present

Note
While finding brisk features I observed the eye region gave the strongest features
Therefore I decided to take top 3 strongest features and for 90% of the time those three points were on both of the eyes.
After that step 8 and 9 were done to select two coordinates which were on each respective eye and not same eye
Brisk works efficiently with images in which faces are close up shots looking straight into the camera with a straight face.

This method of brisk feature extraction has its own **limitations**
1. Does not work 10% of the time
2. Inaccurate for tilted face . Where both eye coordinate are not on the same plane
3. Difficulty in working with images with glasses