Name: Niyati Shah
Email: nxs6032@rit.edu
Due Date: 5th Aug, 2017

TOPIC NAME: ACCURATE EYE CENTRE LOCALISATION BY MEANS OF GRADIENTS

Fabian Timm and Erhardt Barth

There are many eye center detection algorithms that have been proposed and they can be divided into three groups. feature-based methods, (ii) model based methods, and (iii) hybrid methods. But most of them are less accurate in daylight or outdoor settings.
The method proposed by the author is a feature based eye center detection algorithm which can accurately detect the eye center in low resolution, low light images.

Eye Center Localization:
This paper forms a mathematical relationship between the possible center and the orientations of all image gradients. The iris is darker than the sclera and if c is the center and g_i is the gradient vector at position x_i. Then, the normalized displacement vector d_i should have the same orientation (except for the sign) as the gradient g_i. The accurate center c* is then calculated as follows:

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{d}_i^T \mathbf{g}_i \right)^2 \right\}$$

$$\mathbf{d}_i = \frac{\mathbf{x}_i - \mathbf{c}}{\|\mathbf{x}_i - \mathbf{c}\|_2}, \quad \forall i : \|\mathbf{g}_i\|_2 = 1$$

The displacement vector is then scaled to unit length to get equal weight to all pixels.
To increase the robustness of the algorithm some pre and post processing is applied. Inaccurate results are formed due to wrinkles, prominent eyebrows, glasses etc.
As pupils are usually darker then the sclera and skin so we can apply a weight for each possible centers such that the dark centers are now more prominent.
Light reflection due to glasses can cause errors, and can be avoided by using Gaussian filters.
Heavy light reflection and hair (eyebrows or eyelashes) can cause errors due to image gradients Which will not have the same orientation as the image gradients of the pupil and the iris. This is removed by the floodKillEdges fuction and is based on the maximum value found in the image.

Things Learnt:

After implementing the paper and looking at the given code, I found that there were some constants declared by the author of the code, after further reading and understanding it was found that some of them were important for the code to run correctly.

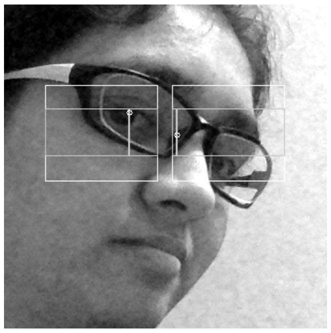
- The eye region fractions.
- The gradient magnitude threshold.
- The size of the eye regions used.

The author of the code contacted Dr. Timm and found that the eye region fraction where the eye center was located at (x + 0.3, y + 0) for right eye and (x + 0.7, y + 0.4) for left eye if x,y were the upper left corner of the rectangle of the detcted face

Different gradient methods gave different accuracy for this algorithm and the algorithm has to be tuned (ie the different variables need to be adjusted) according to the different gradient method. Dr. Timm used Matlab's gradient function and so the author of the code implemented the same for openCV so that he could directly use the gradient threshold used by Dr. TImm.

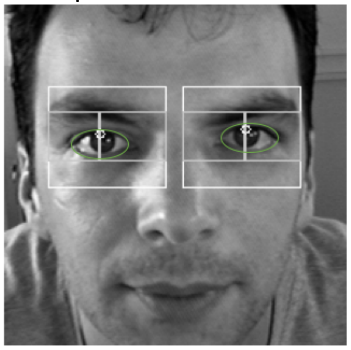
The dot product used give us negative values when the vectors point to different directions. As the iris is darker then sclera the vectors will point from the dark to light regions. SO the center they face will be same as displacement vector d, and anything in opposite side would not be applicable so to avoid any errors they are rounded to zero.

As seen here the code inaccurately detects the eye as the glasses, to eliminate the such errors the below ideas can be used.



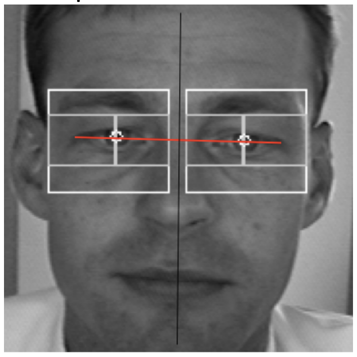
Create an ellipse and fitting it in the eye that is found and making sure that only the possible eye center points detected are within that ellipse. This can eliminate a lot of those errors that are caused by prominent eyebrows, or glasses.

Example:



After a face is detected we can approximately tell where the eyes can be located. The two centers will be in a single straight line between the eyes(rectangles) detected, So we can make use of this information and eliminate those false positives detected that do not fall on that line.

Example:



Final Output Images from BioID database: