Automated Strabismus Detection using Smartphones

Pranav Rawal

Department of Electrical & Computer Engineering, McMaster University

Introduction

Strabismus is the misalignment of the eyes and is commonly referred to as cross eyedness. It is one of the most common eye conditions and affects approximately 4% of children under the age of six (Crossed-Eyes (Strabismus), n.d.). The misalignment can occur in one or both of the eyes where the eye can be turned inwards, outwards, upwards or downwards. It is common for more than one of these misalignments to occur simultaneously. Early detection and treatment of this condition is necessary to achieve the best chances of restoring normal vision (Cooper & Cooper, 2016).

Normally, an eye specialist would administer corneal light reflex test (or Hirschberg test) to diagnose ocular misalignment. The test is relatively simple, a light source is used to shine a light in the patient's eyes from about 2 feet from the patient's face. The eye specialist then determines the alignment of each of the eyes based on the location of the light sources' reflection on the cornea (Hu, 2016). In a patient with properly aligned eyes, the reflections should appear in the center of the pupils, as can be seen in figure 1 below.

Figure 1: *Patient with normal alignment.* Hu, K. (2016). Retrieved from http://morancore.utah.edu/wp-content/uploads/2017/08/hu_assessment_002.jpg

If the corneal reflections are not located in the center of the pupils then the patient may be diagnosed with some form of strabismus, the four possibilities are highlighted below in figure 2.

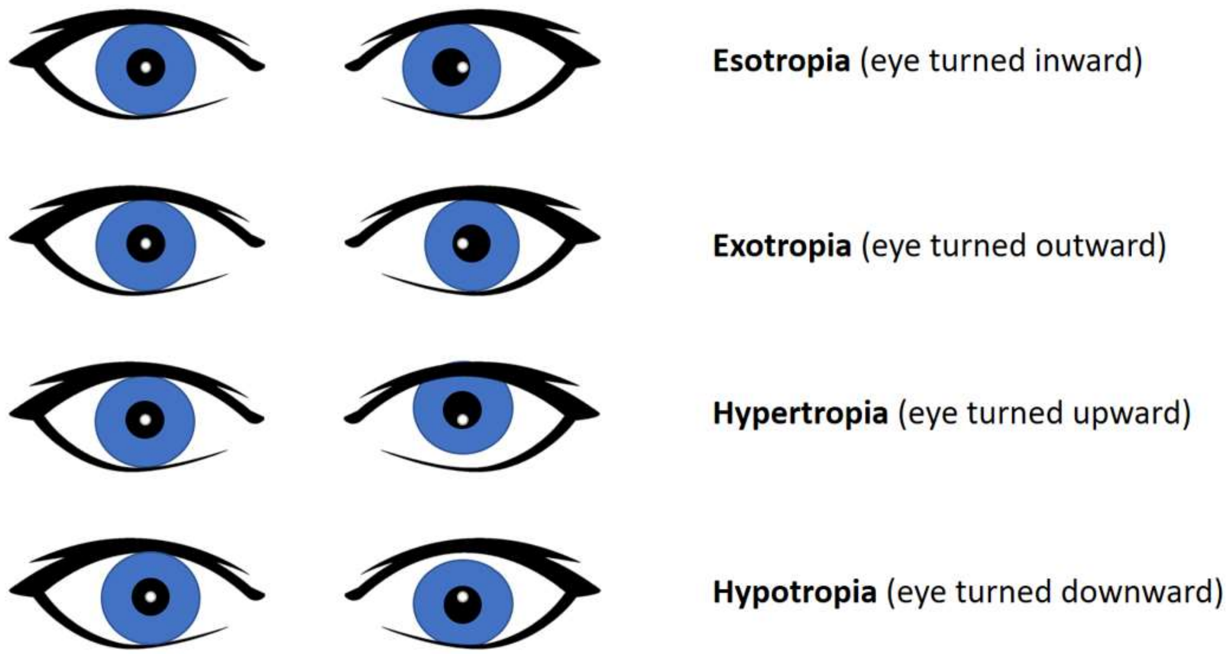


Figure 2: *Ocular misalignment from Hirschberg test.* Hu, K. (2016). Retrieved from http://morancore.utah.edu/wp-content/uploads/2017/08/hu_assessment_003.jpg

The goal of this project is to provide access to such diagnostic tools by automating the Hirschberg test with the use of smartphones. This would allow concerned parents and guardians to flag issues regarding their child's eye alignment by simply taking a picture of their child's face with the flash enabled and then using an application to assess the concentricity of the circular flash reflection and the circular iris-sclera boundary of the eye.

The application in this project is implemented using Python programming language, as it can be easily integrated into a mobile application in the future. OpenCV, which is a Python library designed specifically for computer vision applications, was used to reduce the development time of the strabismus detection algorithm. In the next section, the high-level details of the algorithm are discussed.

Overview of the Algorithm

Figure 3: *Overview of the Automated Strabismus Detection algorithm.*

Input Image

The images may be acquired from either the front or the rear smartphone cameras. The procedure for acquiring the images are intended to reproduce the procedure for the Hirschberg test highlighted in the introduction. Below are the steps which were followed to produce the dataset for testing the algorithm.

1. Hold the smartphone approximately 2 feet away from your face, or with your arms fully extended.
2. Align the smartphone such that the flash will be located at the midline of your face.
3. Focus your gaze on the flash, then take the photo.
4. Assess the picture and ensure the following:
    a. Reflection of the flash is visible in the iris as a pinpoint light source.
    b. No other light sources' reflections are present in the iris.
    c. Only one face in the image. This includes the background and the clothing worn by the person in the image.
    d. Minimal blurring.

Eye Detection

The eye-detection works in two steps. First, detecting a face and then within the face, detecting eyes. It is possible to skip the step of detecting the face. However, it is possible to have many false positives if there are other eye-like objects in the input image.

For the detection, OpenCV's Haar Cascade detection function is used. OpenCV provides the ability to train a customized classifier that can be used to detect any feature in an image. But for this project, a pre-trained classifier for the face and eyes were used

(Rahman , 2013).

The Haar Cascade function outputs indices of the input image which define the region of interest. Using these indices, the region containing the eyes can be cropped from the original input image from the previous step. Points worth noting about the outputted eye image patches:

1. Width and height are equal (square patches)
2. Iris is **usually** located somewhere in the center of the patch.
3. Patch includes eyebrows which can cause issues when detecting the iris. Further cropping of the eye image patch is required, through experimentation it was determined that retaining 70% of the center of the outputted eye image patch from the Haar detection step can resolve this issue.

The first point simplifies some of the coding and point 2 can be exploited to help with the iris segmentation in the next step. To illustrate the performance of the Haar Cascade detection the input image **front_cam_bright.jpg** from the working directory is shown in figure 4 below, and the detected face and eye patches are shown in figure 5 below.

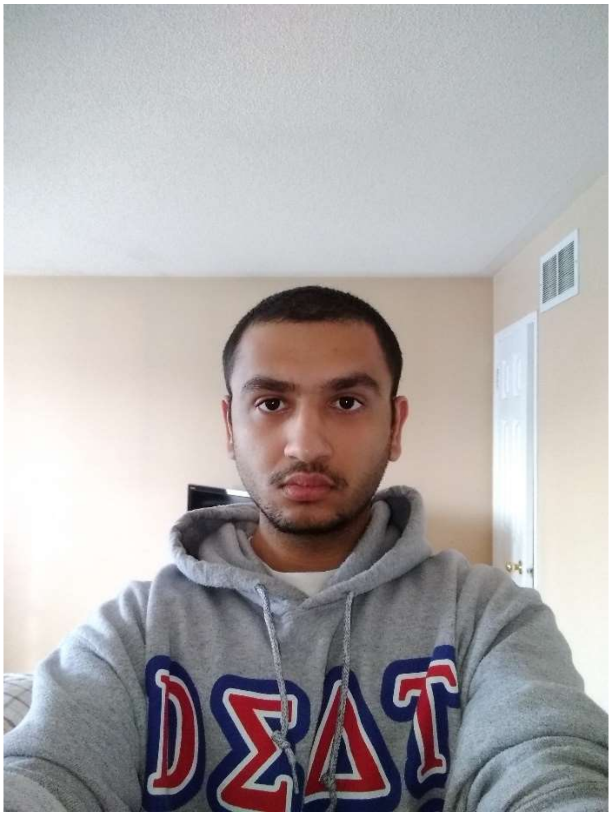


Figure 4: *front_cam_bright.jpg (input image)*

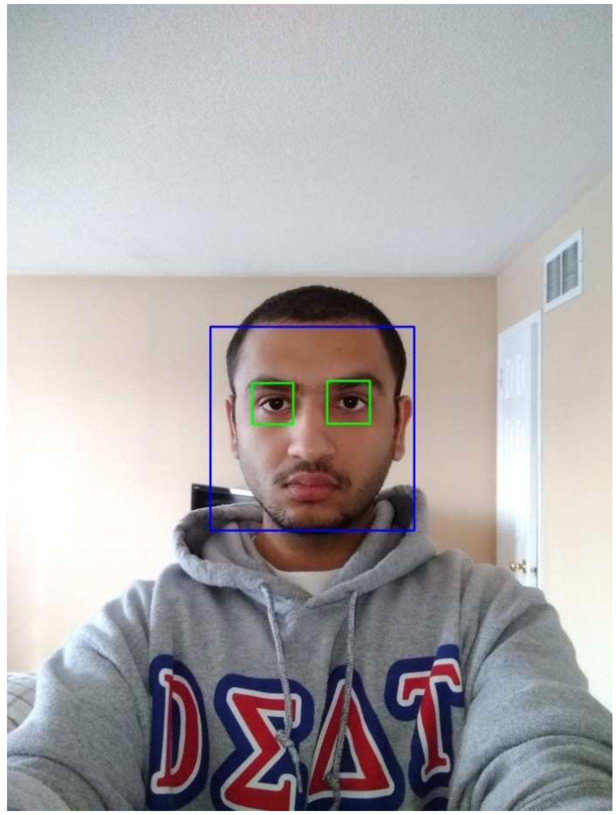


Figure 5: *face_&_eyes.jpg (Haar Cascade detections)*

Iris Segmentation

Using the assumption that the iris will most likely be located somewhere in the center of the cropped eye image, the color of the iris can be learned. With the knowledge of the iris color, a mask can be created which will contain all the pixels from the cropped eye image which best represent the relative location of the iris in the cropped eye image.

Learning the color of the iris is done by sampling twenty (5 x 5) patches near the center of the cropped image. For the input image in figure 4, the first, the twentieth, and all twenty patches for the two eyes are shown in figures 6 to 11 on the following page.

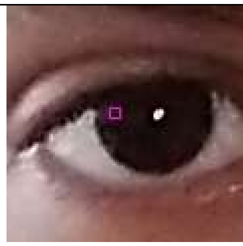


Figure 6: *1st sampled patch in the right eye*

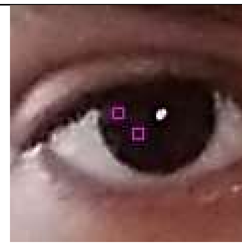


Figure 7: *1st and 20th sampled patch in the right eye*

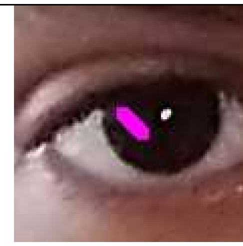


Figure 8: *All 20 sampled patches in the right eye*

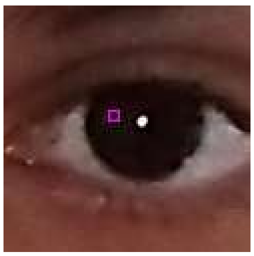


Figure 9: *1st sampled patch in the left eye*

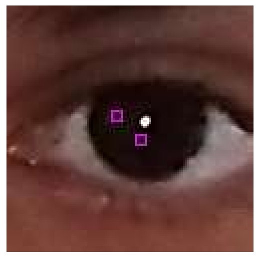


Figure 10: *1st and 20th sampled patch in the left eye*

Figure 11: *All 20 sampled patches in the left eye*

For each of the twenty sampled patches, the average RBG values are computed, the average RGB value is then converted into an average intensity value for each patch. Then the twenty average intensities are sorted in ascending order and all intensities greater than the median average intensity are disregarded. The last step of removing large intensities is done to ensure that if any of the twenty patches included portions of the iris with the flash or the sclera of the eye, then these patches would have a larger average intensity associated with them. The RGB values associated with these large intensity values would not be a good representation of the color of the iris, and should not be considered when learning the color of the iris.

Then the average RGB value of the qualifying patches is computed and is used as the best representation of the color of the iris. A 35% tolerance is added to this learned iris color and a mask is created by iterating through all the pixels in the cropped eye image which fall within this tolerance. Continuing with the example of the input image in figure 4, the masks for the right and left eye are shown in figures 12 and 13 below.

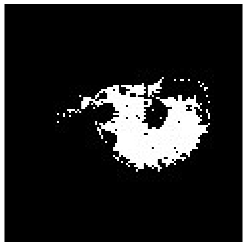


Figure 12: *Mask created for the right eye*

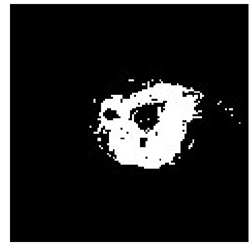


Figure 13: *Mask created for the left eye*

Finally, the center of mass or the centroid of the masks are calculated to get a sense of the approximate center of the iris. The centroid will be used in the subsequent step when the Circular Hough Transform will be used to detect the exact center and radius of the flash and iris in the cropped eye images.

Iris and Flash Detection

The backbone of the iris and flash detection is the Circular Hough Transform (CHT), as mentioned earlier. However, some processing is required to get repeatable results from the algorithm. The processing steps to prepare for CHT are as follows:

1. Converting the cropped eye images to grayscale
2. Binary Thresholding the grayscale cropped eye images
3. Morphological Transformation (Opening) to remove noise in the image.
    - Opening is simply erosion followed by dilation (Morphological Transformations, n.d.)

After these processing steps the image that will be used as an input for the OpenCV's built-in HoughCircles function is shown in figures 14 and 15.

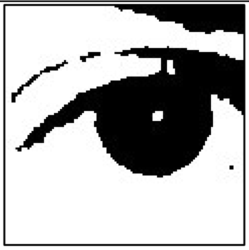


Figure 14: *Processed right eye (input for CHT)*

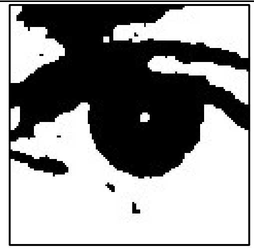


Figure 15: *Processed left eye (input for CHT)*

The HoughCircles function runs Canny Edge Detection before fitting circles to the input images, otherwise that would also be a step in the processing steps above. HoughCircles takes several parameters as inputs, details of which can be found in the OpenCV documentation (Feature Detection, n.d.).

The main takeaway is that strict constraints on these parameters will reduce the number of detections and perhaps not detect the features of interest. Loose constraints will increase the number of detections and increase the complexity of choosing the suitable circles. Therefore, the constraints on the parameters are set such that the number of candidate detections outputted by the function are low enough that selecting the most suitable circles is relatively quick. These constraints were chosen through experimentation. Figures 16 and 17, on the following page, illustrate all the candidate circles detected by HoughCircles on input images from figures 14 and 15. Cyan circles represent the candidate detection for the flash and green circles are candidate detections for the iris.

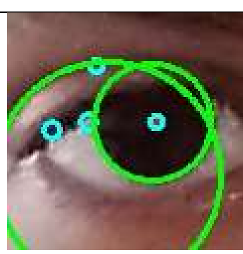


Figure 16: *Candidate circles detected on right eye.*

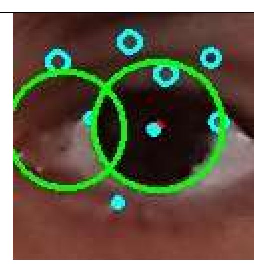


Figure 17: *Candidate circles detected on left eye*

From figures 16 and 17, it is clear that the most suitable circles for the flash and the iris are included in the candidate circles set. The circles located closest to the centroid, calculated from the Iris Segmentation stage, will be selected as the suitable detections. Figures 18 and 19 show the chosen circles in this case.

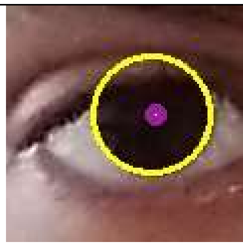


Figure 18: *Selected iris and flash detections on the right eye*

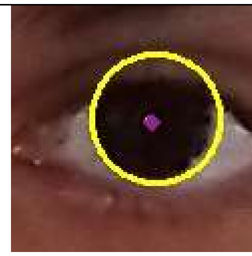


Figure 19: *Selected iris and flash detections on the left eye*

At the end of this stage the coordinates of the flash and the iris, also the radius of the iris for each of the eyes are known. These three pieces of information are what are used to understand the alignment of the eye and provide insight into whether the person in the image has strabismus or not. Figure 20 is the culmination of all the steps thus far and illustrating the location of the iris and flash in the original input image from figure 4.

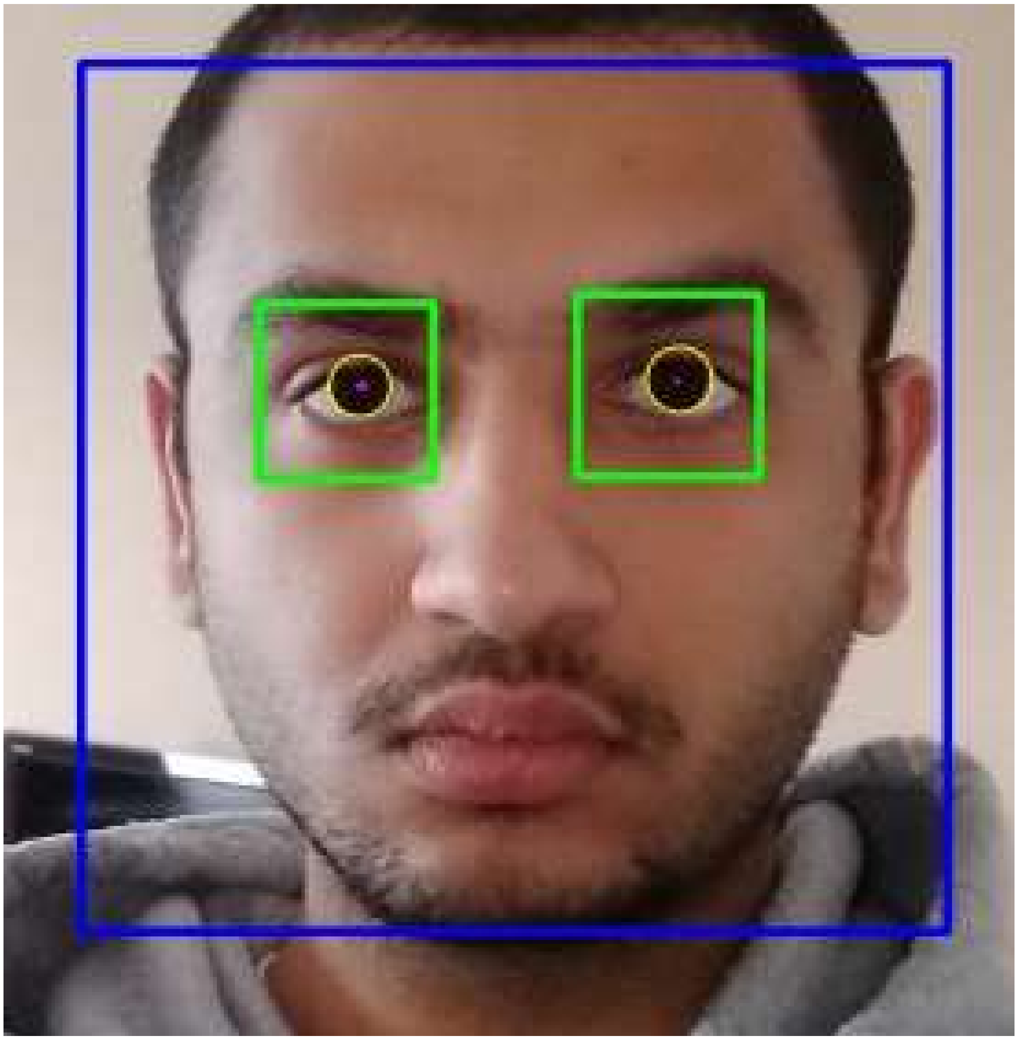


Figure 20: Input image with the Iris and flash detection.

Strabismus Classification

The four cases of strabismus (refer to figure 2) are detected by this algorithm based on the conditions described in table 1 below.

Table 1: Strabismus Classifications

Alignment Ratio	Right Eye	Left Eye
$\frac{f_X - I_X}{I_R} < -threshold$	Estropia (eye turned outwards)	Extropia (eye turned inwards)
$\frac{f_X - I_X}{I_R} > threshold$	Extropia (eye turned inwards)	Estropia (eye turned outwards)
$\frac{f_Y - I_Y}{I_R} < -threshold$	Hypotropia (eye turned downwards)	Hypotropia (eye turned downwards)
$\frac{f_Y - I_Y}{I_R} > threshold$	Hypertropia (eye turned upwards)	Hypertropia (eye turned upwards)
$\frac{\lvert\vec{f} - \vec{I}\rvert}{I_R} < threshold$	Normal	Normal

Where $\vec{f} = (f_X, f_Y)$ represents the coordinates of the center of the flash and $\vec{I} = (I_X, I_Y)$ represents the coordinates of the center of the iris. I_R represents the radius of the iris. The value for the threshold was gathered through trial and error using control images: **left_estropia.jpg**, **rear_cam_up.jpg**, and **rear_cam_down.jpg**. The threshold was chosen such that all other tested images in the directory should be classified as "Normal" and the control images mentioned should have the appropriate classifications. In this case, the threshold value was set as **0.155**.

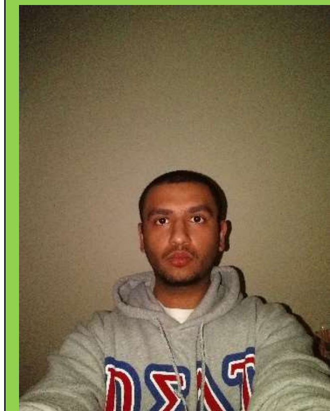
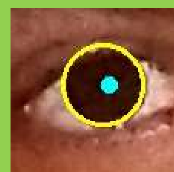
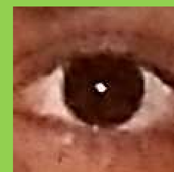
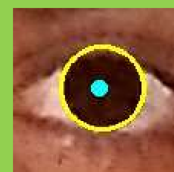
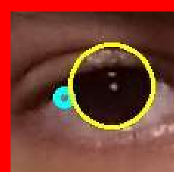
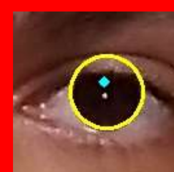
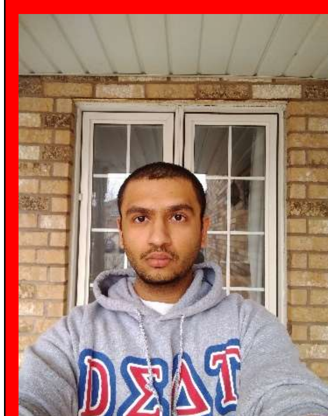
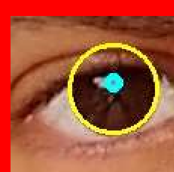
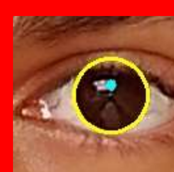
This is an unreliable method for classifying. The correct method would be to use a Machine Learning clustering approach to train a model to learn the five classes and then trying to classify a new image based on the trained model. However, this would require a large dataset of labeled images which were not available at the time of developing this algorithm.

Regardless, the classification results along with the alignment ratios for the input image from figure 4 are provided below in figure 21.

```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.06451612903225806 , 0.06451612903225806 )
Right Eye Alignment Ratio: ( 0.06896551724137931 ,  0.0 )
Left eye is normal
Right eye is normal
[Finished in 2.2s]
```

Figure 21: Classification results for front_cam_bright.jpg

Results

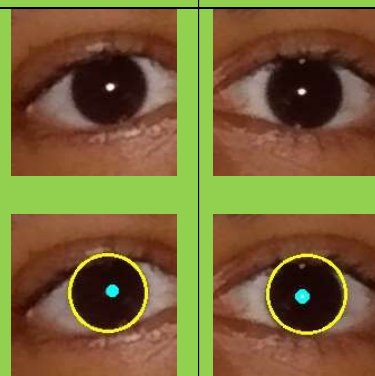
Input Image	Right Eye	Left Eye	Classification Results & Discussion
 **front_cam_dark.jpg**	 	 	Alignment Ratio Notation: (Horizontal Alignment, Vertical Alignment) Left Eye Alignment Ratio: (-0.07407407407407407 , 0.0) Right Eye Alignment Ratio: (0.15384615384615385 , 0.0) Left eye is normal Right eye is normal • Picture is taken in complete darkness with the front camera of the smartphone • Lit only by the flash resulting in overall blocky/low resolution image • Despite the low-light the algorithm still performs well and classifies alignment as normal
 **front_cam_light.jpg**	 	 	Alignment Ratio Notation: (Horizontal Alignment, Vertical Alignment) Left Eye Alignment Ratio: (-0.07142857142857142 , -0.2857142857142857) Right Eye Alignment Ratio: (-1.1428571428571428 , 0.2857142857142857) Hypotropia of the left eye Esotropia of the right eye Hypertropia of the right eye • Picture taken indoors while standing under ceiling light with front camera • The reflection of the ceiling light is present in both eyes • Iris detected correctly in both eyes • Flash not detected at all in the right eye ➢ Resulting in false positive strabismus classifications • Reflection of ceiling light detected as the flash in the left eye ➢ Resulting in false positive strabismus classifications
 **front_cam_outside.jpg**	 	 	Alignment Ratio Notation: (Horizontal Alignment, Vertical Alignment) Left Eye Alignment Ratio: (0.0 , -0.2857142857142857) Right Eye Alignment Ratio: (0.0 , -0.13793103448275862) Hypotropia of the left eye Right eye is normal • Picture taken outside on a cloudy day with front camera • The reflection of the sky is much brighter than the flash • Iris detected correctly in both images • Reflection of sky incorrectly detected a flash in both eyes ➢ Resulting in false positive strabismus classification

rear_cam_bright.jpg

```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.05714285714285714 , 0.0 )
Right Eye Alignment Ratio: ( 0.05714285714285714 ,  0.0 )
Left eye is normal
Right eye is normal
```

- Picture taken indoors, lit by window but reflection not present in iris
- Taken using the rear camera and flash of the smartphone
- Iris and flash correctly detected by the algorithm
- Correctly classifies both eyes as normal

rear_cam_dark.jpg



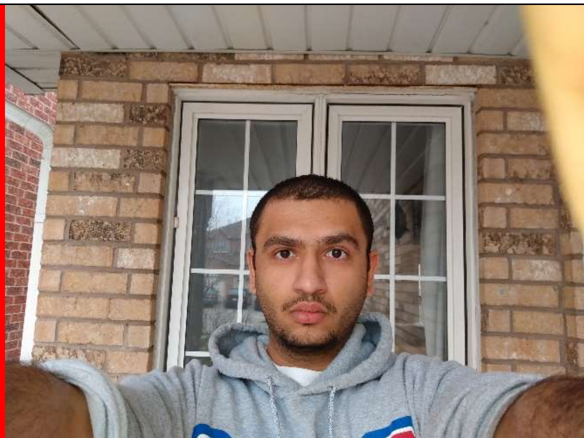
```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.11428571428571428 , 0.05714285714285714 )
Right Eye Alignment Ratio: ( 0.11764705882352941 ,  -0.058823529411764705 )
Left eye is normal
Right eye is normal
```

- Picture is taken in complete darkness by the rear camera and flash of the smartphone
- Iris and flash correctly detected by the algorithm
- **The head is slightly tilted which gives the appearance of strabismus in both eyes**, but because the alignment ratio is not exceeding the threshold of 0.155, both eyes are still correctly classified as normal

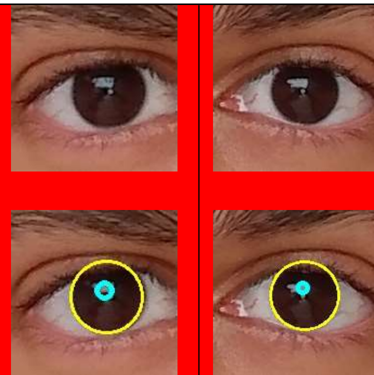
rear_cam_light.jpg

```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.1 , 0.1 )
Right Eye Alignment Ratio: ( 0.05714285714285714 ,  0.0 )
Left eye is normal
Right eye is normal
```

- Picture taken indoors while standing under ceiling light with rear camera
- Reflection of ceiling light present in the left eye, but the rear flash is strong enough to still be detected correctly
- Iris and flash are correctly detected by the algorithm
- Correctly classifies both eyes as normal

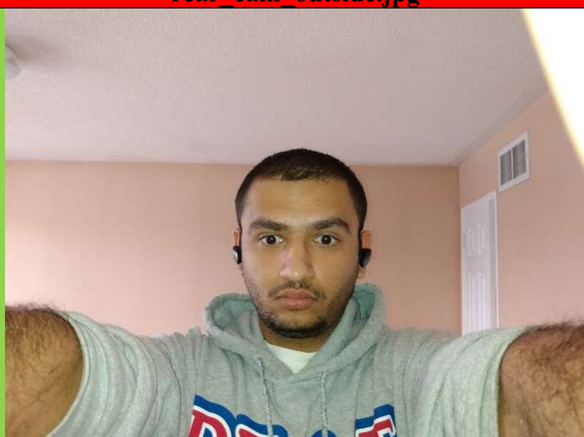


rear_cam_outside.jpg

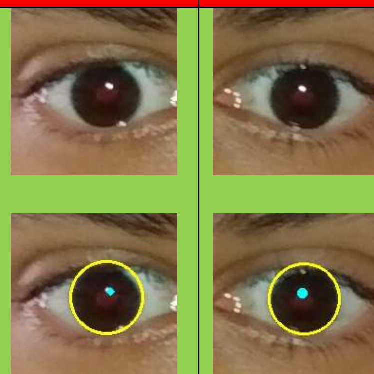


```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.05714285714285714 , -0.22857142857142856 )
Right Eye Alignment Ratio: ( -0.06060606060606061 ,  -0.18181818181818182 )
Hypotropia of the left eye
Hypotropia of the right eye
```

- Picture taken outside on a cloudy day with rear camera
- The reflection of the sky is much brighter than the flash
- Iris detection is slightly off for the right eye
- Iris is correctly detected for the left eye
- Reflection of sky incorrectly detected a flash in both eyes
  - Resulting in false positive strabismus classification

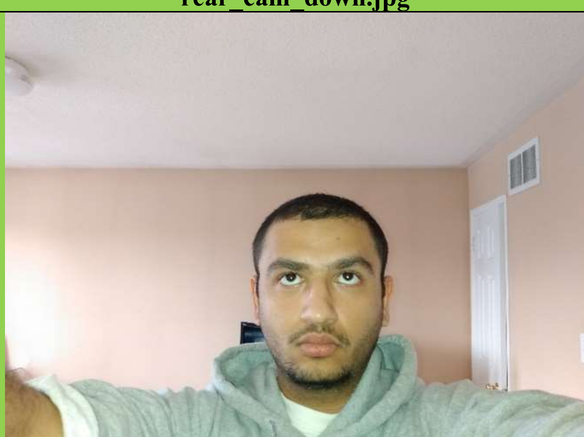


rear_cam_down.jpg

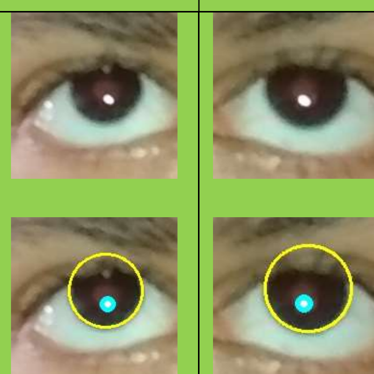


```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.05405405405405406 , -0.16216216216216217 )
Right Eye Alignment Ratio: ( 0.10810810810810811 ,  -0.16216216216216217 )
Hypotropia of the left eye
Hypotropia of the right eye
```

- Picture taken indoors, lit by window but reflection not present in iris
- Taken using the rear camera and flash of the smartphone
- Iris and flash correctly detected by the algorithm
- Subject is purposely looking downwards to force a hypotropia classification
- The algorithm correctly classifies both eyes as hypotropia

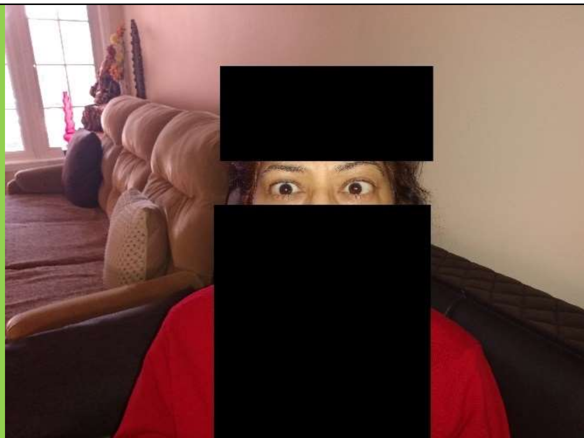


rear_cam_up.jpg



```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.1 , 0.35 )
Right Eye Alignment Ratio: ( 0.05263157894736842 ,  0.3684210526315789 )
Hypertropia of the left eye
Hypertropia of the right eye
```

- Picture taken indoors, lit by window but reflection not present in iris
- Taken using the rear camera and flash of the smartphone
- **Iris detection is slightly off for both eyes due to the blurriness, likely due to tremor in the hands while taking the picture**
- Flash correctly detected by the algorithm
- Subject is purposely looking upwards to force a hypertropia classification
- The algorithm correctly classifies both eyes as hypertropia

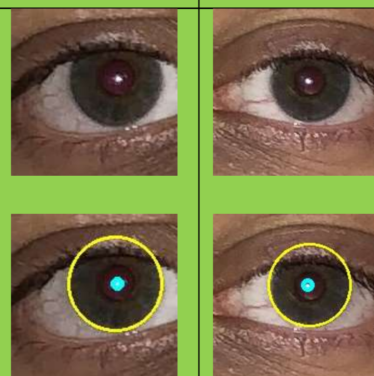


left_estropia.jpg

```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( 0.17391304347826086 , 0.0 )
Right Eye Alignment Ratio: ( 0.041666666666666664 ,  -0.08333333333333333 )
Esotropia of the left eye
Right eye is normal
```

- Picture taken indoors, lit by window but reflection not present in iris
- Taken using the rear camera and flash of the smartphone
- Iris and flash correctly detected by the algorithm
- Subject suffers from Estropia of the left and the algorithm is able to correctly identify it
- **Note: The algorithm was run on the uncensored version of the image.**

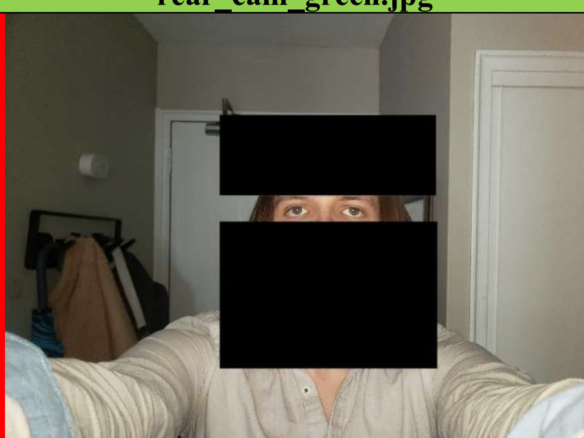
rear_cam_green.jpg



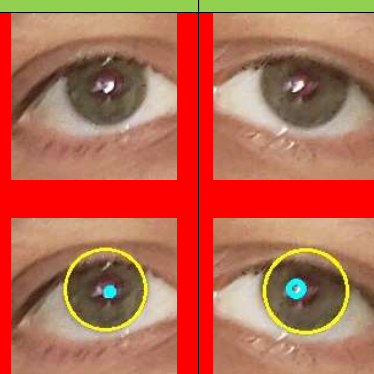
```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.044444444444444446 , 0.0 )
Right Eye Alignment Ratio: ( 0.046511627906976744 ,  0.0 )
Left eye is normal
Right eye is normal
```

- Picture taken indoors, lit by ceiling light but reflection not present in iris
- Taken using the rear camera and flash of the smartphone
- Iris and flash correctly detected by the algorithm
- Subject has green eyes and the algorithm correctly classifies the subject as having normal alignment for both eyes regardless of eye color

Note: The algorithm was run on the uncensored version of the image.



rear_cam_green2.jpg



```
Alignment Ratio Notation: ( Horizontal Alignment, Vertical Alignment)
Left Eye Alignment Ratio: ( -0.23529411764705882 , -0.058823529411764705 )
Right Eye Alignment Ratio: ( 0.11428571428571428 ,  0.05714285714285714 )
Exotropia of the left eye
Right eye is normal
```

- Picture taken indoors, lit by ceiling light and reflection is partially present in iris
- Taken using the rear camera and flash of the smartphone
- Iris detection is slightly off for the left eyes due to the blurriness
  - Resulting in incorrect classification of strabismus
- Iris and flash correctly detected for the right eye
- **Subject has green eyes and the algorithm correctly classifies the subject as having normal alignment for the right eye regardless of eye color**

Conclusion

All input images in the results section were captured on Motorola G7 Play, a relatively cheap smartphone, and underwhelming camera specifications. This is only to say that the algorithm may perform much better on input images captured from a newer smartphone such as an iPhone, which is known for its superior cameras. That being said, there are still some meaningful conclusions that can be drawn from the results above.

Firstly, the performance of the algorithm seems to be greatly improved on the images captured from the rear camera. This is because the rear camera Moto G7 has a higher megapixel count and better-quality image sensor resulting in a higher resolution image with better image quality compared to the front camera. In addition to a better camera, the rear flash is much more powerful resulting in a more prominent corneal reflection which can be more easily detected by the Circular Hough Transform. This can be seen in the **front_cam_light.jpg** and **rear_cam_light.jpg** examples above, where the environmental lighting overpowered the front camera's flash and resulted in an incorrect flash detection, resulting in a false strabismus classification. However, the rear camera flash was powerful enough to be detected by the algorithm and correctly classify the subject with normal ocular alignment.

The powerful rear flash may not always be desirable, because it leaves the subject temporarily blinded. And considering this is an application for young children who are more sensitive to bright lights, its flash may be doing more harm than good. In such cases, the algorithm is just as effective at detecting the iris and the flash in any lighting condition so long as there are no secondary light reflections present in the iris. This can be seen in **front_cam_bright.jpg** (the example that was explored in detail throughout the overview of the process) and **front_cam_dark.jpg**.

Next, some improvements to the algorithm itself will be discussed. In particular, there were two instances in which blurring was present in the input images, **rear_cam_up.jpg**, and **rear_cam_green2.jpg**. In both cases the blurring resulted in a suboptimal selection of the iris circle, however, in **rear_cam_up.jpg** the algorithm was still able to correctly classify the subject as having normal alignment. Perhaps this means that the algorithm may be able to perform with a certain amount of motion blur in the image. If there is a low complexity method by which blurring in an image can be quantified, then learning the threshold of such blurring can be used to reject an image and request the user to retake an image instead of fitting a suboptimal circle to features of interest and providing false strabismus classifications.

Another area for improvement is the method by which iris segmentation is performed. The current method relies on the assumption that the iris will be located approximately in the center of the detected eye image (output from the Haar detection), but that may not be the case all the time. K-means clustering for image segmentation is a technique that should be explored to replace the current segmentation technique because it does not rely on such assumptions.

This project was initiated to provide parents and guardians accessibility to tools to help screen children for ocular misalignment to ensure detection and treatment for such issues can be administered as soon as possible. To do this, a cross-platform smartphone application must be developed to process the images immediately and connect parents with a medical professional if there is a cause for concern and prevent any delays in treatment.

References

Cooper, J., & Cooper, R. (2016). *What is Strabismus?* Retrieved from Optometrists Network: https://www.strabismus.org/detection_diagnosis_lazy_eye.html

Crossed-Eyes (Strabismus). (n.d.). Retrieved from Stanford Children's Health: https://www.stanfordchildrens.org/en/topic/default?id=crossed-eyes-strabismus-90-P02109

Feature Detection. (n.d.). Retrieved from OpenCV: https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=houghcircles

Hu, K. (2016). *ALIGNMENT ASSESSMENT (HIRSCHBERG)*. Retrieved from Moran CORE : http://morancore.utah.edu/basic-ophthalmology-review/alignment-assessment-hirschberg/

Morphological Transformations. (n.d.). Retrieved from OpenCV: https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html

Rahman , A. (2013, August 22). *Face Detection using Haar Cascades*. Retrieved from GitHub: https://github.com/abidrahmank/OpenCV2-Python-Tutorials/blob/master/source/py_tutorials/py_objdetect/py_face_detection/py_face_detection.rst