Introduction

Image processing techniques can be used to diagnose and monitor various eye diseases. Here are some techniques which can be used to detect eye disease:

Optical Coherence Tomography (OCT): OCT (Fujimoto, 2000) is commonly used to diagnose and monitor diseases such as macular degeneration, diabetic retinopathy, and glaucoma. Using OCT in combination with catheters and endoscopes enables high-resolution intraluminal imaging of organ systems.

Fundus photography: Fundus photography (A.G., 1994) is a technique that captures images of the retina using a specialized camera. These images can be used to detect and monitor diseases such as diabetic retinopathy, macular degeneration, and glaucoma. Littmann's formula relating the size of a retinal feature to its measured image size on a telecentric fundus camera film is widely used. It requires only the corneal radius, ametropia, and Littmann's factor q obtained from nomograms or tables (A.G., 1994).

Fluorescein angiography: Fluorescein angiography (Yannuzzi, n.d.,) involves injecting a special dye into a patient's bloodstream and taking photographs of the retina as the dye circulates through the blood vessels.

Fluorescein angiography (FA) (Yannuzzi, n.d.) is a medical imaging technique used to visualize the blood vessels in the retina, the light-sensitive layer at the back of the eye. It is used to diagnose and monitor a range of eye conditions, including macular degeneration, diabetic retinopathy, and uveitis.

Automated visual field testing: Automated visual field testing measures the patient's visual field to detect any abnormalities that may be indicative of glaucoma or other eye diseases.

Machine learning-based techniques: Machine learning algorithms can be trained on large datasets of eye images to detect and classify eye diseases automatically. For example, A machine learning and deep learning algorithm can be trained to detect diabetic retinopathy from fundus images, achieving accuracy levels that are comparable to human ophthalmologists.

Overall, machine learning techniques have shown great potential in the detection and diagnosis of various eye diseases. By enabling earlier detection and more accurate diagnosis, machine learning can help improve patient outcomes and prevent vision loss.

The various approaches to determine eye disease using image processing.

Retinal imaging: Retinal imaging is a non-invasive technique that uses specialized cameras to capture detailed images of the retina. These images can be analyzed using image processing to detect and monitor various eye diseases like diabetic retinopathy, glaucoma, and age-related macular degeneration.

Optical coherence tomography (OCT): OCT is a non-invasive imaging technique that uses light waves to create detailed images of the retina.

Fundus photography: Fundus photography is a technique that captures images of the retina using a specialized camera and captured images can be analyzed using image processing and analysis techniques to detect and monitor various eye diseases, such as diabetic retinopathy and macular degeneration.

Corneal topography: Corneal topography images can be analyzed using image processing and analysis techniques to detect and monitor various eye diseases, such as keratoconus and corneal dystrophies.

Iris recognition: Iris recognition is a technique that uses unique patterns in the iris to identify individuals.

Overall, image processing and analysis techniques play an important role in the analysis of eye images, enabling earlier detection and more accurate diagnosis of various eye diseases.

Here, in this particular experiment, based on the retinal image, using image processing techniques, we need to determine whether the eye is having an issue or not.

Methods

In image processing, an image is a two-dimensional array of pixels. Each pixel is a tiny square or dot that represents a single point in the image. The value of each pixel represents the brightness or color of that point in the image. In grayscale images, each pixel has a single value representing its brightness level, typically ranging from 0 (black) to 255 (white). In color images, each pixel has three values representing the intensity of the red, green, and blue color channels, respectively.

Image processing techniques can be used to manipulate the values of individual pixels or groups of pixels in an image to achieve various goals, such as enhancing image quality, detecting features, or segmenting objects. These techniques can involve mathematical operations on the pixel values, filtering to remove noise or enhance features, or machine learning algorithms trained on large datasets of images.

The size of an image is determined by its dimensions, which are typically expressed as the number of rows and columns of pixels in the image. In python the image size is captured using the shape function, for example, the output comes like this for an image, 120x225 then the image has 120 pixels in each row and 225 pixels in each column, for a total of 27,000 pixels.

There are basically two images are given as part of the experiment purpose.

Sample Image:

|  | IImage |
| --- | --- |

An image, which is having the dark region present, that image is showing the damaged eye, while an image which containing almost zero value for the dark region then that eye is primarily healthy.

Since the image is 2- Dimensional in the case of a gray image, while the color image it is 3- dimensional matrix. So for the image manipulation, here in the experiment, the RGB image to GRAY Scale conversion has been done.

Results

As a part of the experiment, here python programming has been used. Since we are dealing with the images, matplotlib library is useful and for the inbuilt algorithm of the image processing technique we have used the openCV library.

OpenCV is a popular open-source library for computer vision and image processing tasks. OpenCV provides a comprehensive set of tools and algorithms for image and video processing, including image filtering, feature detection, object recognition, and machine learning.

TensorFlow was initially tested for image processing and analysis, however , TensorFlow is a framework for machine learning and it is more optimal with generalproblems such as regression, clustering and classifi cation. Therefore OpenCV, a library forcomputer vision, was used instead as this is more appropriate for shape and colour detection.

Importing the image and store it in the image array, which will act a matrix:

Example Image:

Sample Image 1:

image\_name\_in = "/content/CHMP2B-FTDfly-eye(1) (1).png"

# imread assumes BGR colour space

image = cv.imread(image\_name\_in)

RGB\_img = cv.cvtColor(image, cv.COLOR\_BGR2RGB)

plt.imshow(RGB\_img) # plot the image

plt.title('RGB Image'), plt.xticks([]), plt.yticks([])

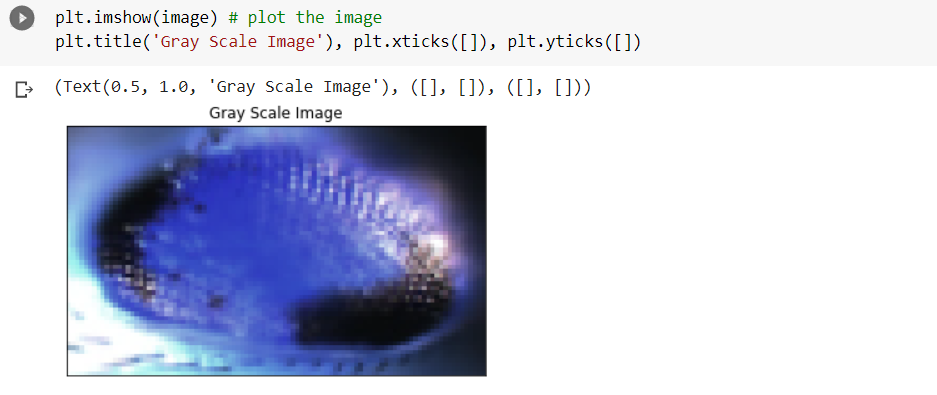
print("Size of the image is:",image.shape )



Converting the image to gray scale image:

plt.imshow(image) # plot the image

plt.title('Gray Scale Image'), plt.xticks([]), plt.yticks([])



Sample Image 2:

Considering the second image to test the program outcome.

image\_name\_in = "/content/WTflyeye(1)png (1).png"

# imread assumes BGR colour space

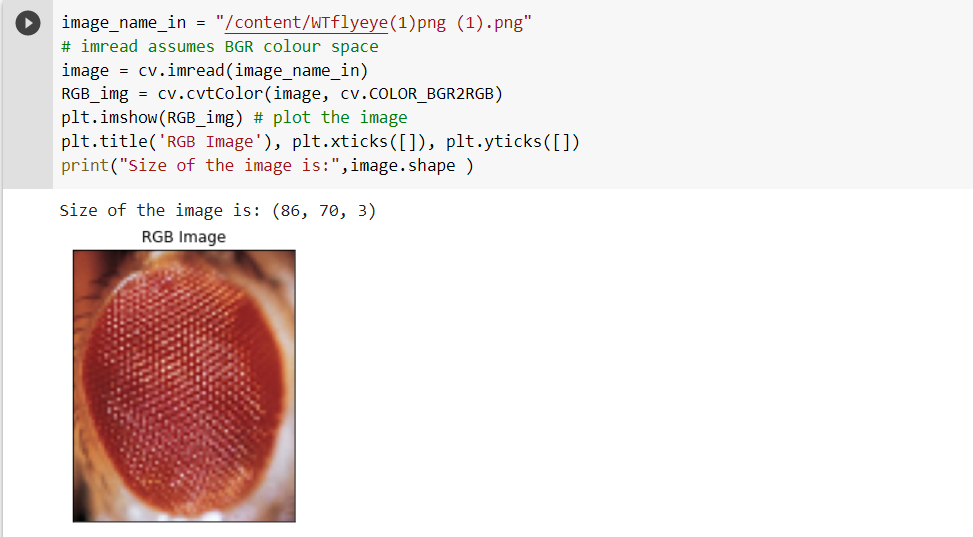
image = cv.imread(image\_name\_in)

RGB\_img = cv.cvtColor(image, cv.COLOR\_BGR2RGB)

plt.imshow(RGB\_img) # plot the image

plt.title('RGB Image'), plt.xticks([]), plt.yticks([])

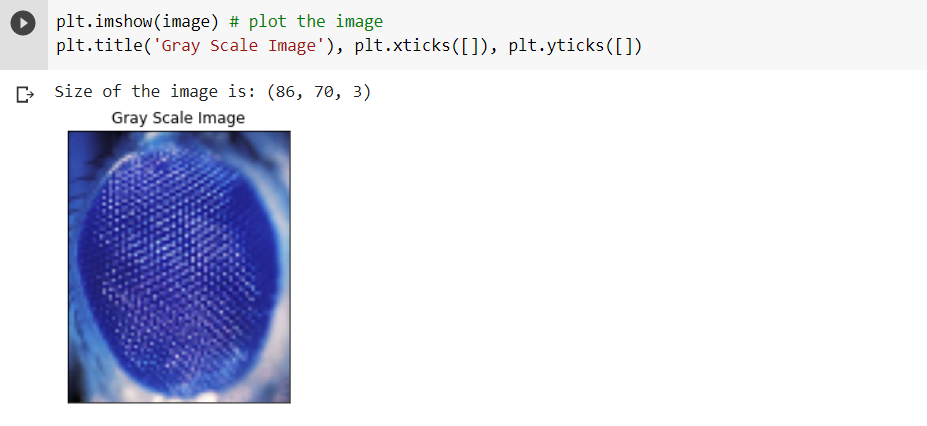
print("Size of the image is:",image.shape )



Converting the image to gray scale image:

plt.imshow(image) # plot the image

plt.title('Gray Scale Image'), plt.xticks([]), plt.yticks([])



Here, the sample images are in the form of color images, so image.shape function will return the 3-dimensional array, the last dimension represent the RGB content of image.

Using the OpenCV library we are going to detect, black region presence and if it is found then eye is not healthy and if the black region is not present then eye is categorized as a healthy image.

RGB image histogram is a graphical representation of the distribution of pixel values in an image's red, green, and blue color channels. The histogram displays the frequency of occurrence of each pixel value in the image, which can provide insights into the image's overall brightness, contrast, and color balance.

Sample Image and code:

def plotHistogram(a):

plt.figure(figsize=(10,5))

plt.subplot(1,2,1)

plt.imshow(a)

plt.axis('off')

histo = plt.subplot(1,2,2)

histo.set\_ylabel('Count')

histo.set\_xlabel('Pixel Intensity')

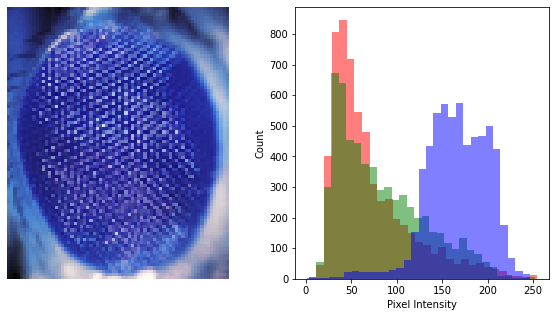
n\_bins = 30

plt.hist(a[:,:,0].flatten(), bins= n\_bins, lw = 0, color='r', alpha=0.5);

plt.hist(a[:,:,1].flatten(), bins= n\_bins, lw = 0, color='g', alpha=0.5);

plt.hist(a[:,:,2].flatten(), bins= n\_bins, lw = 0, color='b', alpha=0.5);

plotHistogram(image)



For sample image the intensity is skewed for the 0-50 pixel value, so we can say that black region is there in image.

Console output for the image 1:

### Returns an informative message categorising the eye (to the console or saved to a file

if((thresh.size - np.count\_nonzero(thresh))>=50):

print("eye is showing damage")

else:

print("eye is primarily healthy")

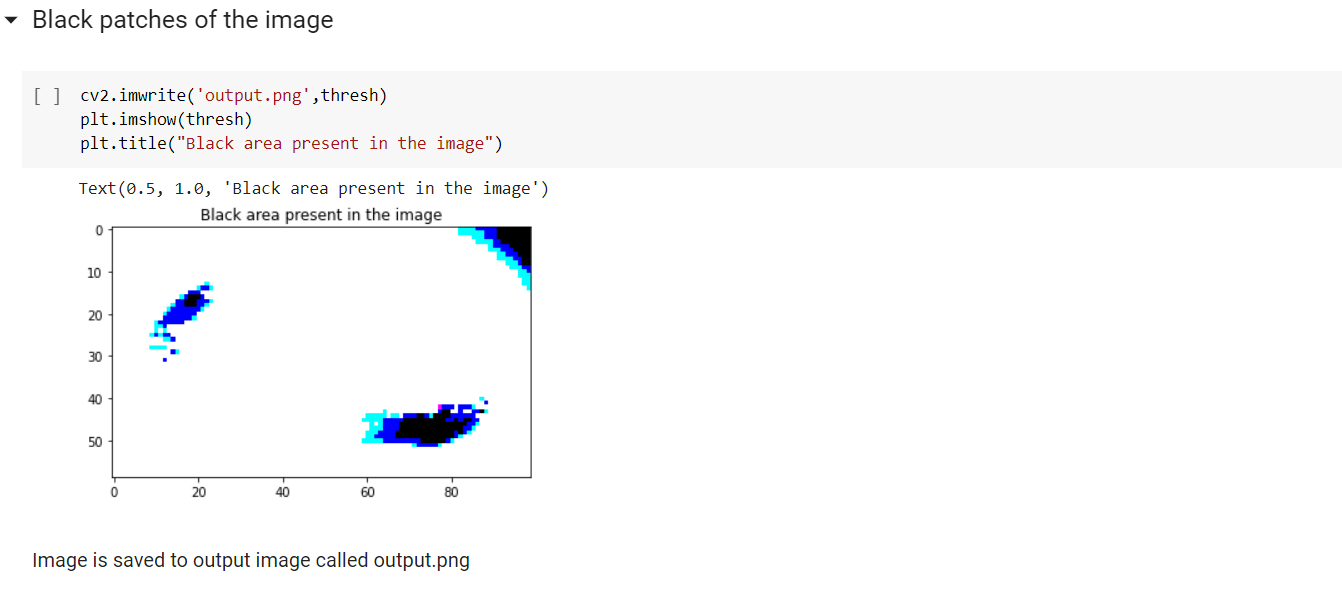
Output of the above is : eye is primarily healthy

Black patches of the image

cv2.imwrite('output.png',thresh)

plt.imshow(thresh)

plt.title("Black area present in the image")



Experiment for the sample image 2:

def plotHistogram(a):

plt.figure(figsize=(10,5))

plt.subplot(1,2,1)

plt.imshow(a)

plt.axis('off')

histo = plt.subplot(1,2,2)

histo.set\_ylabel('Count')

histo.set\_xlabel('Pixel Intensity')

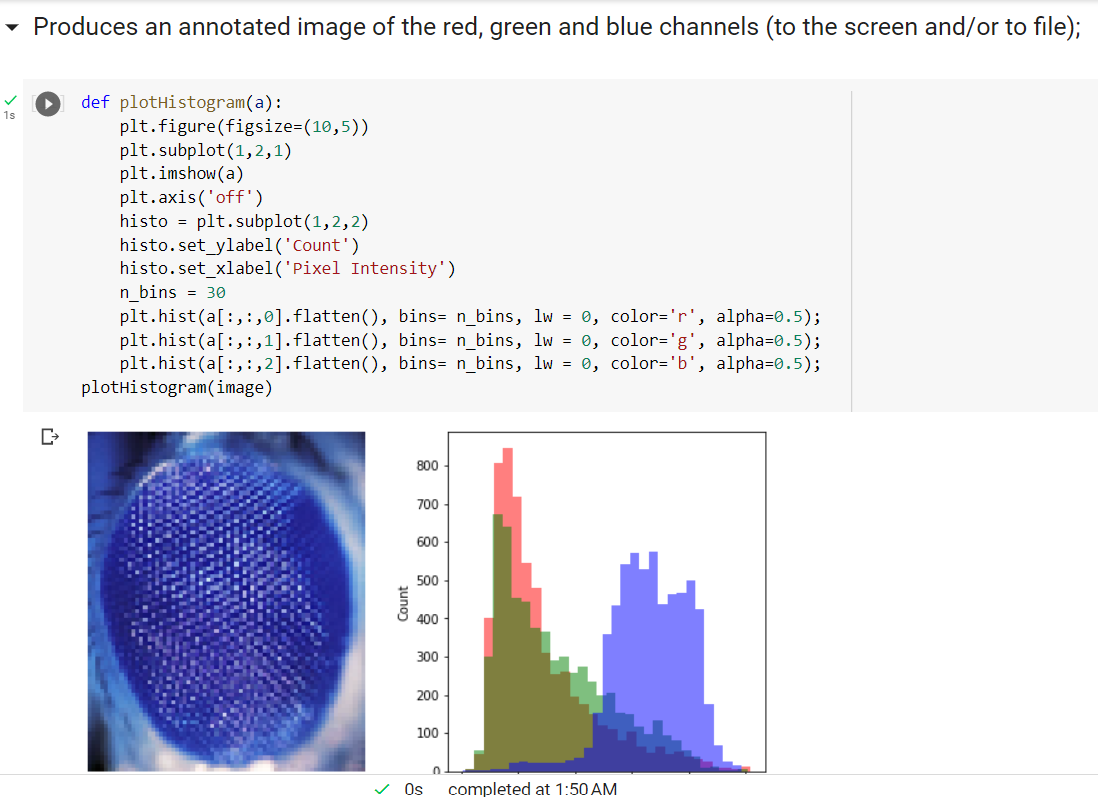
n\_bins = 30

plt.hist(a[:,:,0].flatten(), bins= n\_bins, lw = 0, color='r', alpha=0.5);

plt.hist(a[:,:,1].flatten(), bins= n\_bins, lw = 0, color='g', alpha=0.5);

plt.hist(a[:,:,2].flatten(), bins= n\_bins, lw = 0, color='b', alpha=0.5);

plotHistogram(image)



For sample image the intensity is evenly distributed across the pixel value, so we can say that absence black region in image.

if((thresh.size - np.count\_nonzero(thresh))>=50):

print("eye is showing damage")

else:

print("eye is primarily healthy")

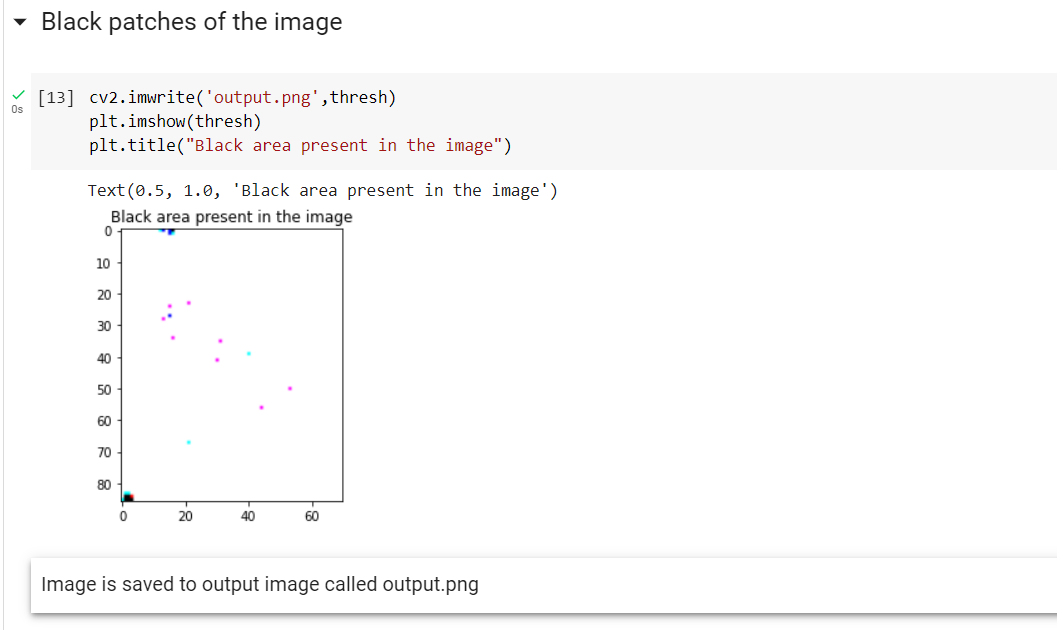
Output will be eye is primarily healthy

### Black patches of the image:

cv2.imwrite('output.png',thresh)

plt.imshow(thresh)

plt.title("Black area present in the image")



For the image 1, since the black region is present so it is showing output as eye is showing damage while for the image 2, since there is an absence of black region, so it is showing that eye is primarily healthy.

Evaluation/Applications

Image processing methods can be used to detect eye diseases, the various example applications for the identification and diagnosis are mentioned below.

Image processing is a technique used to enhance and analyze digital images. It involves the use of algorithms and software to manipulate and extract information from images. One application of image processing is in the field of ophthalmology, specifically in the analysis of retinal images.

Retinal imaging is the process of capturing high-resolution images of the retina, which can be used to diagnose and monitor a range of eye diseases, including diabetic retinopathy, macular degeneration, and glaucoma. These images can be analyzed using various image processing techniques to extract useful information about the retina, such as blood vessel density and retinal thickness.

Cataract detection: Cataracts are a common cause of vision loss, particularly in older adults. Image processing techniques can be used to analyze images of the lens and detect signs of cataracts, such as opacity and cloudiness. Early detection and treatment of cataracts can help prevent vision loss and improve outcomes.

Automated screening for retinopathy of prematurity (ROP): ROP (Yinsheng and Wang, 2019) is a condition that affects premature infants and can lead to vision loss or blindness if left untreated. Image processing techniques can be used to analyze retinal images and automatically detect signs of ROP, such as abnormal blood vessel growth. Automated screening can help identify infants who require further evaluation and treatment, leading to earlier detection and better outcomes.

Corneal disease diagnosis: The cornea is the clear outer layer of the eye, and diseases such as keratoconus and corneal dystrophies can cause vision problems. Image processing techniques can be used to analyze corneal topography images and detect abnormalities in the shape and curvature of the cornea. This can aid in the diagnosis and treatment of corneal diseases.

Uveitis diagnosis: Uveitis is a condition that causes inflammation in the uvea, the middle layer of the eye. Image processing techniques can be used to analyze images of the iris and detect signs of uveitis, such as inflammation and changes in the pattern of the iris.

Here, as a part of the experiment, we have only two samples of the images given. But if we want to build the model then we need to have an image dataset, using a machine learning algorithm, we can generate the model and for the new feed image, we can conclude that the eye is healthy or has some disease. So based on the machine-generated and human diagnosis, patients can avail of proper treatment for the disease.

**References**

A.G., B. (1994). Improvements on Littmann's method of determining the size of retinal features by fundus photography. *springer*, *232*(2), 361–367. https://doi.org/10.1007/BF00175988

Fujimoto, J. G. (2000). Optical Coherence Tomography: An Emerging Technology for Biomedical Imaging and Optical Biopsy. *Neoplasia*, *v.2*(2(1-2)), 9–25. Neoplasia (New York, N.Y.). https://doi.org/10.1038/sj.neo.7900071

Yannuzzi, L. A. (n.d.). ophthalmology. *Fluorescein Angiography Complication Survey*, *93*(5), 611-617. ISSN 0161-6420. https://doi.org/10.1016/S0161-6420(86)33697-2.

Yinsheng and Wang, Z. (2019). Development of an Automated Screening System for Retinopathy of Prematurity Using a Deep Neural Network for Wide-Angle Retinal Images. *IEEE Access*, *7*(2019), 10232-10241. IEEE. 10.1109/ACCESS.2018.2881042