Volume: 7 Issue: V Month of publication: May 2019

DOI: https://doi.org/10.22214/ijraset.2019.5282

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue V, May 2019- Available at www.ijraset.com

Contact Lens Detection for Security

Hazel San Loverez Patilano[1], Gerald Cayabyab[2], Ma. Cristina Aragon[3], Ruji P. Medina[4]
[1, 2, 3, 4]Graduate Studies, Technological Institute of the Philippines

Abstract: This paper feature an efficient extraction and matching approach on eye with or without contact lens to innovate authentication process. The proposed feature extraction and matching approach is develop based on FGM (Feature Group Matching) which extract the key points and descriptor of each images using the two feature extraction algorithm which are SIFT (Scale-Invariant Feature Transform) and SURF(Speeded-Up Robust Features) and to match the key points and descriptors using FLANN based matcher. The major advantages of the developed approach are the simplicity and accuracy as compared to the prior extraction and matching approaches developed for contact lens detection images. The new used approach was tested in AMD Ryzen 5 2500U with 2GHz processor, 4GB Mobile RAM, dual graphic card and GPU rendering of RX 560 x with 5GB. The experimental results obtained from IIITD- Contact lens Iris Database and publicly available images and respectively achieved better performance in terms of simplicity and accuracy as compared to the previous proposed methods and Filipino Iris Database as an added contribution to the field of contact lens.
***Keywords:* Contact Lens, SIFT Algorithm, SURF Algorithm, FGM Algorithm, FLANN based matcher**

I. INTRODUCTION

It is well-known how the processing of an image is time-consuming, particularly with recent high-resolution images. Besides, many algorithms require the simultaneous processing of many images and often in real time [1]. Nevertheless, during the last year, the applications of computer vision have increased greatly in many different contexts. This was due to the availability of more powerful hardware rather than to more efficient algorithms.
In human vision, which is what one, would normally expect a computer to be able to mimic. It is more complex process than people might imagine it is. The human eye is composed of a lens that is flexible. A human eye belongs to a general group of eye found in nature called “camera type eye” just a camera lens focuses light into filters. A human eye structure is analogous to that of a camera [2]. There is the basic structure of the eye that shown in Fig 1.

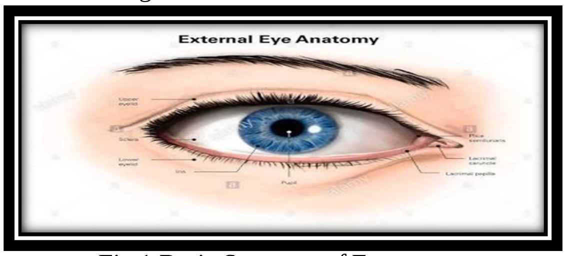


Fig 1 Basic Structure of Eye

According to the study of Ph.D. John Daugman Iris is considered to be most reliable and accurate biometric trait for person identification because irises patterns provide rich texture that is highly discriminative between individuals and stable during aging of a subject. Unfortunately, counterfeit iris texture can be presented to the acquisition device, thus iris-based biometric system are prone to sensor level attacks (spoofing and obfuscation) like system using any other biometric modalities. Counterfeit irises can be presented in many forms including artificial glass and plastic eyes, photographs and videos and printed textured contact lenses [3].
The irises use for iris recognition has pursued several anti-spoofing approaches. The most popular involves designing a set of texture filters and a classifier that can be used to categorize an iris image as representing either natural iris texture or contact lenses [4].
Since contact lenses are designed for reshaping the appearance (color and texture) of the iris tailored to wearer’s preferences. Example transforming one’s apparent eye color from brown to blue, they are also referred to as cosmetic contact lenses. A contact lens is used to correct eyesight and for cosmetic reasons. The color and texture in contact lens can superimpose on natural iris pattern. This contact lens has the ability to change the optical properties of the eye and thereby it can reduce the overall accuracy of iris biometric systems. Spoofing is the method in which one tries to hide their own identity. So contact lens detection is an important anti-spoofing method [5].

The various stage of iris recognition include taking the input image, detecting the iris image, extracting feature to enhance the robustness and finally matching the extracted image with that in the database to recognize the correct iris of the intended individual [6].
Detection of the presence of the contact lenses is the first step to improving the usability and reliability of iris recognition for contact lens wearers. One simple solution might be to change the decision threshold when a clear non-textured contact lens is detected such that the false non-match rate (FNMR) is identical to user who does not wear lenses [7], [8], [9], [10].
By wearing contact lenses, the accuracy of iris recognition was degraded. The table 1 show the accuracy of iris recognition by using MLBP algorithm result for iris images belong to three categories/classification N-N (Normal Eye), T-T (Colored Lens), S-S (Soft Lens) [11].

TABLE I TABLE OF ACCURACY

DATABASE	CLASSIFICATION TYPE	MLBP Algorithm
IIITD Cogent Contact Lens Database	T – T	66.83
	S –S	94.91
	N – N	56.66
	TOTAL:	73.01

Feature Extraction and Matching is to extract feature primitives that are easy to match from two images, such as point-like feature, a linear feature or regional characteristics. In order to guarantee the matching speed, reliability and stability, it's important to select appropriate images feature algorithm [12]. Feature Extraction accuracy has some drawbacks and many researchers continue to present different algorithms to mitigate those problems. Previously Co-occurrence Matrix feature algorithm applied by [13] that the mean and range of the values served as a feature of 14 measures of textual features based on co-occurrence matrix that shows redundancies. Some studies use Local Binary Pattern algorithm applied by [14] that extract rotation invariant texture feature from a local region that fails to classify the image because of LBP codes of two texture image which is composed of two LBP micro-patterns that exhibits different texture information. In [15] applied Zernike Moments algorithm that can capture the changes of shape between a spoofed and normal iris image but it requires lower computational precision to represent images to the same accuracy as a regular moment. The two researchers [16], [17] applied Modified Local Binary Pattern algorithm to detect contact lens in iris recognition but it is sensitive to noise mainly in an identical region and it supports only a binary level assessment for encoding.

II. RELATED LITERATURE

Feature Extraction is one of the important steps in pattern recognition. It extracts a set of descriptors, various characteristic attributes, the relevant information associated to form and representation of input pattern [18].
In detecting contact lens and extracting from the segmentation iris image is important to verify an image that wearing a contact is same to person who not wearing a contact lens. In Iris is one of the most reliable and accurate biometric modalities due to the highly unique character of iris tissue structure [19]. Flom and Safir have shown that iris texture is unique for each indi11vidual in 1987 [9]. It also proved that iris recognition system performance degrades when an iris without soft contact lens [20], [21]. John Daugman patented the first successful iris recognition algorithm in 1994 [22], it was based on a test of statistical independence of the phase of Gabor wavelets fitted on a grid of location superimposed on a pseudo-polar transformation of the iris texture. That basic design remains the dominant recognition method for years. It has been used successfully in numerous applications including national ID projects and border security. The success of large-scale identity application using iris recognition, in turn, means there are now individuals who, by means of presentation attack or spoofing, can gain unauthorized access to locations or resources or escape recognition as a person of interest. Some typical iris spoofing attacks are printed iris images, textured contact lens and synthetic creation of iris images. Some relevant works directly related to the three class of iris image problem which is to propose of classifying iris image in (color) textured contact lenses, soft (prescript or clear), contact and non-lenses.
Textured contact lens, with advances in technology and low cost, contact lenses are gaining popularity around the world. Apart from being used for eyesight correction, they are increasingly being used for the cosmetic purpose as well. These textured (cosmetic) lenses cover the original texture of the iris with a thin textured lens which can severely degrade the performance of iris recognition system. Several studies [4], [15], [23] have demonstrated the need for detecting contact lens as both transparent (soft) and textured (cosmetic) lenses have been shown to affect iris recognition system.
Detection of the cosmetic lens is an easier problem as compared to detection of soft contact lens because former has a specific texture present over it. The soft contact lens has no such texture, no color that shown in fig 2. Hence techniques for detecting

cosmetic contact lens detection cannot be used to detect soft contact lens. In most of the NIR image, it is very difficult to detect soft contact lens even by the human eye. But lens boundary is fairly visible because of specular reflection which can be used to identify the contact lens boundary. Algorithm for detection of soft contact and cosmetic contact lenses has been proposed [24]. The algorithm proposed in [7] is based on traditional edge detection exploiting sharp changes in pixel intensity.

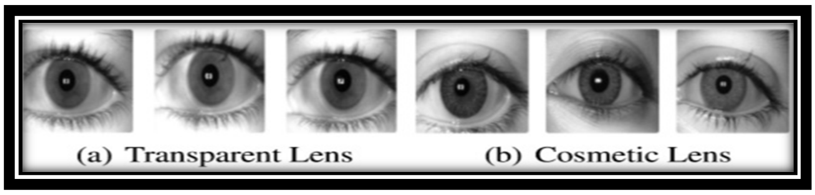


Fig 2 Type of Contact Lens

With all the different algorithms used for contact lens detection, Feature Group Matching Algorithm (FGM) was not used. FGM algorithm is a new used algorithm to be used for contact lens detection. FGM algorithm introduced by [25], it is a novel approach to local matching to select stable feature and obtain a more reliable similarity value between two images. Feature Group Matching algorithm used descriptors to detect a set of key points in the image and provide a description on vectors which is Scale Invariant Feature Transform (SIFT) [26], [27] and Speeded Up Robust Features (SURF) [28] algorithm which represent a starting point for the FGM elaboration. Local Feature matching has become commonly used method to compare images; despite it is highly probable that at least some of the matching it detects is incorrect.

A new technique for matching an image was developed of Knowledge Management System that used Brute-Force algorithm to search for knowledge on the SECI Model in Knowledge Management System [30]. The Knowledge Management system modelling done and tested by discovery system those search for knowledge of data and information. It performs data collection knowledge and modelling string matching and conduct modelling with Brute-Force algorithm [30].

III. DESIGN ARCHITECTURE

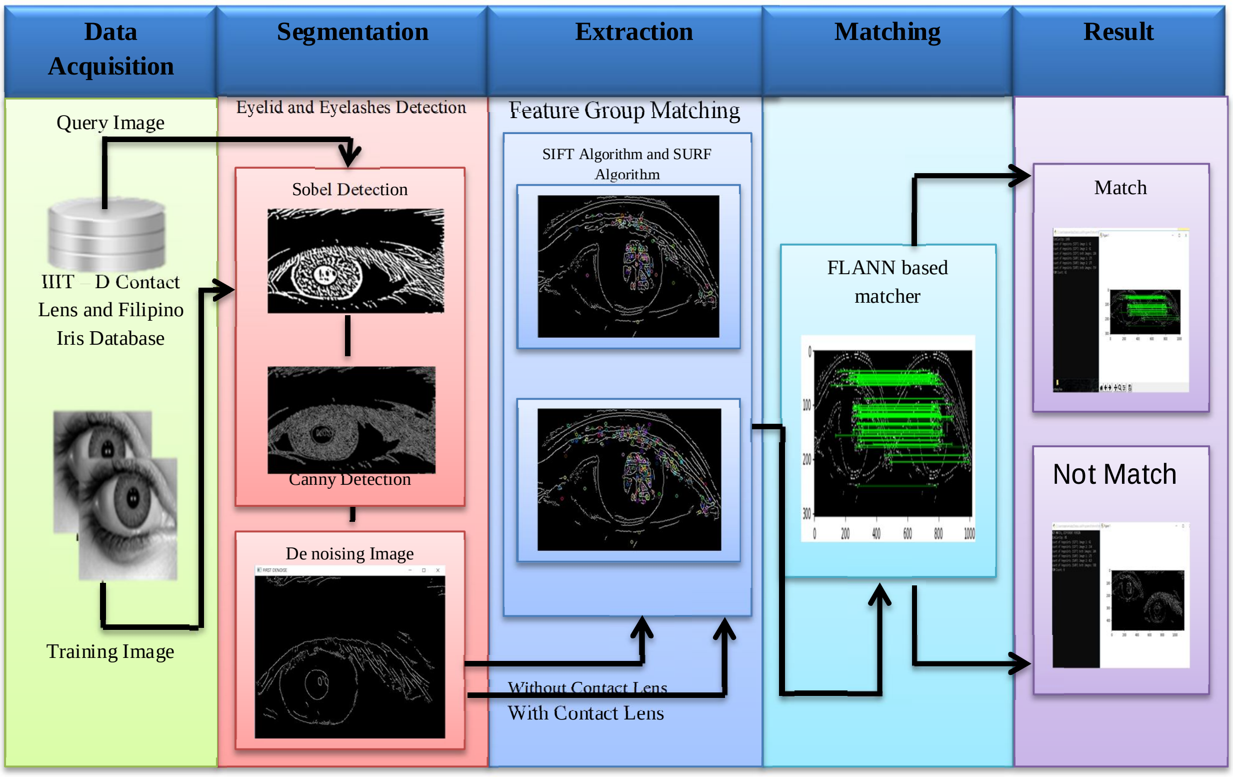


Fig 3 Design Architecture

The process went through four major processes: Data Acquisition, Segmentation, Extraction and Matching.

A. Data Acquisition

In Data acquisition, a license agreement has been downloaded to be signed by the authorized signatory for research and sent by the research institution to the Indraprastha Institute of Information Technology, Delhi requesting to use their iris images known as IIIT-Delhi Contact Lens Iris Database [10] for the proposed study. Upon approval, the data will be collected from their image repository. And my contribution database called Filipino Contact Lens Iris Database that comprised 120 images of 20 Filipino persons. It consist of image of both left and right iris, some iris images wear contact lens and without contact lens.

B. Iris Segmentation

The images will undergo Iris Segmentation which applied Sobel detection to search regions to detect the eyelids. Iris localization is processing to isolate that required iris region from the image taken in the database. Canny edge detection is performed to detect the edges [5]. After detecting the edges, the image applied image smoothing techniques like Gaussian Blurring, Median Blurring [29] for removing high frequency content of noise and edges from the image resulting in edges.

C. Extraction

The Feature Group Matching was experimented by the group of [25] and they called it as feature extraction and matching algorithm. It is a novel approach to local feature matching to select stable feature and obtain a more reliable similarity value between two images.

Feature Group Matching algorithm used descriptors to detect a set of key points in the image and provide a description on vectors which is Scale Invariant Feature Transform (SIFT) [26], [27] and Speeded Up Robust Features (SURF) [28] algorithm which represent a starting point for the FGM elaboration. Local Feature matching has become commonly used method to compare images; despite it is highly probable that at least some of the matching it detects is incorrect.

A key point descriptor is created by first computing the gradient magnitude and orientation at each image sample point region around the key point location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These sample are then accumulated into orientation histogram summarizing the contents over 4 x 4 sub regions, as shown on the right, with the length or each arrow corresponding the sum of the gradient magnitudes near that direction within region. This fig. 4 shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the sample use 4x4 descriptors computed from 16x16 sample array.

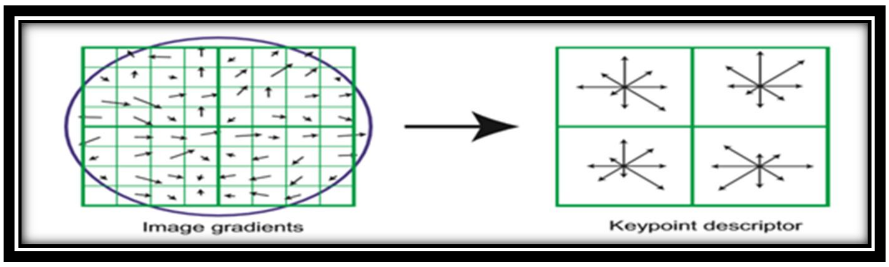


Fig 4 Computation of key point descriptor

In order to be invariant to rotation, identify a reproducible orientation for the interest points. For that purpose, first calculate the Haar-wavelet responses in x and y-direction, show in fig 5, and this is a circular neighbourhood of radius 6s around the interesting point, with s the scale at which the interesting point was detected.

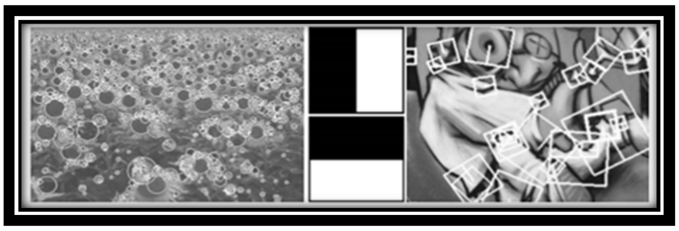


Fig 5 Left: Detected interest points for a sunflower. Middle: Haar wavelet types used SURF. Right: Detail of the Graffiti scene showing the size of the descriptor window at different scales.

The FGM elaboration can be decomposed into two main functions: one that calculates the groups by analysing the key points of the image and another that calculates the descriptor for each group previously found. The second is not suitable to introduce a parallelization because it needs frequent exchanges of memory and it has many if statements. On the contrary, the first function, which constructs the groups, involves two steps that can be both parallelized:

1) Calculation of the distance of each point from a reference point (center of the group).
2) Finding the nearest point (example: with maximum distance to the center of the group).

D. Matching

After getting the key points of each images using FLANN based matcher to eliminate and match the key points that detect by FGM. A FLANN (Fast Library for Approximate Nearest Neighbor) it perform a quick and efficient matching [32]. FLANN builds an efficient data structure (KD-Tree) that will be used to search for an approximate neighbour.

E. Experimental Result and Evaluation

Opencv-python Interpreter and IIIT-D Contact Lens Iris Database [10] and Filipino Contact Lens Database will be used to evaluate the proposed the contact lens detection for security based on their applicability, simplicity and accuracy. The extraction and matching will be evaluated against the previously proposed algorithm if the accuracy will vary from the previous record by presenting the results of previously conducted study.

1) *Segmentation Results:* The following figures show how to detect the eyelids using Sobel algorithm. After detecting the eyelid it will follow by Canny Edge algorithm is performed to detect the edges. The result image from detecting the edges it will undergo in removing the noise by using image smooth techniques. These images were experimented on AMD Ryzen 5 for segmentation.

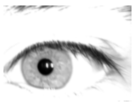
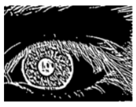
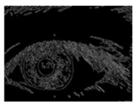
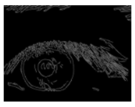
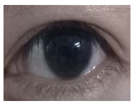
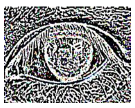
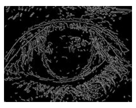
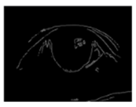
FOREIGN CONTACT LENS IRIS			
a. Original Images	b. Detect the eyelid	c. Detect the edges	d. Removing noise
			
FILIPINO CONTACT LENS IRIS			
			

Fig 6 Segmentation Image of Normal Eye: (a). Source image, (b.) detecting eyelid using sobel algorithm, (c) detecting edges using canny edges algorithm, (d.) removing noise in an image

The algorithm has been experimented using different eye images. Out of 202 folders that composed of 6570 iris image pertaining to 101 subjects. Both left and right iris images with contact lens (soft and transparent) and normal iris image from IIIT Delhi Contact Lens Iris Database [10] dataset and 120 iris images with and without contact lens from Filipino Contact Lens dataset, only three (3) images from IIIT Delhi Contact Lens Iris dataset and Filipino Contact Lens Iris dataset were randomly selected and used for the experiment and the rest of the images was use as a training data set. The aim of the study is to test he applicability and accuracy of the algorithm in different sizes wherein the dataset from IIIT-Delhi Contact Lens [10] and Filipino Contact Lens Iris composed of images having the same sizes. As shown in Fig. 6, the result shows the eyelid and edges is detected and the noise is removed in the terms of normal eye. While Fig. 7, the result shows the eyelid and edges is detected and the noise is removed includes the eye with colored and transparent contact lens.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue V, May 2019- Available at www.ijraset.com

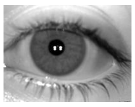
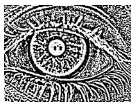
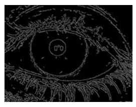
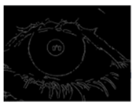
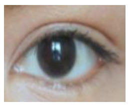
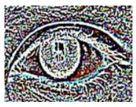
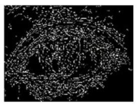
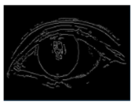
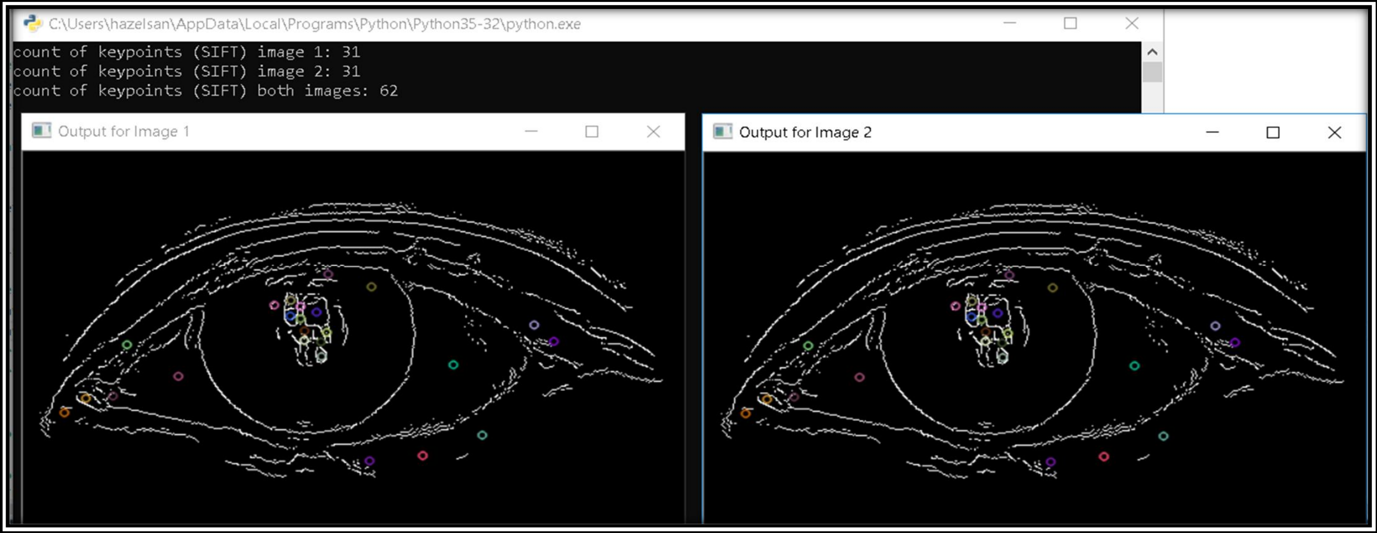
FOREIGN CONTACT LENS IRIS			
a. Original Images	b. Detect the eyelid	c. Detect the edges	d. Removing noise
			
FILIPINO CONTACT LENS IRIS			
			

Fig 7 Segmentation Image of Eye with Contact Lens

2) *Extraction Result:* The proposed feature extraction show how the Feature Extraction Matching (FGM) used descriptors to detect a set of key points in the image and provide a descriptor on vectors which is Scale Invariant Feature Transform (SIFT) [26], [27] and Speeded Up Robust Features (SURF) [28] algorithm which represent a starting point for the FGM elaboration.

It transform image data into scale-invariant coordinate relative to local features [18]. First, extract from a set of reference images by construct a SIFT object to detect the key point and draw them. Each key point is a special structure which has many attributes like its (x, y) coordinates, size of the meaningful neighbourhood, angle which specifies its orientation, response that specifies strength of key points.

In fig 8, the result show the detection of key point each images by using Scale Invariant Feature Transform and display the number of key point of each images that will perform the Feature Group Matching.


C:\Users\hazelsan\AppData\Local\Programs\Python\Python35-32\python.exe
Output for Image 1
Output for Image 2

```
count of keypoints (SIFT) image 1: 31
count of keypoints (SIFT) image 2: 31
count of keypoints (SIFT) both images: 62
```

Fig 8 Detecting key point using SIFT

In fig 9 shown three categories results in detecting the key point using SIFT in same eye. A shown Normal to Normal eye, B shown Normal to Contact Len and C has shown Contact lens to Contact Lens. Fig 10 shown three categories results in detecting the key point using SIFT in different eye. A shown Normal to Normal eye, B shown Normal to Contact Len and C has shown Contact lens to Contact Lens.

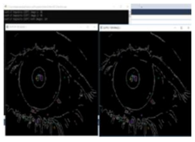
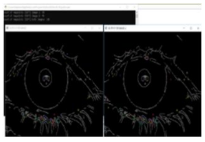
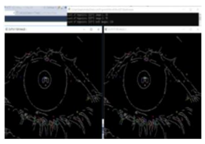
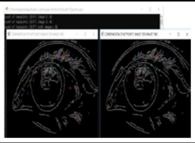
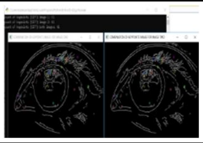
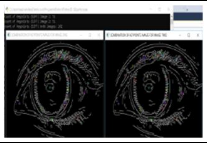
FOREIGN EYE		
A. Normal to Normal	B. Normal to Contact Len	C. Contact Lens to Contact Lens
		
FILIPINO EYE		
		

Fig 9 Detecting the key point using SIFT in same eye

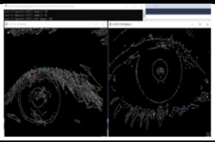
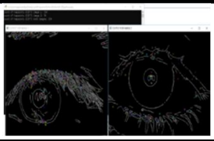
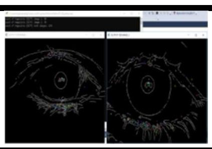
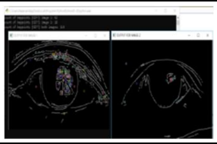
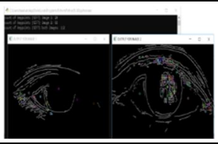
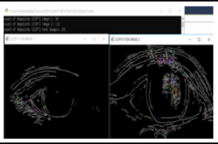
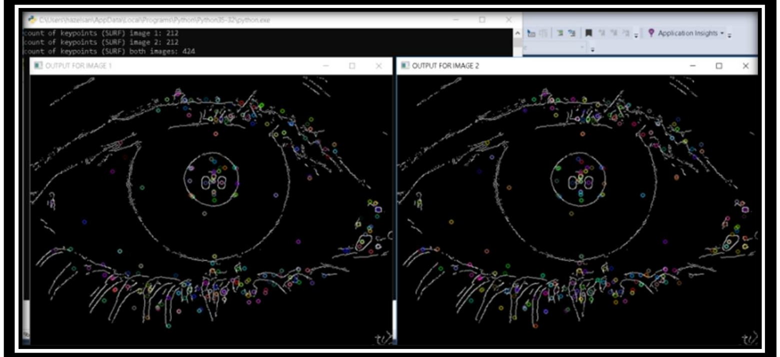
FOREIGN EYE		
A. Normal to Normal	B. Normal to Contact Lens	C. Contact Lens to Contact Lens
		
FILIPINO EYE		
		

Fig 10 Detecting the key point using SIFT in different eye

In SIFT; Lowe approximated Laplacian of Gaussian with Difference of Gaussian for finding scale-space. SURF goes a little further and approximate LoG with Box Filter. One big advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images. And it can be done in parallel for different scales. Also rely on determinant of Hessian matrix for both scale and location.

For key point descriptor, in fig 11 shown the SURF uses Wavelet responses in horizontal and vertical direction (again, it use of integral images makes things easier). A neighbourhood of size 20sX20s is taken around the key point where s is the size. It is divided into 4x4 sub regions. For each sub region, horizontal and vertical wavelet responses are taken and a vector is formed like this, v =$(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. This when represented as a vector gives SURF feature descriptor with total 64 dimensions. Lower the dimension, higher the speed of computation and matching, but provide better distinctiveness of features.



```
count of keypoints (SURF) image 1: 212
count of keypoints (SURF) image 2: 212
count of keypoints (SURF) both images: 424
```

C:\Users\hazelsan\AppData\Local\Programs\Python\Python35-32\python.exe
OUTPUT FOR IMAGE 1
OUTPUT FOR IMAGE 2
Application Insights

Fig. 11 Detecting the key points using SURF

For distinctiveness, SURF feature descriptor has an extended 128 dimension version. The sum of d_x and $|d_x|$ are computed separately for $d_y < 0$ and$d_y \geq 0$. Similarly, the sum of d_y and $|d_y|$ are split up according to the sign of d_x, thereby doubling the number of features. It doesn't add much computation complexity. OpenCV supports both by setting the value of flag extended with 0 and 1 for 64-dim and 128-dim respectively (default is 128-dim)

```
image1=cv2.drawKeypoints(canny,kp,canny1,None, flags = 0)
image2=cv2.drawKeypoints(canny,kp0,canny1,None,flags=0)
```

Another improvement is the use of sign Laplacian (trace of Hessian Matrix) for underlying interest point. It adds no computation cost since it is already computed during the detection. The sign of Laplacian distinguishes bright blobs on dark background from the reverse situation. In matching stage, we only compare features if they have the same type of contrast (as show in fig 11). This minimal information allows for faster matching without reducing the descriptor's performance.
In fig 12, shown the three categories results in detecting the key point using SURF in same eye. A shown Normal to Normal eye, B shown Normal to Contact Len and C has shown Contact lens to Contact Lens. Fig 13, shown three categories results in detecting the key point using SURF in different eye. A shown Normal to Normal eye, B shown Normal to Contact Len and C has shown Contact lens to Contact Lens.

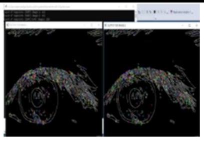
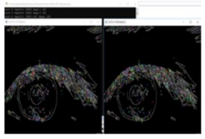
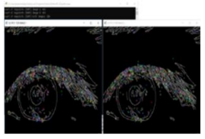
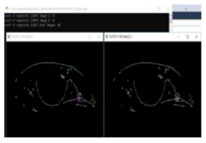
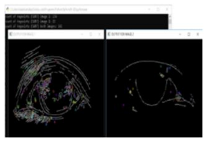
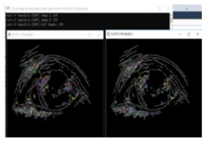
FOREIGN EYE		
A. Normal to Normal	B. Normal to Contact Len	C. Contact Lens to Contact Lens
		
FILIPINO EYE		
		

Fig 12 Detecting the key point using SURF in same eye

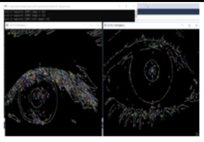
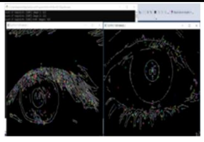
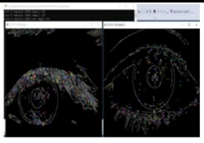
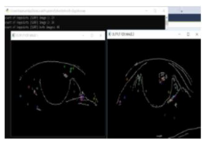
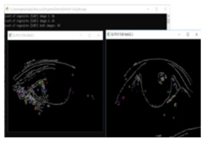
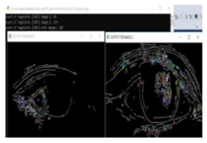
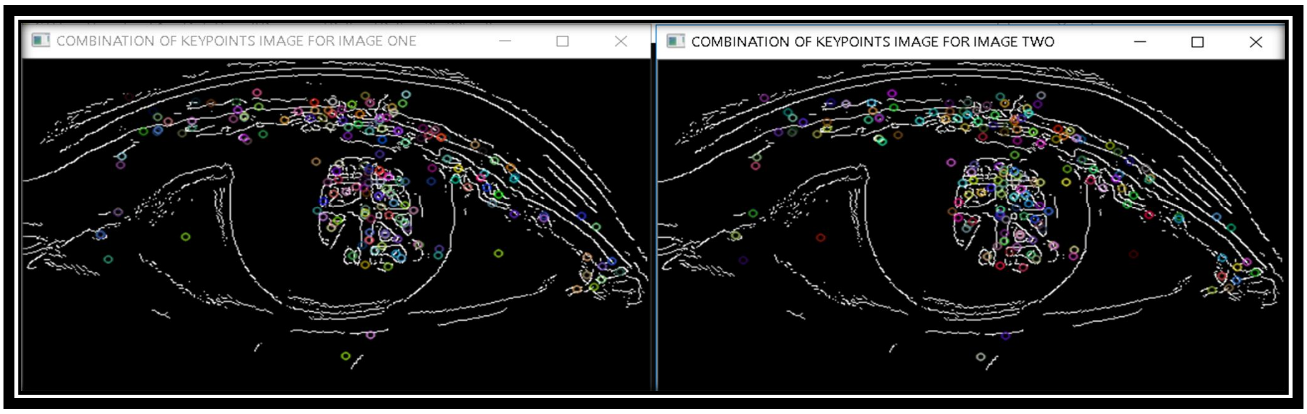
FOREIGN EYE		
A. Normal to Normal	B. Normal to Contact Len	C. Contact Lens to Contact Lens
		
FILIPINO EYE		
		

Fig 13 Detecting the key point using SURF in different eye

The result of group of key points using FGM and it shown in fig 14 the group of key point using the two extraction algorithm which is SURF and SIFT. Using the function of drawKeypoint() in opencv which will draws the small circle on the location of key points and it will even shows the orientations.

```
imageSIFT=cv2.drawKeypoints(canny,kp1,canny,None)
imageSIFT1=cv2.drawKeypoints (canny1,kp0,canny1,None)
imageSURF=cv2.drawKeypoints (canny,kp,canny,None)
imageSURF1=cv2.drawKeypoints (canny1,kp0,canny1,None)
keypointsTotal=cv2.drawKeypoints (canny1,kp0,kp2,canny1,None)

cv2.imshow("COMBINATION OF KEYPOINTS IMAGE FOR IMAGE ONE",keypointsTotal)
cv2.imshow("COMBINATION OF KEYPOINTS IMAGE FOR IMAGE TWO",keypointsTotal)
```

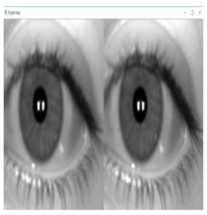
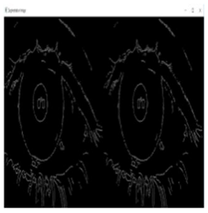
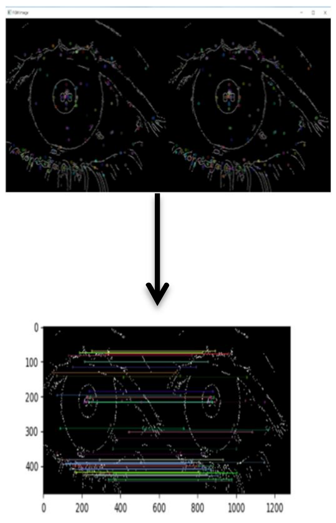

COMBINATION OF KEYPOINTS IMAGE FOR IMAGE ONE
COMBINATION OF KEYPOINTS IMAGE FOR IMAGE TWO
Fig 14 Group of Key points

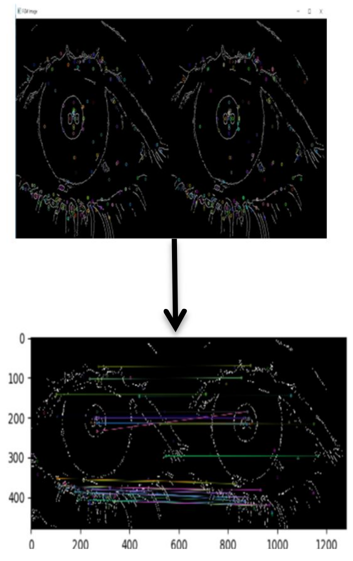
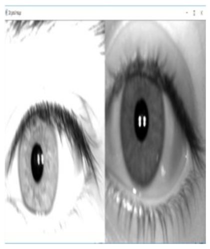
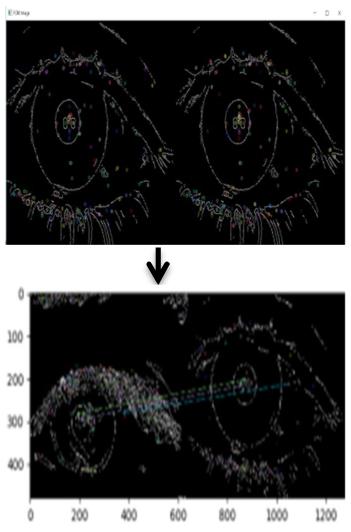
A new image is matched by individually comparing each feature from the new image to another image and finding candidate matching features based on the Ratio test by [31] distance of their feature vectors.
The result of finding and calculated the distance of each point from a reference point in Table 2.And using the Brute-Force Algorithm it can match if the key points of one image to another image are match. Distance is calculated using OpenCV python software as described below:

```
good=[]

for m, n in matches:
   if m.distance<0.7*n.distance:
      good.append(m)
      print("COUNT DISTANCE",m)

min_match_count = 0
if len(kp1)<=len(kp2):
   min_match_count = len(kp1)
else:
   min_match_count=len(kp2)

if len(good)>MIN_MATCH_COUNT:
   src_pts = np.float32([kp1[m.queryIdx].pt for m in good]).reshape(-1,1,2)
   dst_pts = np.float32([kp2[m.trainIdx].pt for m in good]).reshape(-1,1,2)

   M, mask = cv2.findHomography(src_pts, dst_pts,cv2.RANSAC, 5.0)
   matchesMask = mask.ravel().tolist()
```

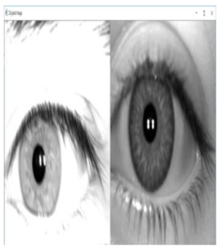
```
    h,w = canny.shape
    pts = np.float32([ [5000,2000] ]).reshape(-1,1,2)
    dst=cv2.perspectiveTransform(pts,M)

    img2 = cv2.polylines(img2, [np.int32(dst)], True, 255,3, cv2.LINE_AA)
else:
    print ("NOT MATCH, DIFFERENT PERSON") #%
(len(good),MIN_MATCH_COUNT)
    matchesMask = None

FGM=len(good)
print("FGM Count: " + str(int(FGM)))

img3 = cv2.drawMatches(canny, kp1, canny1,kp2, good, None)
plt.imshow(img3, 'gray'),plt.show()
```

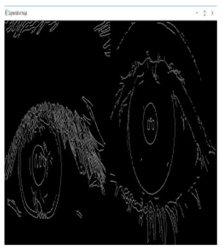
TABLE III
RESULT FOR FINDING THE DISTANCE

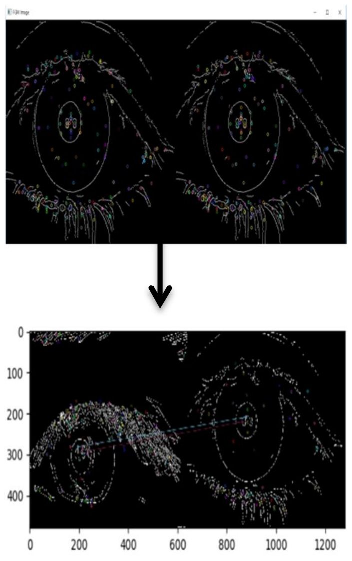
ORIGINAL	SEGMENTATION	KEYPOINTS IN FGM	DISTANCE (HEX REPRESENTATION)
FOREIGN EYE			
Without Contact lens SAME EYE			
		 TOTAL FGM : 58	1. COUNT DISTANCE <DMatch 0D3B3968> 2. COUNT DISTANCE <DMatch 0D7D76B0> 3. COUNT DISTANCE <DMatch 0E2FBED8> 4. COUNT DISTANCE <DMatch 01788F08> 5. COUNT DISTANCE <DMatch 0E3BE110> 6. COUNT DISTANCE <DMatch 0E3BE188> 7. COUNT DISTANCE <DMatch 0E3BE1B8> 8. COUNT DISTANCE <DMatch 0E3BE1D0> 9. COUNT DISTANCE <DMatch 0E3BE140> 10. COUNT DISTANCE <DMatch 0E3BE230> 11. COUNT DISTANCE <DMatch 0E3BE260> 12. COUNT DISTANCE <DMatch 0E3BE290> 13. COUNT DISTANCE <DMatch 0E3BE2C0> 14. COUNT DISTANCE <DMatch 0E3BE2F0> 15. COUNT DISTANCE <DMatch 0E3BE320> 16. COUNT DISTANCE <DMatch 0E3BE350> 17. COUNT DISTANCE <DMatch 0E3BE380> 18. COUNT DISTANCE <DMatch 0E3BE3B0> 19. COUNT DISTANCE <DMatch 0E3BE3E0> 20. COUNT DISTANCE <DMatch 0E3BE410> 21. COUNT DISTANCE <DMatch 0E3BE440> 22. COUNT DISTANCE <DMatch 0E3BE470> 23. COUNT DISTANCE <DMatch 0E3BE4A0> 24. COUNT DISTANCE <DMatch 0E3BE4D0> 25. COUNT DISTANCE <DMatch 0E3BE500> 26. COUNT DISTANCE <DMatch 0E3BE530> 27. COUNT DISTANCE <DMatch 0E3BE560> 28. COUNT DISTANCE <DMatch 0E3BE590> 29. COUNT DISTANCE <DMatch 0E3BE5C0> 30. COUNT DISTANCE <DMatch 0E3BE5F0> 31. COUNT DISTANCE <DMatch 0E3BE620> 32. COUNT DISTANCE <DMatch 0E3BE650> 33. COUNT DISTANCE <DMatch 0E3BE680> 34. COUNT DISTANCE <DMatch 0E3BE6B0> 35. COUNT DISTANCE <DMatch 0E3BE6E0> 36. COUNT DISTANCE <DMatch 0E3BE710> 37. COUNT DISTANCE <DMatch 0E3BE740> 38. COUNT DISTANCE <DMatch 0E3BE770> 39. COUNT DISTANCE <DMatch 0E3BE7A0> 40. COUNT DISTANCE <DMatch 0E3BE7D0> 41. COUNT DISTANCE <DMatch 0E3BE800> 42. COUNT DISTANCE <DMatch 0E3BE830> 43. COUNT DISTANCE <DMatch 0E3BE860> 44. COUNT DISTANCE <DMatch 0E3BE890> 45. COUNT DISTANCE <DMatch 0E3BE8C0> 46. COUNT DISTANCE <DMatch 0E3BE8F0> 47. COUNT DISTANCE <DMatch 0E3BE920> 48. COUNT DISTANCE <DMatch 0E3BE950> 49. COUNT DISTANCE <DMatch 0E3BE980> 50. COUNT DISTANCE <DMatch 0E3BE9B0> 51. COUNT DISTANCE <DMatch 0E3BE9E0> 52. COUNT DISTANCE <DMatch 0E3BEA10> 53. COUNT DISTANCE <DMatch 0E3BEA40> 54. COUNT DISTANCE <DMatch 0E3BEA70> 55. COUNT DISTANCE <DMatch 0E3BEAA0> 56. COUNT DISTANCE <DMatch 0E3BEAD0> 57. COUNT DISTANCE <DMatch 0E3BEB00> 58. COUNT DISTANCE <DMatch 0E3BEB30>

ORIGINAL	SEGMENTATION	KEYPOINTS IN FGM	DISTANCE (HEX REPRESENTATION)
FOREIGN EYE			
Without and with Contact Lens SAME EYE			
		 TOTAL FGM : 29	1. COUNT DISTANCE <DMatch 0CAE8440> 2. COUNT DISTANCE <DMatch 0D4ED110> 3. COUNT DISTANCE <DMatch 0D4ED188> 4. COUNT DISTANCE <DMatch 0D4ED1D0> 5. COUNT DISTANCE <DMatch 0D4ED260> 6. COUNT DISTANCE <DMatch 0D4ED2C0> 7. COUNT DISTANCE <DMatch 0D4ED320> 8. COUNT DISTANCE <DMatch 0D4ED3B0> 9. COUNT DISTANCE <DMatch 0D4ED4D0> 10. COUNT DISTANCE <DMatch 0D4ED500> 11. COUNT DISTANCE <DMatch 0D4ED530> 12. COUNT DISTANCE <DMatch 0D4ED6E0> 13. COUNT DISTANCE <DMatch 0D4ED800> 14. COUNT DISTANCE <DMatch 0D4ED830> 15. COUNT DISTANCE <DMatch 0D4ED8C0> 16. COUNT DISTANCE <DMatch 0D4ED8F0> 17. COUNT DISTANCE <DMatch 0D4ED980> 18. COUNT DISTANCE <DMatch 0D4EDA40> 19. COUNT DISTANCE <DMatch 0D4EDAD0> 20. COUNT DISTANCE <DMatch 0D4EDB30> 21. COUNT DISTANCE <DMatch 0D4EDB60> 22. COUNT DISTANCE <DMatch 0D4EDCB0> 23. COUNT DISTANCE <DMatch 0D4EDD10> 24. COUNT DISTANCE <DMatch 0D4EDD40> 25. COUNT DISTANCE <DMatch 0D4EDDA0> 26. COUNT DISTANCE <DMatch 0D4EDDD0> 27. COUNT DISTANCE <DMatch 0D4EDE00> 28. COUNT DISTANCE <DMatch 0D4EDE60> 29. COUNT DISTANCE <DMatch 0D4EDE90>
Without Contact lens DIFFERENT EYE			
		 TOTAL FGM : 3	1. COUNT DISTANCE <DMatch 0CE8EEC0> 2. COUNT DISTANCE <DMatch 0CEC7BD8> 3. COUNT DISTANCE <DMatch 0CEC7D28>

Without and with Contact lens
DIFFERENT EYE





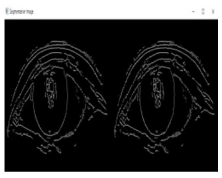


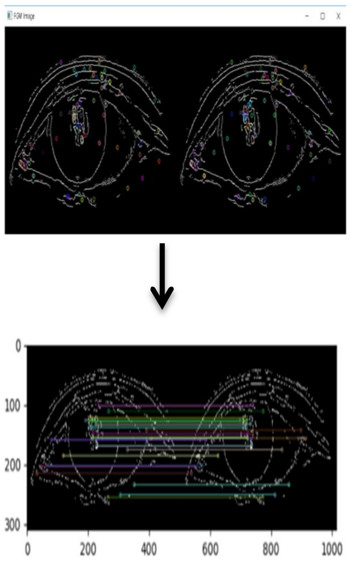
TOTAL FGM : 2

1. COUNT DISTANCE <DMatch 0D10DEC0>
2. COUNT DISTANCE <DMatch 0D149D28>

FILIPINO EYE

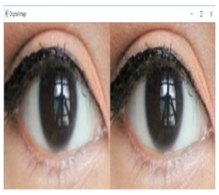
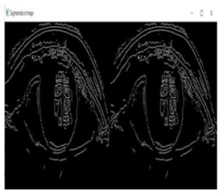
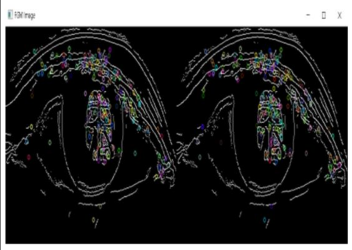
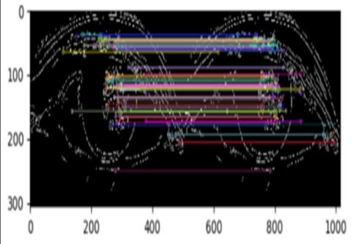
With and Without Contact lens
SAME EYE





TOTAL FGM : 31

1. COUNT DISTANCE <DMatch 0BF59FC8>
2. COUNT DISTANCE <DMatch 0C048CF8>
3. COUNT DISTANCE <DMatch 0C480FE0>
4. COUNT DISTANCE <DMatch 0BEA4E00>
5. COUNT DISTANCE <DMatch 0CEEFC98>
6. COUNT DISTANCE <DMatch 0CEEFD10>
7. COUNT DISTANCE <DMatch 0CEEFD40>
8. COUNT DISTANCE <DMatch 0CEEFD58>
9. COUNT DISTANCE <DMatch 0CEEFCC8>
10. COUNT DISTANCE <DMatch 0CEEFDB8>
11. COUNT DISTANCE <DMatch 0CEEFDE8>
12. COUNT DISTANCE <DMatch 0CEEFE18>
13. COUNT DISTANCE <DMatch 0CEEFE48>
14. COUNT DISTANCE <DMatch 0CEEFE78>
15. COUNT DISTANCE <DMatch 0CEEFEA8>
16. COUNT DISTANCE <DMatch 0CEEFED8>
17. COUNT DISTANCE <DMatch 0CEEFF08>
18. COUNT DISTANCE <DMatch 0CEEFF38>
19. COUNT DISTANCE <DMatch 0CEEFF68>
20. COUNT DISTANCE <DMatch 0CEEFF98>
21. COUNT DISTANCE <DMatch 0CEEFFC8>
22. COUNT DISTANCE <DMatch 0CF61020>
23. COUNT DISTANCE <DMatch 0CF61050>
24. COUNT DISTANCE <DMatch 0CF61080>
25. COUNT DISTANCE <DMatch 0CF610B0>
26. COUNT DISTANCE <DMatch 0CF610E0>
27. COUNT DISTANCE <DMatch 0CF61110>
28. COUNT DISTANCE <DMatch 0CF61140>
29. COUNT DISTANCE <DMatch 0CF61170>
30. COUNT DISTANCE <DMatch 0CF611A0>
31. COUNT DISTANCE <DMatch 0CF611D0>

Without Contact lens SAME EYE			
		  TOTAL FGM : 92	1. COUNT DISTANCE<DMatch 0C362968> 2. COUNT DISTANCE<DMatch 0C7886B0> 3. COUNT DISTANCE<DMatch 0D2A9ED8> 4. COUNT DISTANCE<DMatch 01DB8F08> 5. COUNT DISTANCE<DMatch 0D36D110> 6. COUNT DISTANCE<DMatch 0D36D188> 7. COUNT DISTANCE<DMatch 0D36D1B8> 8. COUNT DISTANCE<DMatch 0D36D1D0> 9. COUNT DISTANCE<DMatch 0D36D140> 10. COUNT DISTANCE<DMatch 0D36D230> 11. COUNT DISTANCE<DMatch 0D36D260> 12. COUNT DISTANCE<DMatch 0D36D290> 13. COUNT DISTANCE<DMatch 0D36D2C0> 14. COUNT DISTANCE<DMatch 0D36D2F0> 15. COUNT DISTANCE<DMatch 0D36D320> 16. COUNT DISTANCE<DMatch 0D36D350> 17. COUNT DISTANCE<DMatch 0D36D380> 18. COUNT DISTANCE<DMatch 0D36D3B0> 19. COUNT DISTANCE<DMatch 0D36D3E0> 20. COUNT DISTANCE<DMatch 0D36D410> 21. COUNT DISTANCE<DMatch 0D36D440> 22. COUNT DISTANCE<DMatch 0D36D470> 23. COUNT DISTANCE<DMatch 0D36D4A0> 24. COUNT DISTANCE<DMatch 0D36D4D0> 25. COUNT DISTANCE<DMatch 0D36D500> 26. COUNT DISTANCE<DMatch 0D36D530> 27. COUNT DISTANCE<DMatch 0D36D560> 28. COUNT DISTANCE<DMatch 0D36D590> 29. COUNT DISTANCE<DMatch 0D36D5C0> 30. COUNT DISTANCE<DMatch 0D36D5F0> 31. COUNT DISTANCE<DMatch 0D36D620> 32. COUNT DISTANCE<DMatch 0D36D650> 33. COUNT DISTANCE<DMatch 0D36D680> 34. COUNT DISTANCE<DMatch 0D36D6B0> 35. COUNT DISTANCE<DMatch 0D36D6E0> 36. COUNT DISTANCE<DMatch 0D36D710> 37. COUNT DISTANCE<DMatch 0D36D740> 38. COUNT DISTANCE<DMatch 0D36D770> 39. COUNT DISTANCE<DMatch 0D36D7A0> 40. COUNT DISTANCE<DMatch 0D36D7D0> 41. COUNT DISTANCE<DMatch 0D36D800> 42. COUNT DISTANCE<DMatch 0D36D830> 43. COUNT DISTANCE<DMatch 0D36D860> 44. COUNT DISTANCE<DMatch 0D36D890> 45. COUNT DISTANCE<DMatch 0D36D8C0> 46. COUNT DISTANCE<DMatch 0D36D8F0> 47. COUNT DISTANCE<DMatch 0D36D920> 48. COUNT DISTANCE<DMatch 0D36D950> 49. COUNT DISTANCE<DMatch 0D36D980> 50. COUNT DISTANCE<DMatch 0D36D9B0> 51. COUNT DISTANCE<DMatch 0D36D9E0> 52. COUNT DISTANCE<DMatch 0D36DA10> 53. COUNT DISTANCE<DMatch 0D36DA40> 54. COUNT DISTANCE<DMatch 0D36DA70> 55. COUNT DISTANCE<DMatch 0D36DAA0> 56. COUNT DISTANCE<DMatch 0D36DAD0> 57. COUNT DISTANCE<DMatch 0D36DB00> 58. COUNT DISTANCE<DMatch 0D36DB30> 59. COUNT DISTANCE<DMatch 0D36DB60> 60. COUNT DISTANCE<DMatch 0D36DB90> 61. COUNT DISTANCE<DMatch 0D36DBC0> 62. COUNT DISTANCE<DMatch 0D36DBF0> 63. COUNT DISTANCE<DMatch 0D36DC20> 64. COUNT DISTANCE<DMatch 0D36DC50> 65. COUNT DISTANCE<DMatch 0D36DC80> 66. COUNT DISTANCE<DMatch 0D36DCB0> 67. COUNT DISTANCE<DMatch 0D36DCE0> 68. COUNT DISTANCE<DMatch 0D36DD10> 69. COUNT DISTANCE<DMatch 0D36DD40> 70. COUNT DISTANCE<DMatch 0D36DD70> 71. COUNT DISTANCE<DMatch 0D36DDA0> 72. COUNT DISTANCE<DMatch 0D36DDD0> 73. COUNT DISTANCE<DMatch 0D36DE00> 74. COUNT DISTANCE<DMatch 0D36DE30> 75. COUNT DISTANCE<DMatch 0D36DE60> 76. COUNT DISTANCE<DMatch 0D36DE90> 77. COUNT DISTANCE<DMatch 0D36DEC0> 78. COUNT DISTANCE<DMatch 0D36DEF0> 79. COUNT DISTANCE<DMatch 0D36DF20> 80. COUNT DISTANCE<DMatch 0D36DF50> 81. COUNT DISTANCE<DMatch 0D36DF80> 82. COUNT DISTANCE<DMatch 0D36DFB0> 83. COUNT DISTANCE<DMatch 0D36DFE0> 84. COUNT DISTANCE<DMatch 0D3A5038> 85. COUNT DISTANCE<DMatch 0D3A5068> 86. COUNT DISTANCE<DMatch 0D3A5098> 87. COUNT DISTANCE<DMatch 0D3A50C8> 88. COUNT DISTANCE<DMatch 0D3A50F8> 89. COUNT DISTANCE<DMatch 0D3A5128> 90. COUNT DISTANCE<DMatch 0D3A5158> 91. COUNT DISTANCE<DMatch 0D3A5188> 92. COUNT DISTANCE<DMatch 0D3A51B8>

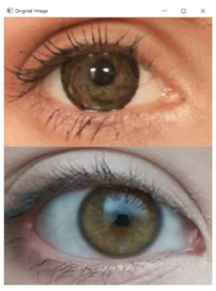
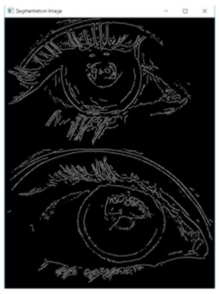
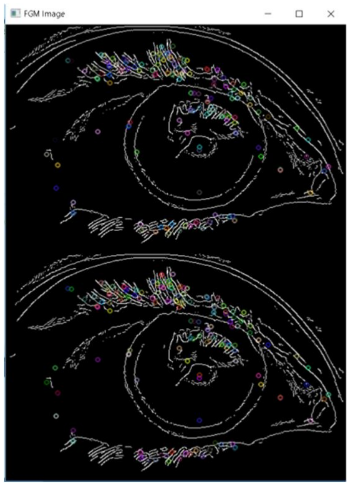
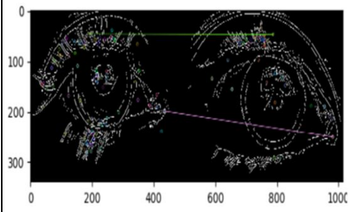
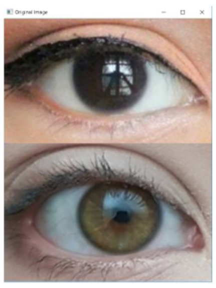
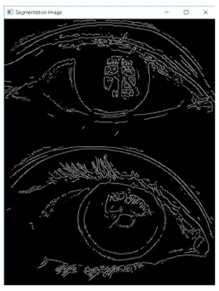
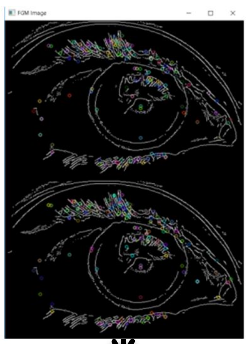
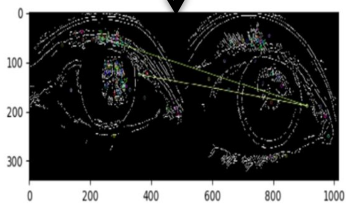
Without and with Contact lens DIFFERENT EYE			
		  TOTAL FGM : 2	1. COUNT DISTANCE <DMatch 0D9DFF80> 2. COUNT DISTANCE <DMatch 0DA172A8>
Without Contact lens DIFFERENT EYE			
		  TOTAL FGM : 2	1. COUNT DISTANCE <DMatch 0D2EB1B8> 2. COUNT DISTANCE <DMatch 0D339098>

Table 2 has shown the two original eyes from foreign and Filipino eye, and it will undergo to segmentation to get the segmented eye and from the segmented eye the FGM will perform to find the key points found. If the FGM below 5 good key points it consider unmatched key points. And using Ratio test the good key points will store and it will find the good key points from one image to another image for matching stage using FLANN based matcher.

3) *Matching Result:* The result of matching by using a new used algorithm name FLANN based matcher it's a simple matcher that shown in Table 3.

```
FLANN_INDEX_KDTREE = 0
index_params = dict(algorithm = FLANN_INDEX_KDTREE, trees=5)
search_params = dict(checks=50)

flann = cv2.FlannBasedMatcher(index_params, search_params)

matches = flann.knnMatch(des1, des2, k=2)
```

It take the descriptor of one feature in first set and is matched with all other feature is second set using some distance calculation and closet one is returned. It specifies the distance measurement to be used by creating the matcher object using the attribute DMatch.distance function to find the distance between descriptors. The lower, the better it is.

In fig 15 shown detect 41 key point that can be match, using the FlannBasedMatcher() function it can find and located the key point distance that can be match.

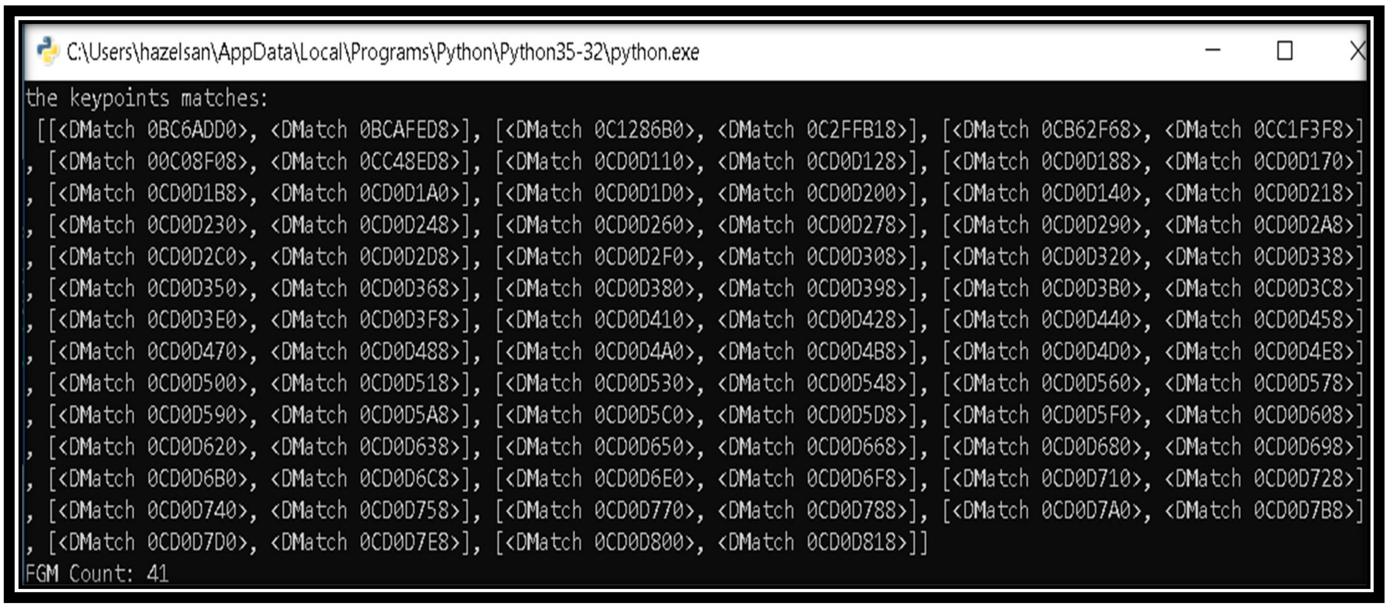

```
C:\Users\hazelsan\AppData\Local\Programs\Python\Python35-32\python.exe
the keypoints matches:
 [[<DMatch 0BC6ADD0>, <DMatch 0BCAFED8>], [<DMatch 0C1286B0>, <DMatch 0C2FFB18>], [<DMatch 0CB62F68>, <DMatch 0CC1F3F8>]
, [<DMatch 00C08F08>, <DMatch 0CC48ED8>], [<DMatch 0CD0D110>, <DMatch 0CD0D128>], [<DMatch 0CD0D188>, <DMatch 0CD0D170>]
, [<DMatch 0CD0D1B8>, <DMatch 0CD0D1A0>], [<DMatch 0CD0D1D0>, <DMatch 0CD0D200>], [<DMatch 0CD0D140>, <DMatch 0CD0D218>]
, [<DMatch 0CD0D230>, <DMatch 0CD0D248>], [<DMatch 0CD0D260>, <DMatch 0CD0D278>], [<DMatch 0CD0D290>, <DMatch 0CD0D2A8>]
, [<DMatch 0CD0D2C0>, <DMatch 0CD0D2D8>], [<DMatch 0CD0D2F0>, <DMatch 0CD0D308>], [<DMatch 0CD0D320>, <DMatch 0CD0D338>]
, [<DMatch 0CD0D350>, <DMatch 0CD0D368>], [<DMatch 0CD0D380>, <DMatch 0CD0D398>], [<DMatch 0CD0D3B0>, <DMatch 0CD0D3C8>]
, [<DMatch 0CD0D3E0>, <DMatch 0CD0D3F8>], [<DMatch 0CD0D410>, <DMatch 0CD0D428>], [<DMatch 0CD0D440>, <DMatch 0CD0D458>]
, [<DMatch 0CD0D470>, <DMatch 0CD0D488>], [<DMatch 0CD0D4A0>, <DMatch 0CD0D4B8>], [<DMatch 0CD0D4D0>, <DMatch 0CD0D4E8>]
, [<DMatch 0CD0D500>, <DMatch 0CD0D518>], [<DMatch 0CD0D530>, <DMatch 0CD0D548>], [<DMatch 0CD0D560>, <DMatch 0CD0D578>]
, [<DMatch 0CD0D590>, <DMatch 0CD0D5A8>], [<DMatch 0CD0D5C0>, <DMatch 0CD0D5D8>], [<DMatch 0CD0D5F0>, <DMatch 0CD0D608>]
, [<DMatch 0CD0D620>, <DMatch 0CD0D638>], [<DMatch 0CD0D650>, <DMatch 0CD0D668>], [<DMatch 0CD0D680>, <DMatch 0CD0D698>]
, [<DMatch 0CD0D6B0>, <DMatch 0CD0D6C8>], [<DMatch 0CD0D6E0>, <DMatch 0CD0D6F8>], [<DMatch 0CD0D710>, <DMatch 0CD0D728>]
, [<DMatch 0CD0D740>, <DMatch 0CD0D758>], [<DMatch 0CD0D770>, <DMatch 0CD0D788>], [<DMatch 0CD0D7A0>, <DMatch 0CD0D7B8>]
, [<DMatch 0CD0D7D0>, <DMatch 0CD0D7E8>], [<DMatch 0CD0D800>, <DMatch 0CD0D818>]]
FGM Count: 41
```

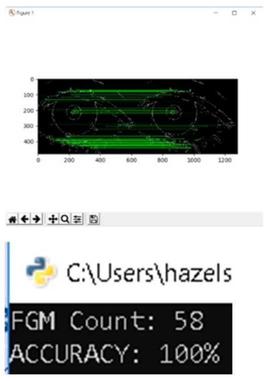
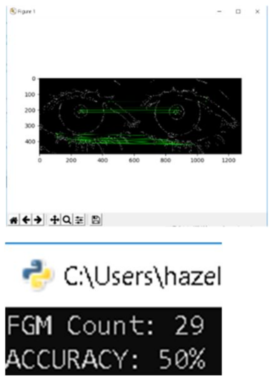
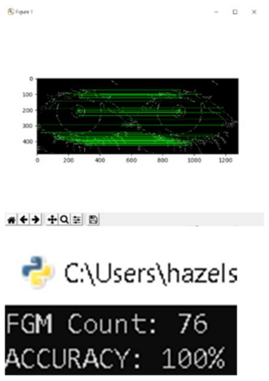
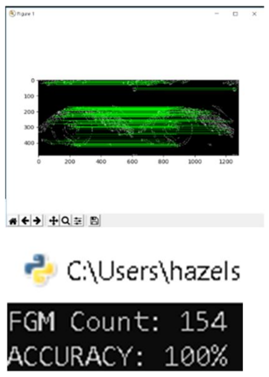
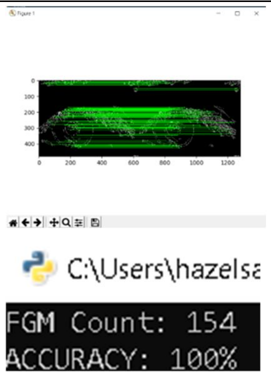
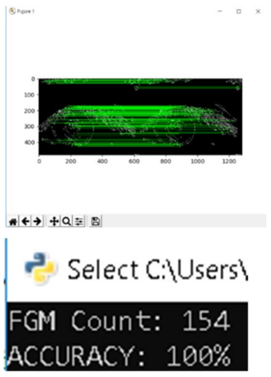
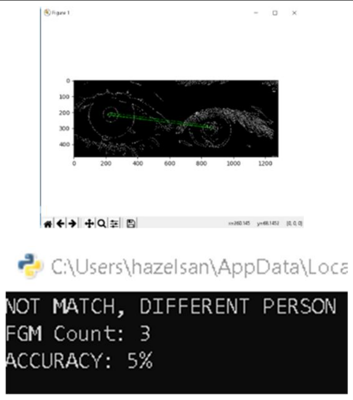
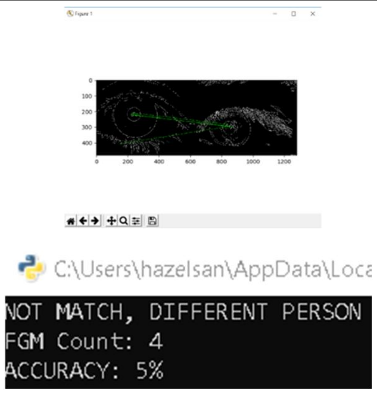
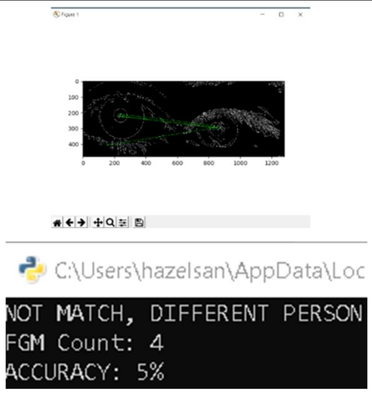
Fig 15 Key point's distance matched

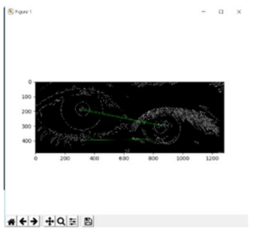
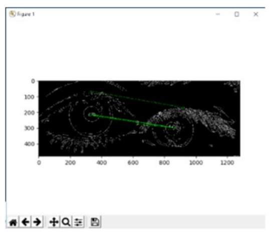
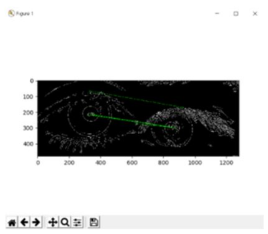
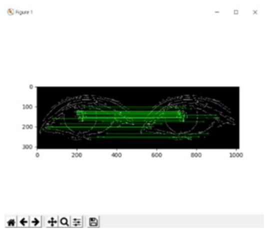
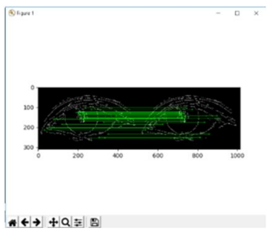
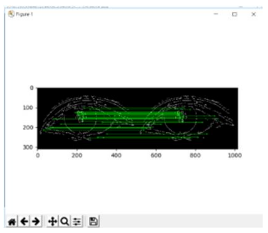
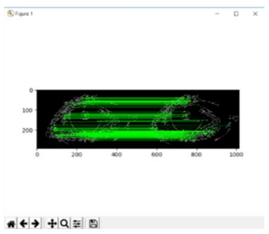
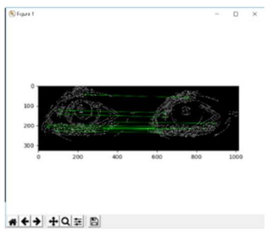
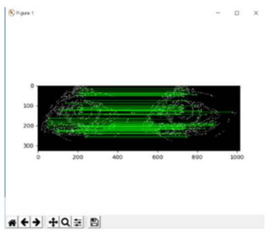
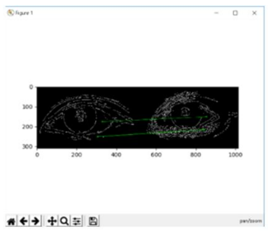
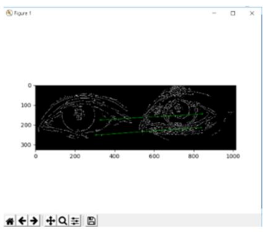
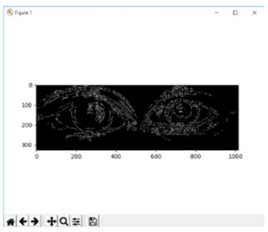
Using FLANN or Fast Library for Approximate Nearest Neighbours it contains a collection optimized for fast nearest neighbor search in large datasets and for high dimensional features. In FLANN can pass in two dictionaries which is FLANN based matcher. First dictionary is IndexParams to pass the key points that found in an image. Then second dictionary is the SearchParams it specifies the number of times the trees in the index should be recursively traversed. The higher values give better precision but its take more time to execute.

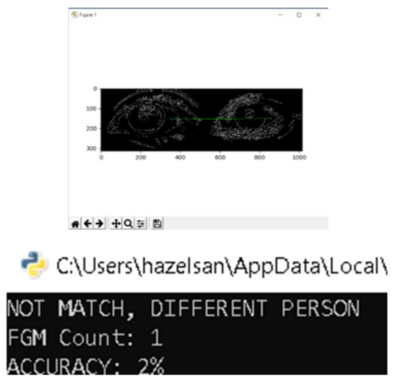
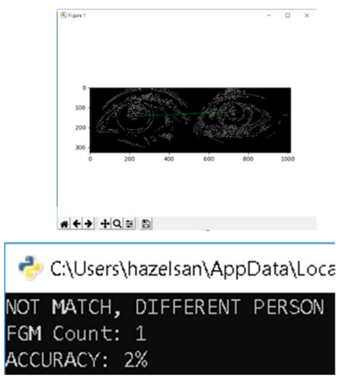
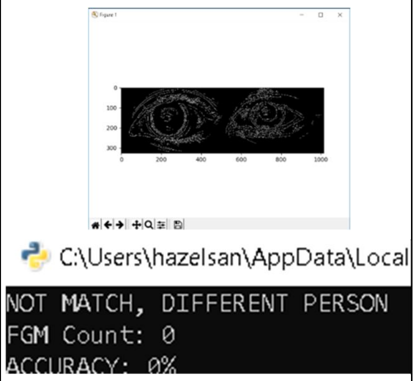
4) *Comparison of Previous study into Proposed Study:* The previous study that use MLBP algorithm that shown in Table 1. It show the result for three categories/classification N-N (Normal Eye) = **56.66%,** T-T (Colored Lens) = **66.83%**, S-S (Soft Lens) = **94.91%.**

The Table 3, shown the accuracy in matching the contact lens if the eyes are same person with or without contact lenses. The result categories into two division an Filipino Eye and Foreign Eye that divide into three N-N (Normal Eye), N-C (Normal to Contact Lens), C-C (Contact Lens to Contact Lens) of same eye and different eye to identify if the eye are same person or not.

TABLE IIIII
ACCURACY OF PROPOSED STUDY

FOREIGN EYE – SAME EYE		
NORMAL TO NORMAL (N-N)	NORMAL TO CONTACT LENS (N-C)	CONTACT LENS TO CONTACT LENS (C-C)
 C:\Users\hazels FGM Count: 58 ACCURACY: 100%	 C:\Users\hazel FGM Count: 29 ACCURACY: 50%	 C:\Users\hazels FGM Count: 76 ACCURACY: 100%
 C:\Users\hazels FGM Count: 154 ACCURACY: 100%	 C:\Users\hazelsa FGM Count: 154 ACCURACY: 100%	 Select C:\Users\ FGM Count: 154 ACCURACY: 100%
FOREIGN EYE – DIFFERENT EYE		
 C:\Users\hazelsan\AppData\Loca NOT MATCH, DIFFERENT PERSON FGM Count: 3 ACCURACY: 5%	 C:\Users\hazelsan\AppData\Loca NOT MATCH, DIFFERENT PERSON FGM Count: 4 ACCURACY: 5%	 C:\Users\hazelsan\AppData\Loc NOT MATCH, DIFFERENT PERSON FGM Count: 4 ACCURACY: 5%

 C:\Users\hazelsan\AppData\Loca NOT MATCH, DIFFERENT PERSON FGM Count: 3 ACCURACY: 4%	 C:\Users\hazelsan\AppData\Loc NOT MATCH, DIFFERENT PERSON FGM Count: 5 ACCURACY: 9%	 C:\Users\hazelsan\AppData\Loc NOT MATCH, DIFFERENT PERSON FGM Count: 5 ACCURACY: 9%
FILIPINO EYE – SAME EYE		
 C:\Users\hazels FGM Count: 31 ACCURACY: 100%	 C:\Users\hazelsa FGM Count: 31 ACCURACY: 100%	 C:\Users\hazel FGM Count: 31 ACCURACY: 100%
 C:\Users\hazelsa FGM Count: 118 ACCURACY: 100%	 C:\Users\haze FGM Count: 17 ACCURACY: 19%	 C:\Users\hazels FGM Count: 89 ACCURACY: 100%
FILIPINO EYE – DIFFERENT EYE		
 C:\Users\hazelsan\AppData\Loca NOT MATCH, DIFFERENT PERSON FGM Count: 3 ACCURACY: 9%	 C:\Users\hazelsan\AppData\Loc NOT MATCH, DIFFERENT PERSON FGM Count: 2 ACCURACY: 6%	 C:\Users\hazelsan\AppData\Loca NOT MATCH, DIFFERENT PERSON FGM Count: 0 ACCURACY: 0%

 C:\Users\hazelsan\AppData\Local\ NOT MATCH, DIFFERENT PERSON FGM Count: 1 ACCURACY: 2%	 C:\Users\hazelsan\AppData\Loca NOT MATCH, DIFFERENT PERSON FGM Count: 1 ACCURACY: 2%	 C:\Users\hazelsan\AppData\Local NOT MATCH, DIFFERENT PERSON FGM Count: 0 ACCURACY: 0%

The summary Table of accuracy for Contact Lens Detection for security using FGM with FLANN based matcher show how effective to match the one image to another image by using key points. And it easy identify if the two eyes are same person or not.

IV. CONCLUSION AND RECOMMENDATION

The new used FGM with FLANN based matcher algorithm proved the applicability and accuracy to match one image to another image even if has a contact lens or without contact lens. The proposed study was being tested and experiment in AMD Ryzen 5 2500U with 2GHz processor, 4GB Mobile RAM, dual graphic card and GPU rendering of RX 560 x with 5GB.Based on the result obtained from the experiments the proposed new used algorithm shows potential for contact lens detection for security challenges as the previous study presented. It is recommended that more work on how to improve the matching in normal to contact lens detection.

V. ACKNOWLEDGMENT

H.S Loverez-Patilano would like to acknowledge the financial support provided by the Commission on Higher Education, K-12 Project Management Unit, and Philippines.

REFERENCES

[1] Zingaretti, P., Tascini, G., Puliti, P., & Zanoli, S. (1996). Imaging approach to real-time tracking of submarine pipeline. https://doi.org/10.1117/12.234645
[2] All About Vision. (2017). Silicone Hydrogel Contact Lenses - A Complete Guide. Retrieved from http://www.allaboutvision.com/contacts/silicone-hydrogel.htm
[3] Komulainen, J., Hadid, A., & Pietik, M. (2014). Generalized textured contact lens detection by extracting BSIF description from Cartesian iris images. https://doi.org/10.1109/BTAS.2014.6996237
[4] Bowyer, K. W., & Doyle, J. S. (2014). Cosmetic contact lenses and iris recognition spoofing. Computer, 47(5), 96–98. https://doi.org/10.1109/MC.2014.118
[5] Rani, K. S. S., & Yuvaraju, M. (2016). MLTP Based Contact Lens Detection in Iris Recognition for Anti-spoofing,
[6] Sooden, S., & Singh, H. (2017). Artificial Neural Network using Zernike Moments with Speeded Up Robust Features based Classification For Iris Recognition, 7(4), 51–61.
[7] Erdogan, G., & Ross, A. (2013). Automatic Detection of Non-Cosmetic Soft Contact Lenses in Ocular Images. SPIE Biometric and Surveillance Technology for Human and Activity Identification X. https://doi.org/10.1117/12.2018096
[8] Kumar, B., Nigam, A., & Gupta, P. (2016). Fully automated soft contact lens detection from NIR iris images. ICPRAM 2016 - Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods.
[9] Kumar, B., Nigam, A., & Gupta, P. (2015). Automated Soft Contact Lens Detection Using Gradient based Information, 1505.
[10] Yadav, D., Kohli, N., Doyle, J. S., Singh, R., Vatsa, M., & Bowyer, K. W. (2014). Unraveling the effect of textured contact lenses on iris recognition. IEEE Transactions on Information Forensics and Security, 9(5), 851–862. https://doi.org/10.1109/TIFS.2014.2313025
[11] Kishan, G. ; V. N. (2015). An Enhanced Approach for Identification of Cosmetic Contact Lenses on Toc H Institute of Science and Technology, 3(03), 2596–2600
[12] Bai, X., Ning, X., & Wang, L. (2012). Analysis and Comparison of Feature Detection and Matching Algorithms for Rovers Vision Navigation, (60874095).
[13] Wei, Z., Qiu, X., Sun, Z., & Tan, T. (2008). Counterfeit Iris Detection Based on Texture Analysis, 1–4. https://doi.org/10.1109/ICPR.2008.4761673
[14] Guo, Z., Zhang, L., & Zhang, D. (2010). Rotation invariant texture classification using LBP variance (LBPV) with global matching. Pattern Recognition, 43(3), 706–719. https://doi.org/10.1016/j.patcog.2009.08.017
[15] Kohli, N., Yadav, D., Vatsa, M., Singh, R., & Noore, A. (2013). Detecting Medley of Iris Spoofing Attacks using DESIST Naman Kohli.

[16] Doyle, J. S., Flynn, P. J., & Bowyer, K. W. (2013). Automated classification of contact lens type in iris images. Proceedings - 2013 International Conference on Biometrics, ICB 2013. https://doi.org/10.1109/ICB.2013.6612954

[17] Kishan, G. ; V. N. (2015). An Enhanced Approach for Identification of Cosmetic Contact Lenses on Toc H Institute of Science and Technology, 3(03), 2596–2600.

[18] Ashoka, H. N., Manjaiah, D. H., & Rabindranath, B. (2012). Feature Extraction Technique for Neural Network Based Pattern Recognition. International Journal on Computer Science and Engineering, 4(03), 331–340.

[19] Hu, Y., Sirlantzis, K., & Howells, G. (2016). A study on iris textural correlation using steering kernels. 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems, BTAS 2016, 1. https://doi.org/10.1109/BTAS.2016.7791160

[20] Azzopardi, G., & Petkov, N. (2015). Computer analysis of images and patterns: 16th international conference, CAIP 2015 Valletta, Malta, september 2???4, 2015 proceedings, part I. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 9256, pp. 702–714). https://doi.org/10.1007/978-3-319-23192-1

[21] Nigam, A., & Gupta, P. (2014.). Designing an Accurate Hand Biometric Based Authentication System Fusing Finger Knuckleprint and Palmprint. https://doi.org/10.1016/j.neucom.2014.03.083

[22] Daugman, J. (2004). How iris recognition works. Circuits and Systems for Video Technology, IEEE Transactions On, 14(1), 21–30.

[23] Yadav, D., Kohli, N., Doyle, J. S., Singh, R., Vatsa, M., & Bowyer, K. W. (2014). Unraveling the effect of textured contact lenses on iris recognition. IEEE Transactions on Information Forensics and Security, 9(5), 851–862. https://doi.org/10.1109/TIFS.2014.2313025

[24] Kywe, W. W., Yoshida, M., & Murakami, K. (2006). Contact lens extraction by using thermo-vision. Proceedings - International Conference on Pattern Recognition, 4, 570–573. https://doi.org/10.1109/ICPR.2006.406

[25] Marinelli, M., Mancini, A., & Zingaretti, P. (2014). GPU acceleration of feature extraction and matching algorithms. Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference On, 1–6. https://doi.org/10.1109/MESA.2014.6935620

[26] Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features (SIFT). International Conference on Computer Vision, 2, 1150–1157 vol.2. https://doi.org/10.1109/ICCV.1999.790410

[27] Lowe, D. G. (2004). Distinctive image features from scale invariant keypoints. International Journal of Computer Vision, 60, 91–11020042. https://doi.org/http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[28] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3951 LNCS, 404–417. https://doi.org/10.1007/11744023_32

[29] Chouhan B., Shukla S. (2010). Analysis of statistical feature extraction for Iris Recognition System using Laplacian of Gaussian filte. Internnational Journal of Applied Engineering Research, Dindigul, Volume 1, No 3,2010

[30] Ermatita, Budianta D. (2017). Brute Force Algorithm Implementation on Knowledge Management System Overcoming Heavy Metal of PB and CD in Soil and Palm Oil Plantation. International Journal of Latest Trends In Engineering and Technology. Vol.(8), pp 297-301. http://dx.doi.org/10.21172/1.82.039

[31] D.G. Lowe (2004). Distinctive image features from scale-invariant keypoints http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[32] M. Muja, D. Lowe (2009). Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.