Face Anti-spoofing using Pupil Response

Aishwarya Mala GM
EEMCS, University of Twente,
S2242842
a.m.gurusamymuthuvelrabindran@student.utwente.nl

Christopher Benjamin Leonard
EEMCS, University of Twente,
S2378515
c.benjaminleonard@student.utwente.nl

Dharanish NH
EEMCS, University of Twente,
S2372525
d.nambinayakanahallihanumantha@student.utwente.nl

***Abstract* – An anti-spoofing algorithm for face detection systems has been proposed through this work. The algorithm aims at using the sensitivity of the human eye's pupil to light to detect if the face recognition system is being deceived by a display of a human face or not. The pupil of a live face is expected to shrink in size when light is flashed upon it. Thus in this work, a method to detect the area of the pupil has been developed. The colour information of the input image is used to derive an EyeMap that helps separate the eye region from the rest of the face. Further, a radial symmetry transform is used to locate the pupil in the eye region. As the final step, the region that corresponds to the pupil is separated from the eye region and its area is calculated. The pupil areas are calculated on images that are acquired in two different light settings. These areas are analyzed to check if the changes in the pupil area are in line with the changes in light settings. Based on this analysis a spoofed or not-spoofed decision is made. Using this approach we were able to achieve accurate performance for spoof detection on the data that we collected.**

Keywords*—*Face Recognition anti-spoofing, Fast Radial Symmetry Transform, Pupil Detection.

I. Introduction

In the era of digitalization of technology, biometric security is one such technology which has seen significant improvement over time. This technology is used everywhere from unlocking a mobile device to unlocking a safe vault in the bank. To do this, we know that biometric features are unique for every individual and using them for authentication would be more secure than using passcode based or key-based security.

Face recognition system is an application of face detection where it can verify a human face from an image or video frame with a database of the faces. The first automated facial recognition was achieved by Woody Bledsoe, Helen Chan Wolf, and Charles Bisson in 1960 [1]. Their early version needed both man and machine interference to recognize the face. Now in 2020, we use autonomous technology like for example Face ID developed by Apple Inc. The system can recognize faces with glasses, clothing, makeup, and facial hair, and adapts to changes in appearance over time [2].

With the advancement in the recognition technology there were also some notable drawbacks. Facial anti-spoofing is the task of preventing false facial verification and the examples of attacks are the following: Print attack - the attacker uses someone's photo. Replay/video attack - a more sophisticated way to trick the system, which usually requires a looped video of a victim's face. 3D mask attack - a mask is used for spoofing. It's a more sophisticated attack than playing a face video. In addition to natural facial movements, it enables ways to deceive some extra layers of protection such as depth sensors.

To overcome these issues, a liveness detector is used. Liveness detection is a process that reads the subject's physiological signs of life. In our work, we have proposed a liveness detection algorithm based on pupil response to light stimuli. We know that the pupil changes its shape based on the surrounding light. The aim of our algorithm is to extract the pupil area from the input images that are captured under different light settings and then use these pupil areas to detect the liveness of the subject and to anti-spoof face recognition.

II. Related Works

There are a number of liveness detection algorithms that are based on pupils in the literature.

Shen et al., have developed a liveness detection algorithm based on iris tracking for the prevention of face spoofing attacks [3]. Users are required to move their eyes along with a randomly generated poly-line, and trajectories of irises are then used as evidence for liveness detection.

Killioğlu et al., in their work, have introduced a face recognition anti-spoofing technique that uses pupil tracking [4]. Haar-Cascade Classifier is trained to identify the eyes in the input image. From the cropped out eye regions the pupils are extracted. A square frame with 8 LEDs on each side is designed. A random LED is lighted up and the eye pupil direction and the direction of the lighted LED is compared to estimate the liveness of the subject.

Huang et al., have come up with a liveness detection algorithm based on pupil constriction [5]. Two eye images are acquired in two different illumination conditions. A feature descriptor that contains the similarity measure between the iris patches and the ratio of iris to pupil diameter is created. This is used to train a Support Vector Machine to detect liveness.

Toth, in his work has listed the common liveness detections methods [6]. Under the algorithms that are based on eyes, he has mentioned the red-eye effect. Light entering the eye is reflected back to the light source by the retina. Functional eye cavity optics make the eye to appear red when this effect is captured by the camera. Thus this could be used as a liveness detection method. Another anti-spoofing method discussed is the detection of the Purkinje reflections in the pupil.

As for pupil detection, Santini, T. et al., discussed several approaches to detect pupil under challenging conditions for eye tracking [7]. A Gradient-based approach, the dot product of the gradient vector and the normalized displacement vector with the same orientation is calculated. An optimal pupil centre is determined as the position which has the maximum response. A voting-based approach, the circular Hough transform is used to determine the position of the pupil. A fitting based approach, RANdom SAmple Consensus

(RANSAC) algorithm is used to detect the elliptical shape of the pupil. In the divided condition approach, Canny edge detection is used to detect the longest curved edge in the darkest area. Using least square method the calculated points are fit into the ellipse.

Skodras E et al., in their work, have introduced an algorithm that uses colour information to detect the eye centres in a given low-resolution face image [8]. The chrominance values extracted from the YCbCr colour space is used to distinguish the eyes from the face. The authors propose to use radial symmetry transform to locate the centre of the eye. The authors claim that this algorithm is really effective in identifying the position of the eye centres in low-resolution images.

These are the major pupil based liveness detection algorithms and pupil detection algorithms that exist in the literature. Based on these, an anti-spoofing algorithm that uses the pupil response to detect liveness has been designed.

III. METHOD

A. *Dataset Collection*

The image dataset for the project consists of 65 images of the face of 6 people. All the images contain only the face of the people and with a clear background. From everyone approximately 10 or 11 images were capture. Light was flashed all the time during image acquisition to make the pupil visible. The variation in the pupil size was induced by moving the light source towards the eye. Initially a set of images were captured with the light source positioned in such a way that the pupil is visible in the images acquired. Another set of images were captured by moving the light source towards the eye of the individual to induce a change in the pupil size. All the images were captured with the face positioned approximately 15cm to 20cm away from the camera. A close distance was maintained between camera and face to make sure that we can see the complete face and recognize the eye and the pupil properly. Every image has a slight variation in the head pose to improve the dataset. The images were captured on a mobile phone with a 48mp camera and each image is of 4000x1824 resolution.

B. *Anti-spoofing algorithm*

An algorithm to anti-spoof face recognition systems using the response of the pupil to light flash has been discussed. The pupil size is expected to decrease with increase in light in the surrounding thus the proposed methodology will majorly focus on an algorithm to estimate the area of the pupil. The pupil area is calculated on two images that are captured with different light settings. The areas of the pupils are compared and if a significant difference is observed then the input is declared to be genuine.

The anti-spoofing methodology is divided into the following five main steps: Pre-processing, Eye detection, Pupil centre detection, Pupil area calculation and Spoof detection. Each step is explained in detail in each of the sub-sections below.

1) *Pre-processing*

There should not be too much face movement during the image acquisition process. The subject moving their face towards or away from the camera will induce unnecessary changes in the area of the pupil. This must thus be kept under check. For this purpose, the SURF keypoints and descriptors calculated on the input images were used. The most strongly matched keypoints were chosen and the mean of the differences in the position of each of these matched keypoints was calculated. Given that the camera position was fixed and was not changed during image acquisition, this calculated mean will be proportional to the change in position of the face. A threshold was set and if this mean distance is greater than the threshold then there is too much unnecessary movement during the input image acquisition and the inputs are rejected. Moving the photograph or any display of the human face towards and away from the camera will also change the size of the pupil even though it is an attempt to spoof the system. This pre-processing step might come in use to reject these types of attempts to fool the algorithm too. Fig.1 shows the SURF points detected in the input images and the matched points that were used for the estimation of the mean shift in face positions.

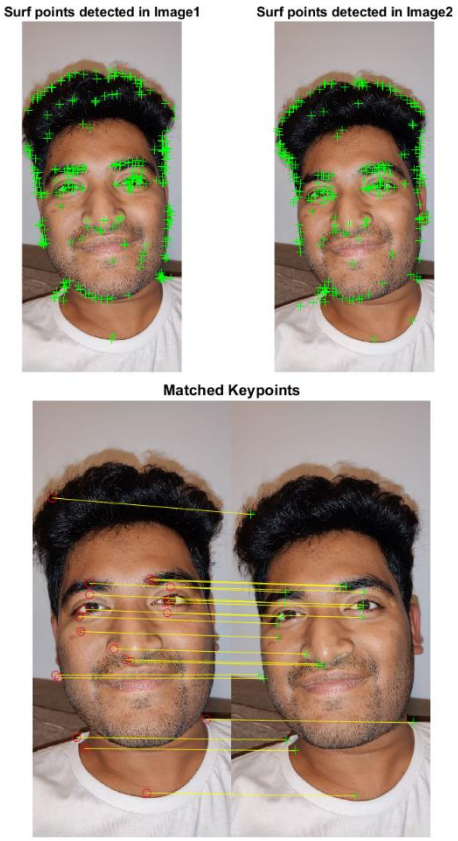

Fig. 1. The SURF keypoints detected in the input images and the matched keypoints that were used for estimating the shift in the face positions.

The second pre-processing step is to extract the face region from the two input images. An object detector based on the Viola-Jones algorithm to detect people's faces was used. The background in the input images are discarded and only the face regions extracted from the input images are passed on for further processing. Fig. 2 shows the input images that were captured in two different light settings and the face regions that were extracted from them. Fig.3 is a flowchart that outlines the pre-processing steps.

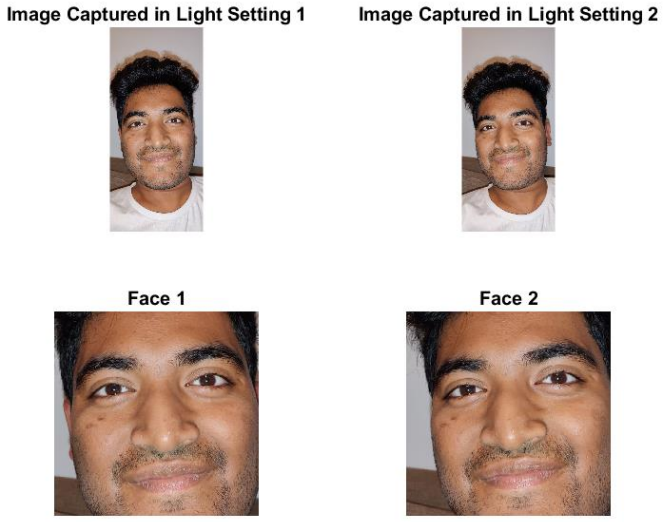

Fig. 2. Input images and faces detected

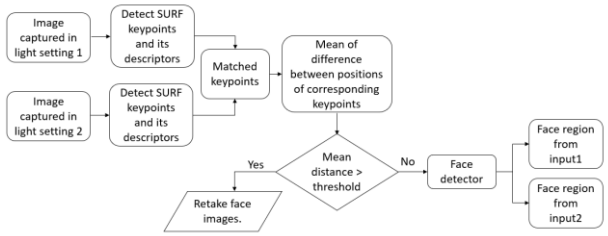

Fig. 3. Pre-processing steps

2) Eye Detection

Our eye and pupil centre detection method was inspired from the work of Skodras E et al [8] where the position of the eye is determined using the colour information. Modifications were added to the initial algorithm according to our application needs. Also, hereafter the results are shown for only one of the face images. It is to be understood that the same set of steps are simultaneously applied on the other face image too.

As the first step, the face image is split in half and the bottom half is discarded as it is a known fact that the eyes are present on the top half of the face. On this top half, finer adjustments based on face geometry are made to get rid of the forehead part and the sides. This will help refine the area of search. The next step is to convert the image from RGB to YCbCr colour space. The human skin irrespective of the skin type and colour shares similar chrominance values and thus the Cb and Cr information from the YCbCr colour space can be used to create a map where the skin regions are suppressed and the other regions in the face are highlighted. This 'EyeMapC' is computed as per the formula given in equation (1). Before applying the formula, Cb and Cr are normalized and lie in the range [0,1]. $\overline{Cr}$ in the equation is the compliment of Cr calculated as per equation (2) using the normalized Cr values.

$$EyeMapC = \frac{1}{3}\left[(Cb)^2 + (\overline{Cr})^2 + (Cb/Cr)\right] \quad (1)$$

$$\overline{Cr} = 1 - Cr \quad (2)$$

Fig.4 shows the ROI for eye search cropped out of the face image and its corresponding EyeMapC calculated using the normalized Cb and Cr values as per equations (1) and (2).

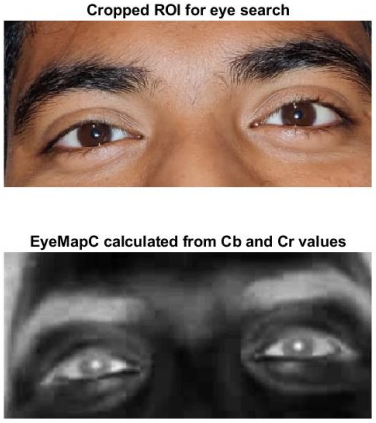

Fig. 4. The search region for eye search cropped out of the face image and its corresponding EyeMapC computed from the Chrominance (Cb and Cr) values.

It could be seen from Fig.4 that the non-skin regions such as the eyebrows and eyes have high intensity values in the computed EyeMapC. To identify the eye region a window was defined and it was moved along the EyeMapC and scores were computed. The window size was initially declared based on the face geometry but later it was tuned according to the performance on our dataset. An overlap was maintained by using a shift of one-fifth the window size for calculating the scores. The score is nothing but the sum of the EyeMapC values calculated for the pixels within the window. These scores are recorded in a score matrix. The position of the eyes is determined by sweeping through this score matrix and retrieving the regions with the highest scores. Fig.5 shows the eyes detected.

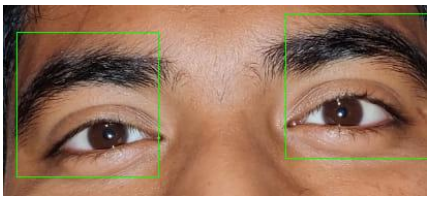

Fig. 5. Eyes detected

3) Pupil Centre Detection

The eyes detected are cropped out of the image and the remaining steps are performed on these eye images. In this section, the steps that were used to obtain the pupil centre from the eye images are explained.

The first step was to create an EyeMapI as defined by equation (3) from Y (Luminance) and the previously computed EyeMapC values. The purpose of EyeMapI is to further accentuate the pupil region for centre detection by using the Luminance value. The pupil is a dark region and thus must have low Luminance value. Thus dividing the previously calculated EyeMapC by the corresponding Luminance value will highlight the pupil area in the eye and this is shown in Fig.6. B1 and B2 are circular, flat structuring elements of radii R_{B1}=[width of the eye image/40] and R_{B2}= $R_{B1}/2$. Applying dilation operation on EyeMapC and erosion operation on the Y (Luminance) component of the eye region will help reduce the effects of flash or other unwanted

reflections in the pupil region. The radii of the structuring elements are chosen based on an approximate pupil radius estimate obtained from the size of the eye region. δ term is added in the denominator to avoid very large values in EyeMapI in case of very low luminance values. δ could be any arbitrary number but a data-based value would be $\delta = mean(Y \ominus B2)$.

$$EyeMapI = \frac{EyeMapC \oplus B1}{(Y \ominus B2) + \delta} \quad (3)$$

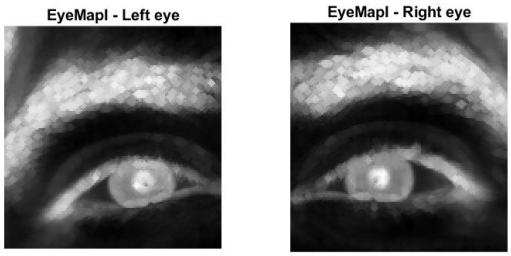

Fig. 6. EyeMapI of the eyes

For the next step, we introduce the Fast Radial Symmetry Transform (FRST). FRST uses the local radial symmetry to identify the key points of interest in a given image [9,10]. The inputs needed for FRST calculation are the image, the set of radii (*N*), mode and α the radial strictness parameter. The FRST represented by *S* is calculated based on the pseudocode in Algorithm 1.

Algorithm 1 Algorithm for calculating FRST

Input: Image; mode; set of Radii; α
Output: FRST of input image

Steps:
Calculate the gradient of the input image **grad**.
FOR each radius **n** in the set of Radii:
FOR each pixel **P** in the input image:

a. Calculate positively affected pixel P_{af} of P.
If mode = 'bright':
$P_{af} = P + round(\frac{grad(P)}{||grad(P)||} . n)$
Else if mode = 'dark':
$P_{af} = P - round(\frac{grad(P)}{||grad(P)||} . n)$, for 'dark'

b. Update $O_n(P_{af}) = O_n(P_{af}) + 1$
c. Update $M_n(P_{af}) = M_n(P_{af}) + || grad(p) ||$

END
$O_n = abs(O_n)$
$O_n = O_n / max(O_n)$
$M_n = abs(M_n)$
$M_n = M_n / max(M_n)$
$S_n = (M_n . O_n^{\alpha}) * G_n$
END
$S = \frac{1}{|N|} \sum_{n \epsilon N} S_n$

Return S

The FRST of EyeMapI is calculated using the 'bright' mode and with *N* in the range [width of the eye image/40] to [2*width of the eye image/40] with increments of 0.5. In this mode, the affected pixels are calculated along the direction of the gradient and thus it is expected for the centre of a bright (almost) circular area in the input image EyeMapI, the pupil centre, to be the affected pixel for a lot of pixels in the image. This will result in large values at the position of this centre pixel in the O_n and M_n matrices for radii values *n* close to the actual radius of the circular region. The choice of *N* is again based on the approximate pupil radius estimate calculated based on the eye image size. The increment is set at 0.5 and it was found that value was able to give sufficiently accurate results. Fig.7 shows the EyeMapI of one of the eyes and the corresponding FRST calculated.

The FRST of the Luminance (*Y*) component of the eye image is calculated using the 'dark' mode and the same *N* that was used for EyeMapI. Before computing the radial transform, morphological erode operation is done on *Y* using the same structural element B1 defined previously to try and reduce the effects of unwanted reflections in the pupil area. In the 'dark' mode, the affected pixels are calculated in the direction opposite to the gradient thus it is expected for the centre of a dark (almost) circular area in the input image, the pupil centre, to be the affected pixel for a lot of pixels in the input image and in turn have large O_n and M_n values for the appropriate *n* values. Fig.8 shows the *Y* component of one of the eye image, *Y* component after eroding operation was applied and the FRST computed on *Y* after eroding operation.

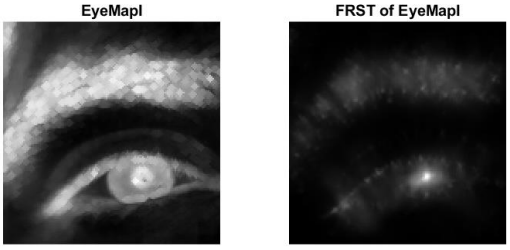

Fig. 7. EyeMapI and the Fast Radial Symmetry Transform calculated on it

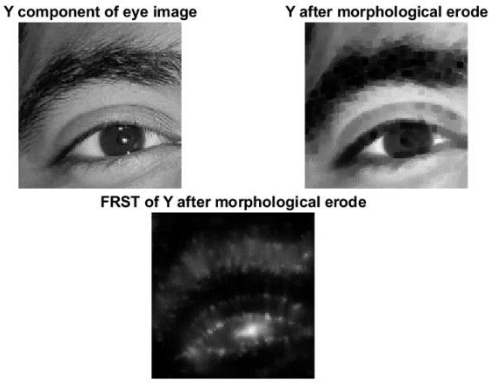

Fig. 8. Luminance (Y) component of the eye image before and after the eroding operation and the Fast Radial Symmetry Transform calculated on Y after the morphological eroding operation.

The FRST calculated on the EyeMapI $S_{EyeMapI}$ and the Luminance component $S_{Luminance}$ are combined and the position of the maximum is declared as the pupil centre. Fig.9 shows the pupil centres calculated on the eye images.

$$(x_{cen}, y_{cen}) = position\ of\ \left(max\left(S_{EyeMapI} + S_{Luminance}\right)\right) \quad (4)$$

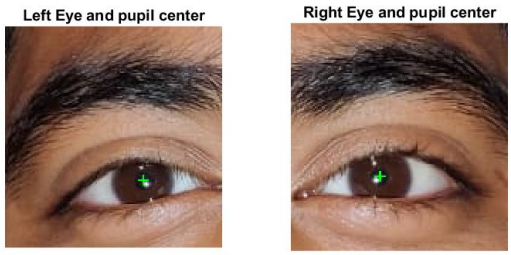

Fig. 9. Pupil centres obtained on the eye images

4) Pupil Area Calculation

In this section, we explain how the pupil area was calculated using the pupil centre from EyeMapC and Luminance (Y) component.

A square window was defined and the region around the detected pupil centre was cropped out from EyeMapC and Y. The window was defined in such a way that it extended to a length of $[\max(size(eye\ image))/10]$ from the centre to the left, right, top and bottom. This window size was found to define an ideal ROI for pupil search. A simple thresholding operation was done based on the distribution of the pixel intensities in the ROI region to find the pupil area. In EyeMapC, the pixels with intensities in the top 10 percentile were thresholded to 1 and the remaining were thresholded to 0 to create a binary image. In Luminance (Y), the pixels with intensities below the 10^{th} percentile were thresholded to 1 and the remaining were thresholded to 0 to create another binary image. Both these binary images are combined using the OR operation. In the case where this final binary image contains multiple white areas, the white area that contains the centre pixel is selected and the others are discarded. This will be the approximate pupil area. If none of the white areas contains the centre then it means that the pupil has not been found and the same is displayed as an error message. Finally, an ellipse is fitted to the area found. The pupil area is given by the area of this ellipse. The process explained above is depicted in Fig.10.

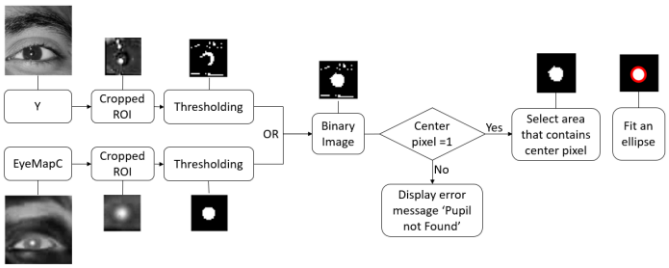

Fig. 10. Pupil area detection process

5) Spoof Detection

There are two pupil area values obtained from each of the two input images. The areas obtained from the left eye in each of the input images and the areas obtained from the right eye are compared. The algorithm checks if the set of pupil areas obtained from the input image 1 (light source away from eyes) is significantly greater than the corresponding pupil areas obtained from image 2 (light source closer to the eye). To make sure the difference in the pupil areas is significant, a threshold value is defined. If the difference in the pupil areas is greater than the defined threshold then the algorithm declares that there is 'no spoof' detected. In Fig.11 we have shown the pupil areas (rounded to the nearest integer) that were detected in each of the input images.

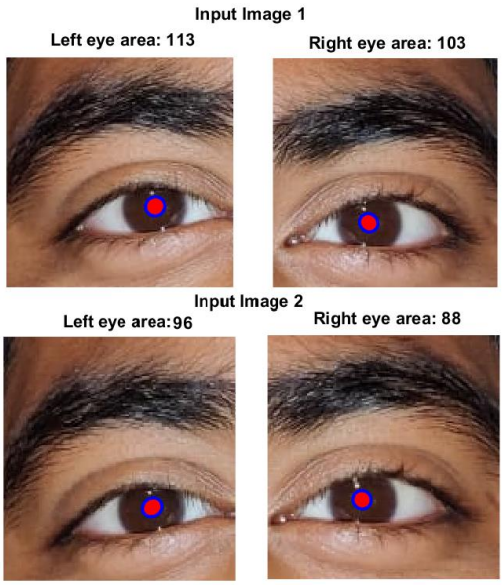

Fig. 11. The pupil areas detected in the input images (red) and the ellipse that was fitted to the area (blue)

IV. Results & Discussion

Manually calculating the area and error in all the 65 images in the dataset is very hectic. Thus we randomly chose six images each belonging to one of the six participants that we had. Table 1 shows the results that were obtained from these six images. The true positive (TP) column represents the number of pupil pixels that were correctly identified by the algorithm. The false positives (FP) column represents the number of pixels that were actually not part of the pupil but were included in the pupil area by the algorithm. The false negatives (FN) column contains the number of pixels that were actually part of the pupil but were neglected and not included under the pupil area by the algorithm. True negatives (TN), the number of pixels that were correctly classified as non-pupil pixels is too large and thus it has not been taken into consideration for performance analysis. Fig.12 shows the magnified version of the detected pupil area detected in eye 1 of image number 6. This view is to show the performance of the algorithm. The performance results computed on the image shown in Fig.12 can be found in Table 1.

TABLE 1. Results obtained on the chosen test images

Image Number	Pupil Area 1			Pupil Area 2		
	TP	FP	FN	TP	FP	FN
1	71	3	5	62	4	3
2	75	2	5	79	10	0
3	92	4	0	96	0	3
4	83	6	0	83	2	2
5	85	4	3	81	1	3
6	96	2	4	88	3	1
				Precision: 0.96		
				Recall: 0.97		

From the spoofing perspective, the algorithm showed very good performance on our dataset. By suitably pairing up images from the dataset, we created 27 input image pairs. The algorithm was able to detect the change in pupil size in all the input image pairs. We were unable to test how the algorithm will react when tried to spoof. We tried using a photograph instead of a live face and the algorithm failed to detect the pupils. This was due to the quality of the photograph and camera that was available to us. Pupils were not visible in these images and thus the algorithm was unable to process these kinds of inputs.

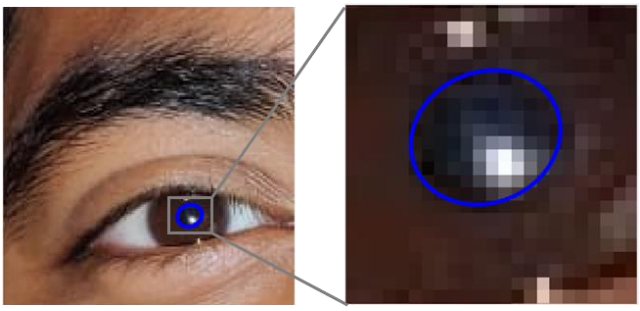

Fig. 12. Detected pupil area – Magnified view

V. Future Work

An algorithm was developed to obtain the area of the pupils when an image of a human face is given as input. This pupil area calculated was further used for anti-spoofing face recognition systems. Scope for improvement would be to include more sophisticated cameras and equipment that can help capture better images with the pupil more clearly visible. This would further increase the robustness of the proposed algorithm. Another scope of improvement from the anti-spoofing algorithm perspective would be to extend this method to video inputs. The series of pupil area measurements could be processed into a feature vector to train Machine Learning models to detect spoofing attempts.

References

[1] en.wikipedia.org. 2020. Facial Recognition System.[online] Available at: <https://en.wikipedia.org/wiki/Facial_recognition_system> .

[2] en.wikipedia.org. 2020. Face ID. [online] Available at: <https://en.wikipedia.org/wiki/Face_ID>.

[3] Shen, M., Liao, Z., Zhu, L., Mijumbi, R., Du, X. and Hu, J., 2018. IriTrack: Liveness Detection Using Irises Tracking for Preventing Face Spoofing Attacks. arXiv preprint arXiv:1810.03323.

[4] Killioğlu, M., Taşkiran, M. and Kahraman, N., 2017, January. Anti-spoofing in face recognition with liveness detection using pupil tracking. In 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 000087-000092). IEEE.

[5] Huang, X., Ti, C., Hou, Q.Z., Tokuta, A. and Yang, R., 2013, January. An experimental study of pupil constriction for liveness detection. In 2013 IEEE Workshop on Applications of Computer Vision (WACV) (pp. 252-258). IEEE.

[6] Toth B. (2009) Liveness Detection: Iris. In: Li S.Z., Jain A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-73003-5_179

[7] Santini, T., Fuhl, W., & Kasneci, E. (2018). PuRe: Robust pupil detection for real-time pervasive eye tracking. Computer Vision and Image Understanding, 170, 40-50.

[8] Skodras, E. and Fakotakis, N., 2015. Precise localization of eye centers in low resolution color images. Image and Vision Computing, 36, pp.51-60.

[9] Loy, G. and Zelinsky, A., 2002, May. A fast radial symmetry transform for detecting points of interest. In European Conference on Computer Vision (pp. 358-368). Springer, Berlin, Heidelberg.

[10] Sandro (2020). Fast Radial Symmetry Transform (https://www.mathworks.com/matlabcentral/fileexchange/45961-fast-radial-symmetry-transform), MATLAB Central File Exchange. Retrieved September 28, 2020.