Iris Recognition using Machine Learning from Smartphone Captured Images in Visible Light

Md. Fahim Faysal Khan,[1*] Ahnaf Akif,[1] and M. A. Haque[1]

[1]Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology
Dhaka -1205, Bangladesh
[*]fahim.faysal11@gmail.com

Abstract—**This work shows the applicability and feasibility of different machine learning techniques on iris recognition from smartphone captured eye images. First, the iris is localized using the popular Daugman's method and the eyelids are suppressed with canny edge detection technique. Then normalization of the extracted iris region is performed in a novel way by setting an adaptive threshold. Next, the normalized image is decomposed using Haar wavelets to obtain the feature vectors. Histogram equalization is performed for better classification accuracy. After that, different classifiers are trained using the extracted feature vectors which yield about 99.7% accuracy for training and 97% accuracy for testing. Finally, the results are compared with other previously applied methods on the same dataset and it is found that the proposed method outperforms most of them.**

Index Terms—**Iris recognition, machine learning, visible spectrum, eyelash removal, smartphone**

I. Introduction

Human iris is well known for its uniqueness, stability and non-evasiveness [1]. Hence iris recognition is a very popular problem for the researchers in the field of bioinformatics, cryptography, computational intelligence etc. Many successful approaches have been taken so far. This approaches can be classified into two categories based whether they used machine learning or not. A significant thing to notice that the datasets which are used in these approaches are iris images captured by NIR (Near Infrared) camera as they offer very good visibility of iris texture, even for heavily pigmented regions [2]. As a result, the extracted iris region provides more accurate information implying better chances for recognition. However, the setup complexity with the above mentioned camera is difficult especially when the issue of portability and simplicity arises. On the other hand, smartphones with cameras are within everyone's reach now a days. The only problem with these cameras is that they capture images in visible light spectrum resulting in less detailed iris images compared to the NIR cameras. So the usual question arises "Are they good enough for iris recognition?"

A good number of studies [2,8] replied positively towards the question. The one thing to notice here that almost all the above mentioned approaches paid a little or no effort to state the feasibility or usability of machine learning techniques in case of the smartphone captured iris database. This is important because machine learning techniques have provided very good results in case of NIR camera captured datasets [3]. As iris images in visible light are likely to offer relatively less details, the question of applicability of ML techniques in this case still remain unanswered. However, one studyRaja *et. al*[4] used Sparse Reconstruction Classifier with K-means clustering which gave a very low EER percentage (Equal Error Rate).It is in other words a very good indication but it does not compare or give any further info on the feasibility of other machine learning techniques.

In our work, we investigate further on the use of machine learning techniques on iris recognition using smartphone captured iris images in visible light spectrum. In order to do so we develop a complete segmentation and feature extraction technique and try to use same set of extracted features to train different classifier. Finally, we compare the classification accuracy of the trained classifiers and decide whether the machine learning techniques are feasible in case of smartphone captured database.

II. Related Works

Several works have been done with the publicly available datasets UBIRISv1 [5], UBIRISv2 [6], MICHE [7] etc. containing iris images in visible light spectrum. The challenges of iris recognition associated with unconstrained iris images in visible light were discussed by Proenca *et al* [8]. Noisy iris images and independent segmentation and noise recognition procedures are most likely the sources of errors. Santos *et al.* [9] explored best illumination configurations for visible light iris images.

Raja *et al.* [10] used deep sparse filtering with visible spectrum iris dataset VSSIRIS, BIPLab and obtained a very promising result (EER less than 2%).

Another interesting study from Trokielewicz *et al.* [2] where a completely new dataset was created by themselves, showed that iris images captured with a mobile phone offer sufficient visibility of iris texture details for all level of pigmentation. And they also justified that this images are readily usable with already available iris recognition solutions such as VeriEye [11], MIRLIN [12], OSIRIS [13], IriCore [14] etc. All of these algorithms offered more than 95% accuracy for the dataset.

Machine learning techniques have also been proved to be very successful in iris recognition. A study from De Marsico*et al.* [3] compared different machine learning techniques in iris recognition. In these studies, they used mostly CASIA-Iris [15] dataset which is created from images taken with NIR camera. Among different approaches, Rai and Yadav [16] were able to obtain 99% accuracy with a combination of Support Vector Machines and Hamming distance.

III. Dataset

In our study, the dataset created by Trokielewicz [2] and their group was used. Total 70 people participated and the photos were taken by iPhone 5s (8 megapixels and *f/2.2*). The final dataset comprises about 3192 images acquired in 2 sessions. We used these images for iris recognition. And to the best of our knowledge, no one used machine learning techniques on this dataset before.

978-1-5386-3374-8/17/$31.00 ©2017 IEEE

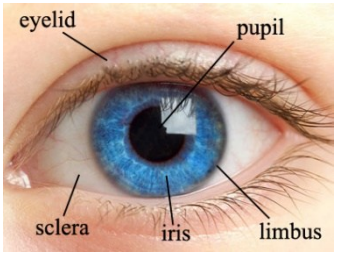


Fig. 1 The typical components in an eye image [20].

IV. METHODOLOGY

A. Image Pre-processing

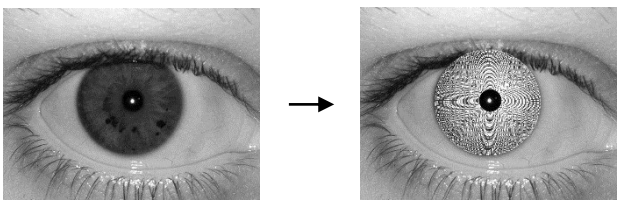
The images provided in the dataset were in RGB format. We had to convert it into a single channel to proceed. While converting, red channel was used as wavelengths corresponding to red light (closest to near infrared) are the longest in our visible spectra, the best iris pattern should be visible this way [2].

B. Iris Localization

For extracting the iris region first, the classic Daugman's Integro-differential operator is used. The Integro-differential operator [17] is defined as

$$max_{(r,x_o,y_0)} \left| G_\sigma(r) * \frac{d}{dx} \oint_{x_o,y_0} \frac{I(x,y)}{2\pi r} ds \right| \quad (1)$$

Where $I(x,y)$is the eye image, r is the radius of the search, $G_\sigma(r)$is Gaussian smoothing function and s is the contour of the circle given by (r, x_o, y_0) i.e. circle of radius r whose centre is at (x_o, y_0). The operator searches for the circular path where there is maximum change in intensity occurs by varying the radius and centre x and y position of the circular contour. First, the iris boundary is localized as the maximum gradient is usually there. Then a fine search detects the pupillary boundary. We used variance ($\sigma = 0.5$) for the Gaussian smoothing function. For faster run the image was scaled down to find (r, x_o, y_0). These values were then rescaled to get the co-ordinates and radius in the original image. Under normal circumstances, this operator successfully localized the iris region. However, if there is any reflection, it might fail locally. So for fine search instead of brute force, we adaptively ran the search for a selected set of points inside the iris region with pupil radius varying from 10% to 90% than that of the iris. By this way, we were able to localize the iris region of all eye images in our database successfully.



Red Channel Localized Iris

Fig. 2 Iris localization

C. Eyelid Suppression

The visible portion of the iris part is not exactly circular. It is partly covered by the eyelids which needs to be suppressed. To do so, we followed an approach inspired from Masek [18]. The total search region was divided into two parts, upper eyelid and lower eyelid. The width of the search region is exclusively the difference between the iris and the pupil radius. First the edges were detected using canny edge detection followed by the gamma adjustment and hysteresis thresholding. Finally, the edge-image was radon transformed to get the eyelid line both for upper and lower sections.

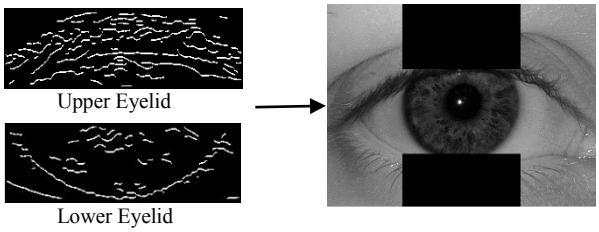


Fig. 3 Eyelid Suppression

D. Normalization

So far we have successfully segmented the iris part and suppressed eyelids. Now we have to transform it into fixed dimensions for further processing. To do that, we used the very popular homogenous rubber sheet model introduced by Daugman [17]. In the homogenous rubber sheet model, each point within the iris region is remapped to a pair of polar coordinates (r, θ) where r is on the interval [0, 1] and θ is angle in the range $[0, 2\pi]$.

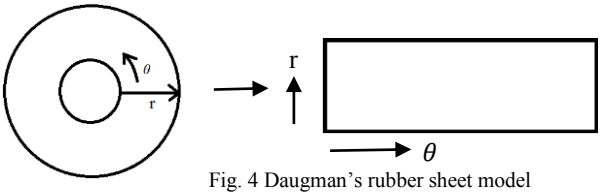


Fig. 4 Daugman's rubber sheet model

The remapping can be modelled as

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta) \quad (2)$$

With,

$$x(r,\theta) = (1-r)x_p(\theta) + rx_i(\theta) \quad (3)$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_i(\theta) \quad (4)$$

Where $I(x,y)$ is the iris region, (x,y) are the original Cartesian coordinates, (r,θ) are the corresponding normalized polar coordinates, (x_p,y_p) and (x_i,y_i) are the centre coordinates of pupil and iris boundary along the θ direction.

Fig. 5 Normalized segmented iris

E. Eyelash Removal

Even though eyelash removal is a part of noise cancellation, it was done after the normalization in our work. Developing a method to do so was a tough job as the eyelashes differ largely from image to image. The most obvious option was applying a threshold. But if a hard threshold value is applied, there is no guarantee that it will work for every image. In some images the iris region was darker than the others. So we had to develop an adaptive algorithm to set the threshold value in each image separately. We did it by analysing the histogram of the normalized iris image. As the eyelashes are usually the darkest parts of the image, histogram inspection led us to find the pixel values of those. These pixel values were used as thresholds to detect the occluded regions. The detected pixels were set to "0" at first and then were restored from the non-occluded regions in the neighbourhood of those pixels.

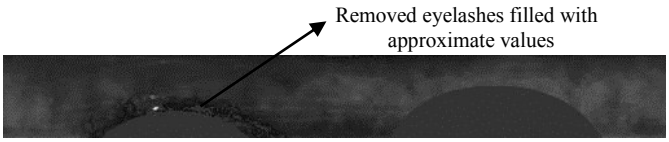


Fig. 6 Eyelash removal

F. Histogram Equalization

Once the iris region is segmented, normalized and noise has been removed, the relevant texture and intensity information needs to be extracted to train a classifier. But before doing that a histogram equalization was performed on the normalized images. This is because the histogram analysis of the normalized image revealed that the image intensities were congested in a very small region making it harder for the classifier to differentiate. In our study we found that the histogram equalization improved the recognition and training accuracy over 2%.

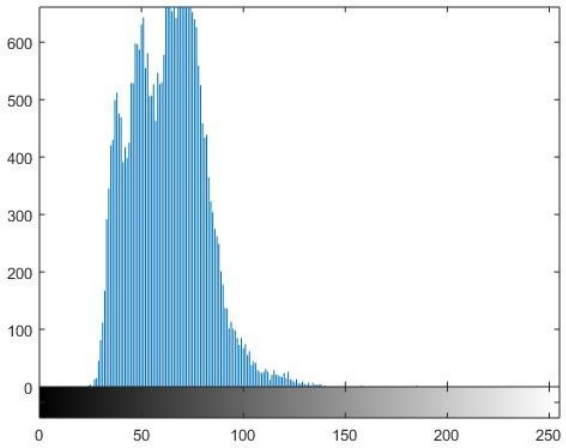


Fig. 7 Before histogram equalization

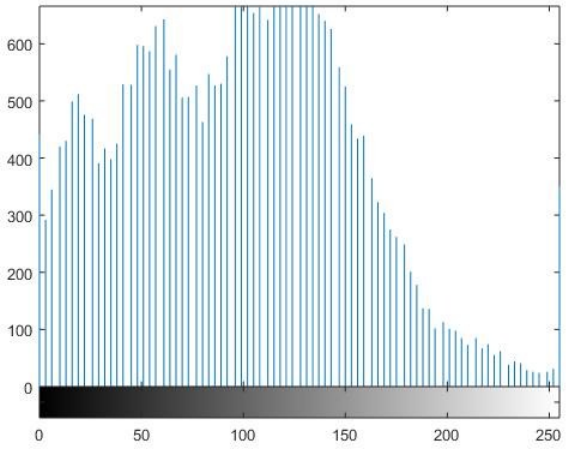


Fig. 8 After histogram equalization

G. Feature Extraction

A typical iris consists of lots of complex patterns such as *arching ligaments, furrows, ridges, crypts, rings, corona, freckles* and a *zigzag collarette*. These complex patterns are very much complicated to extract. That is why we chose to train with the image itself. So far we have a normalized image of size 64 x 512 pixels. If we want to train the classifier with this amount of data, it will be too heavy and training will take a very long period of time. So we need to scale it. But scaling may result in loss of important information.

The solution to this problem lies in wavelet decomposition as the wavelets have localized frequency data i.e. features having same resolution can be matched up. As we know of now that if a 2-d wavelet transformation is applied on an image, it decomposes it into 4 segments: LL, LH, HL and HH. The LL is called the approximation of the image. LH is the horizontal detail, HL is the vertical detail and HH represents the diagonal detail of the image. The most energy and information is contained within the LL coefficients. So these are our desired values. Figure 9 shows the wavelet decomposed iris image after two stages. Finally, we were able to make the image ready for training after successive 3 stages of wavelet decomposition. The LL3 coefficients were taken which contained 8 x 64 = 512 features. For decomposition, Haar wavelets were used.

The extracted feature vector was 2D with dimension 8 x 64. Before Training the classifier, it was converted to a 1D vector of length 512 by placing the rows side by side (Fig- 10).

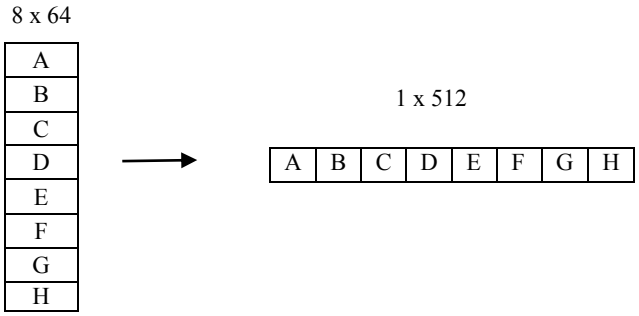


Fig. 10 1-d feature vector of size 512

H. Training Classifier

In the given database, we had eye images of 70 people. For training the model we took 5 images for each person and rest of the images were kept aside for testing the classifier. For training we used 5-fold cross validation method so that each image in the training set can be tested once against the others. We tried several classifier and among them support vector machines, k-nearest neighbour, linear discriminants etc. showed great promise. The results are summarized in the next section.

Fig. 9 Wavelet decomposition after 2 stages

I. Results

For training and testing several classifiers were used. Starting from decision trees, discriminant analysis, support vector machines, RUSBoosted trees, K-nearest-neighbours, Subspace KNN etc. The following table summarizes the best performed classifier accuracy:

TABLE I
CLASSIFIER ACCURACY

Classifier	Train Accuracy (%)	Test Accuracy (%)
SVM (Linear Kernel)	99.1	96.46
SVM (Quadratic)	**99.7**	**97**
KNN	99.4	95.1
LDA	99.4	94.28

It is evident from the above data that Support Vector Machines give the best results. K-nearest-neighbours also performs very well. Though its accuracy is slightly lower than that of SVMs, it takes much less time to train and test. Similar is the case for Linear Discriminant Classifier (LDA). The ROC (Receiver Operating Characteristics) curve matrix attained during training for the best model i.e. SVM with quadratic kernel is given below:

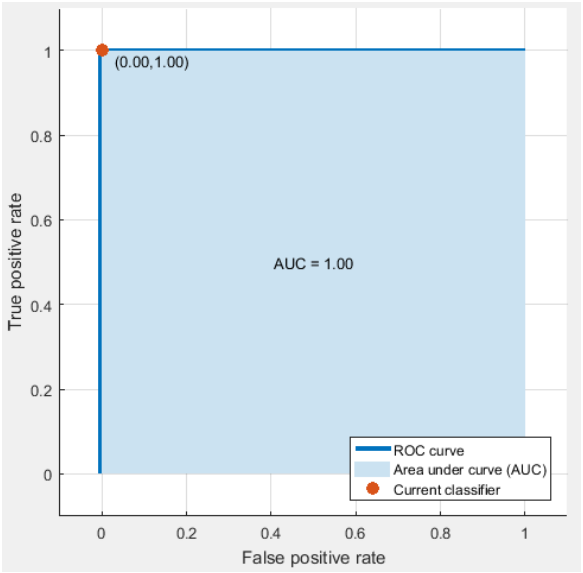


Fig. 11 ROC curve for SVM model

It is evident from the result that our very first doubt about the feasibility and applicability of machine learning techniques on iris recognition from smartphone captured visible light images is well answered through this work.

Now if we compare our approach and its results with the other approaches that is already applied on the same dataset, we see that our approach indeed shows a great promise.

TABLE II
DIFFERENT APPROACHES' RESULT ON SAME DATASET

Classifier	Accuracy (%)
VeriEye[11]	94.57[2]
MIRLIN[12]	95.63[2]
OSIRIS[13]	95.25[2]
IriCore[14]	**99.67**[2]
Proposed Method	**97**

J. Sources of Errors

The Major Source of errors in our findings is failure in segmentation of eye images. Despite our efforts, there were

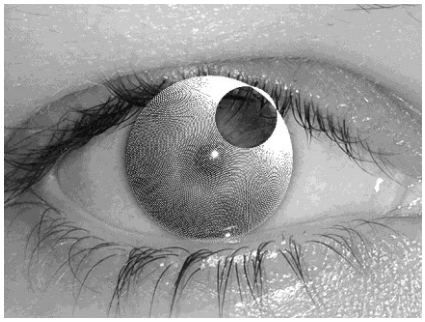


Fig. 12 Failure in correct segmentation

one or two such images whose segmentation could not be done properly. And these are the images who were falsely labelled. Another source of errors are eye images with extremely dark pigments. As a result, it becomes very difficult to extract distinct information from them. Some other noise sources might be blurred images or images with excessive eyelid/eyelash occlusion.

V. CONCLUSION

In this paper a machine learning based approach on iris recognition from smartphone captured images is proposed. With the results above, this paper successfully showed that in case of smartphone captured visible spectrum iris images, the machine learning techniques are equally as good as the other ones, in some cases even better. Still accuracy can be further improved. And in our findings, accuracy largely depends on accurate segmentation. So some robust approaches may be taken to improve the segmentation result. In our approach we tried to stick to some basic segmentation approaches. This was done keeping in mind their easy implementation. As smartphones of todays' are equipped with very good camera, the whole recognition system shows great promise to be implemented on these smartphones for recognition, security and identification purpose. Already Samsung® [19] has developed a built in iris scanner which works for the user who is using it. Our next task would be to develop a cloud based server where iris data can be easily sent through the smartphone. The classifier will run on the server and the sent data would be matched and verified. Thus by just using the smartphones, it will be possible to develop a full security system.

ACKNOWLEDGEMENT

This work was performed in the laboratories of the Dept. of EEE of BUET. The authors would like to thank the concerned authorities of BUET for providing the facilities and help in getting the datasets.

REFERENCE

[1] J. Daugman, "How iris recognition works.," in *IEEE Transactions on circuits and systems for video technology*, 2004.

[2] M. Trokielewicz, "Iris Recognition with a Database of Iris Recognition with a Database of Iris Images Obtained in Visible Light Using Smartphone Camera," in *The IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2016)*, Sendai, Japan, 2016/02.

[3] M. D. Marsico, A. Petrosino and S. Ricciardi, "Iris recognition through machine learning techniques: A survey," *Pattern Recognition Letters,* vol. 82, pp. 106-115, 2016.

[4] K. B. Raja, R. Raghavendra and C. Busch, "features, Smartphone based robust iris recognition in visible spectrum using clustered k-means," in *Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings, 2014 IEEE Workshop on*, IEEE, 2014, pp. 15-21.

[5] H. Proença and L. A. Alexandre, "{UBIRIS}: A noisy iris image database," in *13th International Conference on Image Analysis and Processing - ICIAP 2005*, Cagliari, Italy, Springer, 2005, pp. 970-977.

[6] H. Proenca, S. Filipe, R. Santos, J. Oliveira and L. A. Alexandre, "The {UBIRIS.v2}: A Database of Visible Wavelength Images Captured On-The-Move and At-A-Distance," *IEEE Trans. PAMI,* vol. 32, pp. 1529-1535, 2010.

[7] M. D. Marsico, M. Nappi, D. Riccio and H. Wechslerd, "Mobile Iris Challenge Evaluation (MICHE)-I, biometric iris dataset and protocols," *Pattern Recognition Letters,* vol. 57, pp. 17-23, 2015.

[8] H. Proenca and L. A. Alexandre, "The NICE. I: noisy iris challenge evaluation-part I," in *Biometrics: Theory, Applications, and Systems*, IEEE, 2007, pp. 1-4.

[9] G. Santos, M. V. Bernardo, H. Proenca and P. T. Fiadeiro, "Iris Recognition: Preliminary Assessment about the Discriminating Capacity of Visible Wavelength Data," in *2010 IEEE International Symposium on Multimedia*, IEEE, 2010, pp. 324-329.

[10] K. B. Raja, R. Raghavendra, V. K. Vemuria and C. Busch, "Smartphone based visible iris recognition using deep sparse filtering," *Pattern Recognition Letters,* no. 57, pp. 33-42, 2015.

[11] Neurotechnology, "VeriEye SDK, version 4.3".

[12] Smart Sensors Ltd., "MIRLIN SDK, version 2.23," 2013.

[13] G. Sutra, B. Dorizzi, S. Garcia-Salicetti and N. Othman, "A biometric reference system for iris. OSIRIS," 2014.

[14] IriTech Inc., "IriCore Software Developers Manual," 2013.

[15] http://www.sinobiometrics.com, "Casia Iris Image Database".

[16] H. Rai and A. Yadav, "Iris recognition using combined support vector machine and Hamming distance approach," *Expert systems with applications,* vol. 41, pp. 588-593, 2014.

[17] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," 1993.

[18] L. Masek and P. Kovesi, "MATLAB Source Code for a Biometric Identification System Based on Iris Patterns," in *The School of Computer Science and Software Engineering, The University of Western Australia*, 2003.

[19] "http://www.samsung.com/global/galaxy/galaxy-s8/security/".

[20] "Courtesy: https://www.pinterest.com/".