10.53477/2668-2028-21-37

USABLE SECURITY IN BIOMETRIC AUTHENTICATION SYSTEMS

Adriana-Meda UDROIU, PhD
National Institute for Research and Development in Informatics
meda.udroiu@rotld.ro

Ștefan-Antonio DAN-ȘUTEU, PhD
Colonel, Associate Professor, "Carol I" National Defence University
dan-suteu.antonio@unap.ro

Abstract*: We introduce the term usable security to refer to security systems, models, mechanisms and applications that have as the main goal usability. Secure systems cannot exist without secure authentication methods. Thus we outline biometric authentication methods and we focus on iris recognition because is the most reliable and accurate method for human identification]. The most important advantage of iris biometric over other biometrics is that irises have enormous pattern variability meaning that the variation between individual is almost maximum and variation for any person across time or conditions is minimum. Taking into consideration this observations, this survey covers researches in this field, methods of technical implementation and the usability of this method as an authentication system on iOS environment.*

Keywords: *security; biometrics; authentication.*

Introduction

Personal and even global security have been improved due to biometrics systems and iris appears as one of the main biological characteristics. According to John Daugman[1] iris biometric is stable and remains the same during lifetime. Compared to other biometric systems, iris pattern false reject probability is on in 10^{31} [2]. Also the use of contact lenses or even eye surgery cannot modify the iris characteristics. Taking into consideration face recognition systems is well known that this method does not differentiate between twins. In addition, based on the same article mentioned above twins remain similar to each other in appearance at any age. Although, iris is unique even on monozygotic twins.

Considering iris's uniqueness, we propose in this study a method of authentication on iOS devices based on iris recognition for individuals' identification that cannot be spoofed. In the further sections we present the base algorithm for iris recognition, the convolutional neural networks that will use the same base algorithm to detect fake iris images, the technical implementation and the integration in iOS environment. Also we discuss about the used datasets and the results obtained. In our further work we will try to improve the results already obtained and adapt the iris recognition system to rejects fake irises in order to increase security.

Iris recognition algorithm

Iris recognition algorithm consist into four main steps: segmentation, normalization, encoding and matching. Our work focuses on an analysis of iris segmentation using Hough Transform and Integral Differential Operator techniques for iris recognition system[3].

The first stage, segmentation is the process of extracting features that provide information of iris pattern. We should mention that the original images are converted to a gray-scale images for better performance. Then, the processing of the eye image starts with the detection of two main disks that delimits the pupil from the iris and the iris from sclera. The result of this process is the perfect delimitation of the iris, as can be seen in the image below.

[1] K. W. B. Adam Czajka, *Presentation attack detection for iris recognition: An assess ment of the state of the art. arXiv*, 2018.

[2] . D. Daugman, *High confidence visual recognition of persons by a test of statistical independence. arXiv*, 2000.

[3] A. W. Oad Percy, *Iris localization using daugman's algorithm. arXiv*, 2017.

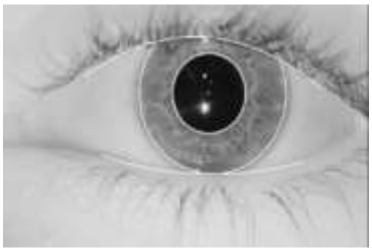


Figure 1. Iris segmentation

The algorithm, named after the professor John Daugman, searches over an image of the eye, pixel by pixel, in order to find the center of the pupil. This is done by calculating at each step the center of the circular contour and the radius. As long as the intensity of the pixels is the same the radius is increasing and once the intensity is changing the difference between the last maximal radius found and the current radius is calculated in order to detect the pupil which is supposed to be the circle with the maximum radius that contains the highest intensity pixels.

Thus, the total number of computations seems to be too large and the method is improved by thresholding. Based on the same observation that, the centre pixels for both the iris and the pupil lie inside the pixels' region with the highest intensity, this method marks some pixels as "object pixels" if those pixels have an intensity value below a certain threshold value. Pixels are considered to be in range of [0;1], where 0 represents the absence of light and 1 the pixel with full light intensity.

After the threshold is applied, image is further scanned, to determine whether the pixel is a local minimum in that particular pixel's immediate 3-by-3 neighborhood. This means that every pixel intensity is compared to the intensities of the pixels in its immediate nine neighborhood pixels. For next calculations the pixel with the lowest intensity value amongst these nine pixels is used. This reduces significantly the number of processed pixels, meaning that the number of steps is divided by nine at each computation.

The main challenges of iris recognition are the detection of the upper and lower eyelid boundaries and the exclusion of eyelashes, eyeglasses or reflection of light. A normal iris that is captured in a video or a picture is not a perfect circle thus the upper and lower boundaries should be treated differently. As Daugman suggested in[4], in detecting the circumferential pixel intensity values, we use 50% of the pixels meaning only those pixels that are at angles of 0 to 45 degrees, 135 to 225 degrees and 315 to 360 degrees as shown below.

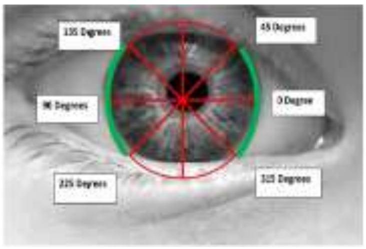


Figure 2. Daugman's suggestion to solve occlusion by eyelid' problem

Also the eyelashes, usually found on Asian people should be eliminated from the iris analysis because even if we try to include them as part of the iris pattern or just ignore small portion if iris covered by them this will affect the entire iris encoding due to the fact that eyelashes can change their position or form depending on different factors and this will lead to false rejections of iris.

In order to avoid light reflection, we use a morphological operator. What we mean by this is that the in this process all the light affected regions will be filled with the average intensity of pixels from the region surrounding it.

[4] J. Daugman, *Recognizing persons by their iris patterns. arXiv,* 2015.

The second step in iris recognition is normalization which refers to preparing a segmented iris image for the encoding process. It measures the range of pixel intensity and binary values using suitable technique to identify the uniqueness in each iris. The motivation is to achieve consistency in dynamic range for a set of data and this is done by changing the intensity of each pixel to obtain a image that fits in a certain intensity range. For example, the normalization process receives as a parameter the intensity desired range. For example, if the initial intensity of the image is between 50 and 180 and the desired range is 0 to 255 the process entails subtracting 50 from each of pixel intensity, making the range 0 to 130.

In iris normalization, the data is trained in order to reduce the errors and a rectangular shape of segmented iris is produced. The output from the normalization technique is stored into the database in two forms: pixel values and iris code which represents the encoding technique, where the iris code is generated.

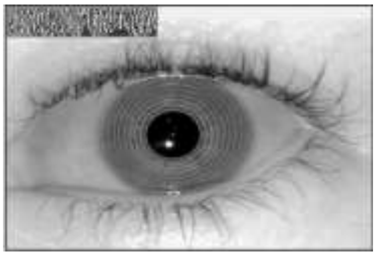


Figure 3. Iris detection and encoding

Further, the iris code will be used for verification process or matching. In the final step, we detect and match the iris code in the database to live iris code.

Attack detection algorithm

Convolutional neural networks have wide applications in image recognition and we will discuss first the basic concepts and then how they are applied in our study.

Convolutional neural networks are like the extremely large interconnected network of neurons in our brain that process information and respond with an output. Basically, each neuron receives several inputs, takes a weighted sum over them, pass it through a function and responds with an output. The entire information processing can be seen as a combination of multiple layers where each layer corresponds to a function. To be more specific the network's architecture is based on four main building blocks or layers: convolution layer, pooling layer, rectified linear units' layer and fully connected layer that repeat themselves. Nevertheless, there are many variations to this architecture but the basic concept remains the same and we will discuss further each of the layers mentioned above.

The convolution layer is the main building block of a convolutional neural network. From a high level, this layer can be seen as a combination of multiple filters, each of them can be thought of as a feature identifier. What we mean by feature is straight edges, simple colors, curves. A filter can be seen like a scanner that analyzes the input image in every region and gives an output corresponding to the fact that the characteristic it was searching for was found in that region or not. Let's say we have an input of 32 x 32 x 3 array of pixel values and our filter of 5 x 5 x 3. As the filter is sliding, or convolving, around the input image, it is multiplying the values in it with the original pixel values of the image. These multiplications are all summed up and we get a single number as a result. We repeat this process for every location on the input volume and we will obtain a 28 x 28 x 1 array of numbers, which is known as a feature map.

Next building block of a convolutional network architecture is a pooling layer. It is a function that progressively reduces the spatial size of a representation to reduce the number of parameters and computation in the network. Pooling layer operates on each feature map independently.

Further, we use rectified linear units (ReLU) as the activation function in the neural network to speed up the process. The computation of e ReLU is very simple meaning that any negative values are set to 0, all other values are kept constant so there is no exponential, multiplication or division operations.

Lastly, the fully connected layer (FC) of neurons network takes an input volume from the previous layer and outputs an N-dimensional vector where N is the number of classes that the program has. For example, if we have a digit classification program, N would be 10 since there are 10 digits and each digit is an independent class. Each number in this N-dimensional vector is called a logit which can be normalized to a probability value between [0,1] by using normalization function such as softmax.

In this thesis, we propose an iris recognition system based on two convolutional neural networks. Thus, the first one will localize the irises on the individual's face and the second one will be trained to recognize the spoof iris images.

Training a network from scratch requires a large number of manually labeled training data and due to the limited number of training samples available in standard iris datasets, we decided to train a fine tuned pre-trained network. For the first CNN we use pretrained YOLO model.

According to [9, 10] the YOLO object detection system is an extremely fast one being able to process input at the rate of 30FPS/s while still having a good accuracy on the ImageNet 1000-classes dataset. At a high level, what makes it soo fast is that while other neural networks apply the process to the full image, the YOLO system divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

The images resulted from the first CNN model that contains only the individual's irises, will represent the input for the second CNN that will identifies if the iris is real or spoofed. The second CNN architecture is also based on a YOLO model like the one presented in[5] and has nine convolutional layers and six pooling layers. All the convolutional layers use 3 × 3 dimension filters. The final layer is the detection layer and it is configured to predict an iris attack attempt

Technical implementation and results

Further, we will discuss the technical implementation of the iris recognition system. Before going on with our discussion we should mention why the simple iris recognition algorithm is not enough and why we need additional convolutional neural networks. Considering[6], we are aware that exists different types of attacks such as: paper printouts, textured contact lenses, image or video displayed on an electronic screen, prosthetic eyes. These types of attacks cannot be detected with a simple iris recognition algorithm and we need a structure that is able to detect spoofed irises. Basically, we need a convolutional neural network.

Taking all mention above into account, we can structure the implementation in three main parts: the base algorithm, the convolutional neural network and the integration in the iOS environment.

The first one, meaning the algorithm that follows Daugman's structure for iris recognition is written in Python3 and uses open-cv, numpy and scipy as the main libraries. On top of this algorithm comes neural networks which have the main scope to detect irises inside a face image and also to detect spoofed irises. Basically, with the first CNN, we analyze an image and extract the two bounding boxes containing each iris. This process will help us to detect the iris inside a image with individual's face. The model is trained using a machine on the cloud with 4 CPUs and 2 GPUs.

Extracting good features is one of the most significant steps in the iris spoof detection part. Here comes the second CNN that uses from Daugman algorithm the segmentation and

[5] A. R. Cunjian Chen, *A multi-task convolutional neural network for joint iris detection and presentation attack detection. arXiv*, 2018.

[6] K. W. B. Adam Czajka, *Presentation attack detection for iris recognition: An assess ment of the state of the art. arXiv*, 2018.

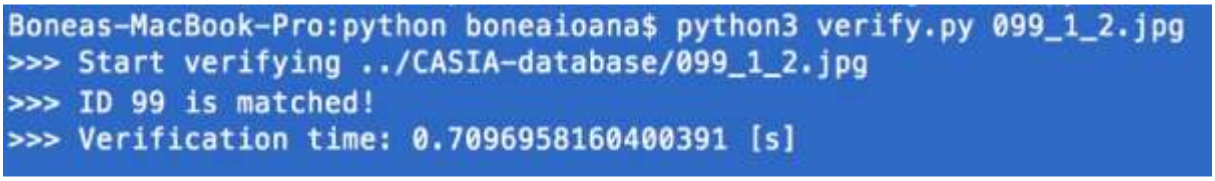
normalization function and tries to extract features from the iris image received as input. This network will detect if the iris is real and will output acceptance or rejection as result.

Using CoreML framework[7] the CNNs are integrated into iOS environment. CoreML allows us to integrate a pretrained machine learning model into our app. It supports Vision library for image analysis. Also, it is optimized for on-device performance, which means a minimization on memory and power consumption. Also an important fact is that the model is running only on the device which ensures the privacy of user data. We converted the pretrained Caffe CNN, which consists of three main files, into a mlmodel using a Python script and coremltools library.

In the next paragraphs, we will discuss image datasets used and the results obtained. The algorithm was tested on CASIA-Iris and Multimedia University (MMU) iris datasets. The first one contains images captured under nonideal conditions characterized by variations in lighting, occlusion, and blur while in the second set, images were taken using the LG IrisAccess 2200 at a range of 7-25 centimeters and contain 450 images, 5 images per iris, 2 irises per subject.

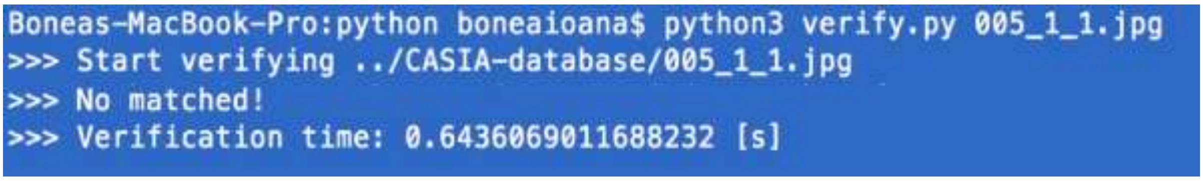
Besides the original images, in the datasets were added iris images taken with an iPhone 8 and X camera for test purposes. Some iris images were registered in the database and as can be seen in the images below if we test the algorithm on one of the legitimate pictures the results iris matches. On the other hand, if the iris encoding is not stored in the database we will receive a result that irises do not coincide.

The results achieved a high verification rate bigger than 95% and the processing time, meaning the iris segmentation, normalization, encoding and matching is less than one second. The accuracy was measured using TensorFlow.



```
Boneas-MacBook-Pro:python boneaioana$ python3 verify.py 099_1_2.jpg
>>> Start verifying ../CASIA-database/099_1_2.jpg
>>> ID 99 is matched!
>>> Verification time: 0.7096958160400391 [s]
```

Figure 4: Matched iris



```
Boneas-MacBook-Pro:python boneaioana$ python3 verify.py 005_1_1.jpg
>>> Start verifying ../CASIA-database/005_1_1.jpg
>>> No matched!
>>> Verification time: 0.643606901168232 [s]
```

Figure 5. Iris doesn't exists in the database

Conclusion

An analysis of the developed iris recognition system has revealed some interesting conclusions. It can be stated that the pre-processing schemes of the algorithm seem to play a significant role in the iris segmentation performance. Thus we will focus on improving the quality of the captured image and getting a stable image of the iris.

Further, the results have shown that the segmentation can be one of the most difficult stages of the process and we will try to extract iris even if the picture is not taken from the front side and we get only half part of the iris.

Another objective is to train the second CNN on new datasets in order to detect an iris attack, thus a system with real-time applicability. Moreover, we will try to achieve good performance on photos taken with mobile device camera and compare the results taking into account multiple factors such as the distance between the camera and the eyes, illumination, picture quality.

[7] A. Developers, *Apple Developers CoreML*, https://developer.apple.com/ document tation/coreml/, 2018.

Recognizing iris images captured using a smartphone in which data can be degraded due to reflection, partial closure of eyes, pupil dilation is a real challenge. In order to perform reliable verification, the set of extracted features should be robust and unique. In this work, we propose a recognition system based on two convolutional neural networks and an adapted form of Daugman's algorithm. Through the experiments on MMU and CASIA iris datasets, we demonstrate the robustness of the newly proposed algorithm which achieves high verification rate bigger than 95%. This research will help with detecting similarities between irises, define characteristics on the human eye and provide a method of authentication in terms of usable security.

Acknowledgements

This research work was supported by a grant on the Romanian Ministry of Innovation and Research, UEFISCDI, project number 8SOL/2018 within PNCDIII, project code: PNIII-P2-2.1-SOL-2017-09-0102, project name: Integrated Information System for Management of Activities (IISMA) (http://siima.pub.ro/en/home/).

BIBLIOGRAPHY

1. CZAJKA K. W. B. Adam, *Presentation attack detection for iris recognition: An assess ment of the state of the art. arXiv*, 2018.
2. CHEN A. R. Cunjian, *A multi-task convolutional neural network for joint iris detection and presentation attack detection. arXiv*, 2018.
3. DAUGMAN J., *Recognizing persons by their iris patterns. arXiv*, 2015.
4. DAUGMAN J. D., *High confidence visual recognition of persons by a test of statistical independence. arXiv*, 2000.
5. YAMBAY A. C. David, *Livdet-iris 2013 – iris liveness detection competition 2013. arXiv*, 2013.
6. DEVELOPERS A., *Apple Developers CoreML*, https://developer.apple.com/ document tation/coreml/, 2018.
7. REDMON A. F. Joseph, *Yolo9000: Better, faster, stronger. arXiv*, 2018.
8. PERCY A. W. Oad, *Iris localization using daugman's algorithm. arXiv*, 2017.