Artificial intelligence in iris recognition

Bartłomiej Szlachta
Faculty of Applied Mathematics
Silesian University of Technology
Kaszubska 23, Gliwice, 44-100, Poland
bszlachta1024@gmail.com

Kamil Rusin
Faculty of Applied Mathematics
Silesian University of Technology
Kaszubska 23, Gliwice, 44-100, Poland
kamirus323@student.polsl.pl

***Abstract*—Containing many characteristic points, human iris is unique for each person. This gives an opportunity to successfully implement people authentication by iris images in many life areas, for example to secure sensitive data.**

The most difficult part of iris recognition is to extract the iris characteristic points and to compare them with those extracted from other iris image. During our research, selected artificial intelligence methods such as Speeded Up Robust Features and Soft sets has been analysed in regard of recognizing people by their iris image. As a result, three application concepts has been described and proposed. Two of them have been also implemented and tested on thousand of iris images. The results have been presented and compared in order to find the most suitable method for iris recognition. Moreover, possible concepts future improvements have been described in order to allow recognition effectiveness improvement.

I. Introduction

Artificial intelligence gradually takes place in our lives and changes the world around us [1], [2]. Neural networks, soft sets, heuristic algorithms are often used in many real world problems, such as

- Recommendations [3],
- Voice assistants [4], [5],
- Recognizing images [6],
- Autopilot [7],
- User verification [8].

Those can also be used in recognizing human iris to provide the best results possible.

Behind each system there exist complex algorithms and methods. As a result, there is no one way to always successfully recognize people's iris. Eye image quality is the most significant aspect. For different images they perform differently, so the results might vary from each other. In this paper we analyzed the example of user verification using the recognition of eye iris based on one eyes images database.

Our research was focused on implementing recognition system providing user registration and logging in possibilities.

II. Proposed approach to the problem

During our research, we prepared three versions of desktop application allowing people to register and login themselves. Each of them has the same principle of operation but differs in used methods and concepts. Each version is described more widely later in this article. Each version has following functionality:

- registration using username and four eye images,
- logging in using username and one eye image.

Using the first variant, to make SURF algorithm (described below) work properly, each image must fit into requirements:

- photo dimension must equal 200x150 px,
- eye must be located in the middle of the image,
- iris radius must be about 45 px,
- pupil radius must be about 10 px,
- upper eyelid must not be located below 62 px from the top edge.

Moreover, in the second application variant an iris images database is required to be located on the computer's drive under a certain path. This requirement is due to usage of soft sets which need to create an example people when the application starts.

For the third variant there is no application written but an idea is prepared and is going to be implemented in the future.

In order to test the functionality of the variants an "UBRIS v1" database was used[9]. The database contains 471 photos of different eyes, 5 images for each eye, each image meets the requirements mentioned before.

Example eyes:

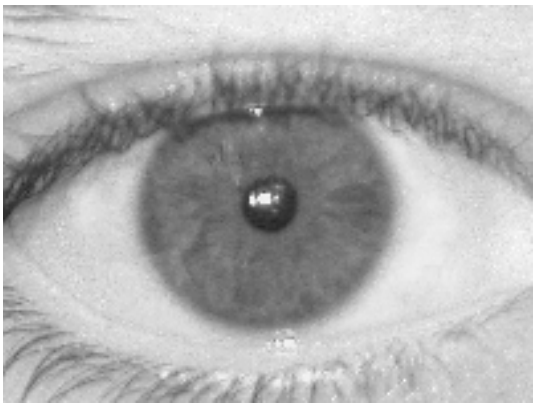

Figure 1: Example eye image

Images are divided into 2 sets:

- Sessao_1,
- Sessao_2.

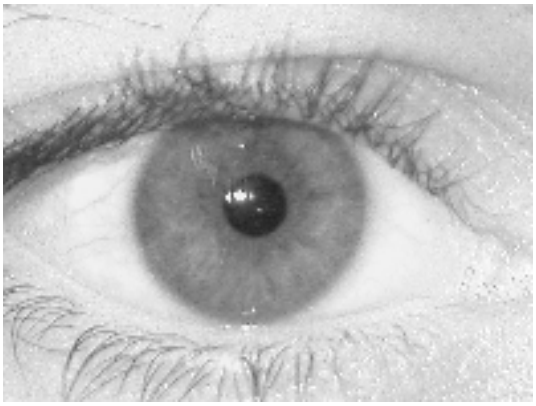

Figure 2: Example eye image

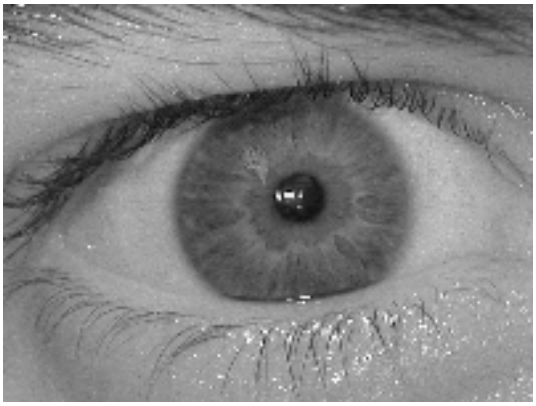

Figure 3: Example eye image

In the second variant of the application, virtual users are registered using the pictures in the Sessao_1 folder.

III. SURF ALGORITHM

Speeded Up Robust Features is used to search for key points in the images [10]. It is inspired by SIFT (Scale-Invariant Feature Transform) descriptor [11]. Several characteristics of this algorithm are:

- it is three times faster than SIFT,
- independence from scaling and rotation operations.
- recurrence,
- fast detection of the point of interest,
- faster matching of descriptors.

Its principle of operation is as follows. First, the algorithm converts the analysed object into a gray scale image. For each pixel, the value in the above-mentioned scale is calculated according to the following formula for each R,G,B values of pixel:

$$X = \frac{R + G + B}{3} \tag{1}$$

Then, the corresponding functions extract key points in the image. 64 elemental vectors called descriptors are calculated for each point. For each point, a circle with a centre at the given point and a radius with a size depending on the strength of the descriptor is determined. There are two processes that influence the effectiveness of local image descriptors: extraction of distinctive features around each point and determining the location of characteristic points. The detectors are unusual in the group of affine-resistant transformations detectors, as [12]. The algorithm is also based on the Hessian matrix, which is defined as follows.

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{2}$$

Where $L_{xy}(x, \sigma)$ are partial derivatives of the second order at the point of the image $I(x)$ in specific directions, smoothed with the Gaussian nucleus with the σ parameter.

As the characteristic points are selected those points of the image which constitute the local maximum of the determinant and the trace of the Hessian matrix with the given formula (2). The determinants of Hessian matrices are sorted from the largest ones. They are a measure of local changes around a point. The larger the determinant, the larger the descriptor and the less important point.

The vector of features is created in the following way. A certain orientation is assigned to each key point. Then, a square area is built around this point. This area is also set according to the designated orientation for the point. It is divided into smaller areas with dimensions of $4x4$. In this way, we keep important spatial information. For each of the subareas, features for exemplary points distributed regularly are counted in the $5x5$ mesh vertices. The dx and dy values are calculated using the Haar Wave. Then they are added to each of the sub-regions and they build the first set of features (vector). The calculated values with the absolute values form a four-dimensional descriptor 4. Calculations are made for each of the sub-regions with a size of $4x4$ 5, then vector of features (descriptor) with a length of 64 is obtained.

The filtration results are independent of the brightness of the image. Moreover, independence of contrast is achieved by transforming the descriptor into a unit vector.

IV. SOFT SETS

The definition of soft sets and theirs entire mathematical inference system is described in [13]. In [14] the soft set is defined as follows. U is an initial universe set and E is a set of parameters. $P(U)$ is the power set of U and A E. Then, a pair (F, A) is called a soft set over U, where F is a mapping given by:

$$F : A \to P(U) \tag{3}$$

Soft sets is a relatively easy method to implement. The initial description of the object is approximate, so there is no need to set the concept of exact solution. Parameterization can be any, including words, sentences, functions, mappings and numbers.

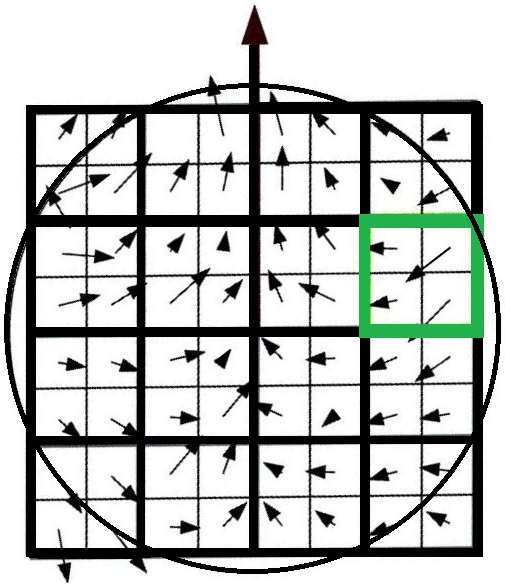

Figure 4: Descriptors

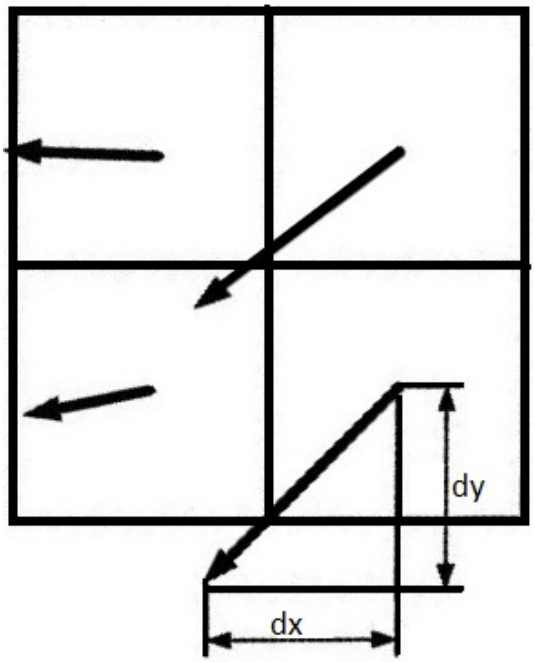

Figure 5: dx and dy

First variant of the application uses soft sets to verify a user. They are used to infer which of the registered users suits the most entered picture during logging in. If the user selected by the soft set system is the one whose username has been entered, the login is correct. In order to make the system work properly with a small number of registered users, a number of false users is registered using eye images from the database. The number of registered users is set to 100 by default, however this value can be parameterized. Sample photos used for this purpose are described before in this paper. Each of the

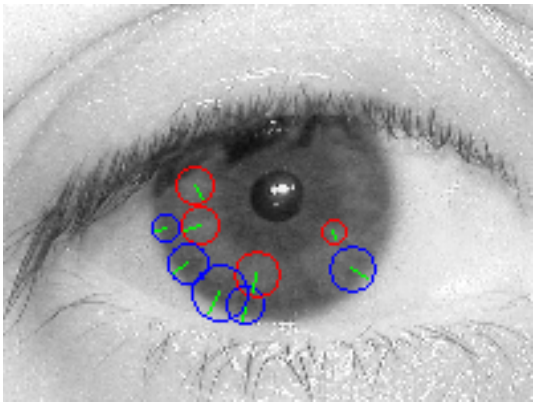

Figure 6: Key points detected

entered images is processed by the SURF algorithm to obtain key points. Then the picture is divided into square fragments. The fragment size is set to $5x5$ by default, however this size can be parameterized. For each of the fragments, the following features are saved:

- average pixel colour values,
- minimum pixel colour values,
- maximum pixel colour values,
- the 5 most significant key points that SURF finds in this part of the image.

Points are compared on the basis of the SURF Point class Scale parameter, the smaller the parameter value, the more significant the point is.

During the user's registration, data from all four entered images is saved along with the username. The principle of operation of soft sets for matching the most suitable user is described using a pseudocode later in this paper.

V. Application concepts

A. First concept: compare SURF points

In this concept we use SURF algorithm, explained above, to detect key points on the eye images. We use default Accord.Net SURF Detector parameters. While registering, key points are being extracted from each image. Because the points vary depending on the image, we need to compare them and exclude those which appear very rarely.

We assume that two points are similar when they meet each of the following requirements:

- Their X coordinates differ by less than 10 px,
- Their Y coordinates differ by less than 10 px,
- Their Laplacian parameters are equal (they are a part of Accord.Net SURF Point),
- Their Scale parameters differ by less than 0.3 (they are a part of Accord.Net SURF Point),
- Their Orientation parameters differ by less than 0.15 (they are a part of Accord.Net SURF Point).

As the result of points comparison we get a list of similar points collections, whose length represents the amount of similarities found. Then the list is saved to the user. When

it comes to logging in, points are again extracted from the image and compared to those similar points extracted while registering. If an assumed points number from the new image are similar to the saved points and usernames are equal, login procedure is successful. By default, the number of similar points required to log in is set to 2 points. This means that at least 2 similar points collections must be found in the registration process. Due to the comparison algorithm dependency on SURF algorithm, it is not certain to occur so some of the registration processes may be unsuccessful.

The concept pseudocodes are as follows:

```
for each image entered do
    search for key points;
end
group corresponding, frequently occurring points;
for each similar points set do
    add surrounding pixels values;
end
if similar points sets list contains at least 2 sets then
    save the user and his eye data;
else
    unable to register;
end
```

Algorithm 1: User registering pseudocode

```
search for key points for the image entered;
matches = 0;
for each point found do
    for each set in userEyes list do
        for each point in the set do
            if points in set is similar to the searched
              point then
                matches++;
                break;
            end
        end
    end
end
if matches > 2 then
    logged in;
else
    unable to log in;
end
```

Algorithm 2: User logging pseudocode

The results are as follows:

First variant of the application was tested for 246 different eyes. For each of them, a user was registered using the first four images of the eye, then there were attempts to log in using fifth image of the same eye and using each image of another eyes. This gives 135 registration attempts and 45401 login attempts. Registration success represents the percentage of how many times in 4 images algorithm found at least 2 similar key points. After successful registration fifth image of user's eye was used to log in right account. The last statistic represents how often user logged into account using another user eye.

Statistics:

- registration success: 58,51% (79 of 135),
- right log in: 84,81% (67 of 79),
- false log in: 12,34% (5 602 of 45401).

For the application concept to reach greater efficiency, we can change the minimum similar points collections required to log into account. The parameters can be higher than used in this parameter. It can lead to finding a less similar points, but amount of them will rise and the algorithm may recognize user with different key points than at the beginning state.

B. Second concept: use soft sets to compare images fragments

In this concept SURF algorithm is not essential and soft sets are much more important. The soft sets' operation is to find the best option from the available ones. In our case, we choose from the registered users the most suited one to the entered image. If the chosen user is the user we want to log in as, the login attempt is successful. In order for this system to work properly, a certain number of users is registered when the application starts, creating a virtual users database. If the database is not created, the first user would always pass the verification because he would be the only user registered and always the most suited among all users, regardless of the image entered. Obviously, the more users registered, the lower system effectiveness. To compare users, we use a data extracted from the images and processed the different way compared to the first concept. Each image is divided into small square fragments. For each fragment the average, minimum and maximum pixels values are computed and saved to user. We also save the SURF points values to the fragment if the points lie on the image fragment. While logging in, for each fragment a value which represents the similarity of the corresponding fragments in both considered images is computed. Then, for each image the fragments' values are summed up, computing the final value. Then the final values are compared in order to find the greatest value which corresponds to the most suitable user.

The concept pseudocodes are as follows:

The results are as follows:

Confusion matrix was made for this variant of application. Confusion matrix is created from the intersection of the predicted class and the class actually observed. There are 4 cases: 2 for compliance and 2 for the non-compliance comparing to actual state. These are:

- True-Positive (TP): positive prediction and actually observed positive class (i.e. the right user and a positive login result),
- True-Negative (TN): negative prediction and actually observed negative class (i.e. fake user and negative login result),
- False-Positive (FP): positive prediction and actually observed negative class (i.e. right user but negative login result),

```
for each image entered do
    search for key points using Accord library;
    divide the image into fragments;
    for each fragment do
        Count average pixel values;
        Count maximum pixel values;
        Count minimum pixel values;
        Choose the 5 most important points located in
         the fragment;
        Save above data to the image data;
    end
    Save the image data to the user;
end
```

Algorithm 3: User registering pseudocode

```
Process the image and extract data;
for each user registered do
    Count the assess value (soft sets) using data from
     image entered
end
Select user with the greatest assess value;
if the chosen user is the user concerned to log in as
 then
    logged in;
else
    unable to log in;
end
```

Algorithm 4: User logging pseudocode

- False-Negative (FN): negative prediction and actually observed positive class (i.e. wrong user but positive login result).

The test of this concept was conducted for database of 200 registered users. Number of users is treated like a parameter, it can be defined at compile time.

Statistics are as follows:

- True-Positive: 1,
- True-Negative: 100,
- False-Positive: 99,
- False-Negative: 0.

Results are not reliable. Number of False-Positive is high, due to really low chance to log into users account. For the application concept to reach greater efficiency, we can add another features that can be crucial in recognizing right user. Also the image of the eye can go through various filters before being analysed. It may help in finding more key points.

For the application concept to reach greater efficiency, for each fragment we could add weights to make fragments containing iris become more important. We could also remove SURF algorithm usage, which makes the concept too complicated to work properly.

C. Third concept - mixed

This concept has been not implemented yet but it is fully considered and is going to be implemented in the future. It is a mix of first and second concept having taken the positives from both concepts and refusing the downsides. It assumes that detected SURF points would be compared in order to find similarities just like in the first concept. The difference would appear in the result which would take a form of coordinates representing the movement of each image. This would remove the problem of iris being located in the different image coordinates, not always in the middle of the image. Then, the points found in all of the images would be filtered to get the most important ones. Having the points' coordinates adapted including the images movement, it would be possible to extract image fragments only from the most important points' locations, not the whole image. Then, the soft sets system could be implemented with the fragments' weights as explained above but there are some other possible ways. Those include using neural networks or fuzzy systems.

VI. Conclusions

In order to test our application, we programmed few functions that tried to register and log in right and fake users. As mentioned in this paper, first concept has as two parameters:

- similar key points needed to register,
- similar key points needed to log in.

With $a = 2$ and $b = 1$, there were 83 right registrations and 75 successful first try of logging in. Probability of right user to log in is 90,36%. The result are slightly better than default ones. There is higher possibility for registering in database and recognizing user. There were 52 exceptions during the registering process, because algorithm did not found required number of similar key points. The higher are the parameters, the lower probability of both events. Moreover, the probability of users to log in also starts to decline. This should be opposite, because, more unique key points are found during registration process. When a parameter equals 5 and b equals 4, there is no chance to log into user's account. There is not any pack of eye's images that provides required number of similar key points to algorithm. There was an exception in all cases during registering process.

There is also a second possibility. More key points required to register, but less to log in. We set 'a' at its maximum value, that allows users to register ($a = 4$) and b is equal to 1. Still probability of right users to log in (82,35%) is less than at default parameter values ($a = 3$, $b = 2$). There is one interesting observation. If parameter b is equal to 1 or 2, there is no difference in number of right registered users.

In order to test the second concept we modified number of fake users in the database. Probability varies around 40-50%. That number is not reliable, because of high value of True Negative. In reality it is almost impossible to log in, as tested with 64 users, only one managed to do that.

Accuracy (%)	100	25	62,5	43,75	46,87	51,56	50,78
N	2	4	8	16	32	64	128

Table I: N - Number of users in a database

REFERENCES

[1] G. Capizzi, G. L. Sciuto, P. Monforte, and C. Napoli, "Cascade feed forward neural network-based model for air pollutants evaluation of single monitoring stations in urban areas," *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 327–332, 2015.

[2] F. Bonanno, G. Capizzi, and G. L. Sciuto, "A neuro wavelet-based approach for short-term load forecasting in integrated generation systems," in *2013 International Conference on Clean Electrical Power (ICCEP)*. IEEE, 2013, pp. 772–776.

[3] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2018.

[4] D. Połap, "Neuro-heuristic voice recognition," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2016, pp. 487–490.

[5] D. Połap, M. Woźniak, R. Damaševičius, and R. Maskeliūnas, "Bio-inspired voice evaluation mechanism," *Applied Soft Computing*, vol. 80, pp. 342–357, 2019.

[6] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[7] A. B. Farjadian, A. M. Annaswamy, and D. D. Woods, "A shared pilot-autopilot control architecture for resilient flight," *IEEE Transactions on Control Systems Technology*, 2018.

[8] L. B. Kimon, Y. Mirsky, L. Rokach, and B. Shapira, "Utilizing sequences of touch gestures for user verification on mobile devices," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 816–828.

[9] H. Proença and L. A. Alexandre, "Ubiris: A noisy iris image database," in *International Conference on Image Analysis and Processing*. Springer, 2005, pp. 970–977.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[11] Wozniak, M., Napoli, C., Tramontana, E., Capizzi, G., Sciuto, G. L., Nowicki, R. K., Starczewski, J. T., "A multiscale image compressor with rbfnn and discrete wavelet decomposition," in IEEE International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1-7.

[12] P. Dalka, "Metody algorytmicznej analizy obrazu wizyjnego do zastosowań w monitorowaniu ruchu drogowego," 2015.

[13] P. K. Maji, R. Biswas, and A. Roy, "Soft set theory," *Computers & Mathematics with Applications*, vol. 45, no. 4-5, pp. 555–562, 2003.

[14] D. Molodtsov, "Soft set theory—first results," *Computers & Mathematics with Applications*, vol. 37, no. 4-5, pp. 19–31, 1999.