Multibiometric System for Iris Recognition Based Convolutional Neural Network and Transfer Learning

Huda Moyasar Therar¹, Lect. Dr. Emad Ahmed Mohammed² and Asst. Prof. Dr. Ahmed Jadaan Ali³

¹,² Dept. of Computer Engineering Technology, Northern Technical University, Mosul, Iraq
³Dept. of Power Engineering Technology, Northern Technical University, Mosul, Iraq

Abstract. Multimodal biometric methods have been commonly used by several implementations because of its capability to work with a variety of important drawbacks in unimodal biometric methods, such as noise affectability, populace coverage, intraclass variety, vulnerability to spoofing, and non-universality. In this research, a multimodal biometric real-time method is suggested depending upon the design of a deep learning model for pictures of a person's (right & left) irises. This system has been implemented by combining the characteristics of convolution neural networks and transfer learning techniques. Through this research, the training system focused on a collection of the backPropagation technique with Adam’s optimization approach utilized to modify weights and adjust learning rates during the learning process. The efficiency of the system is examined on two public datasets obtained in various conditions: IITD and CASIA-Iris-V3 Interval. The implemented system gives an accuracy of 99% for both left & right IITD iris datasets and accuracy of (94% and 93%) for the left and right iris for CASIA-iris-V3 interval datasets respectively after training. OpenCV library for image pre-processing, Keras, and sci-kit learn python libraries for feature extraction and recognition has been utilized.

Keywords: Multimodal biometric, Iris recognition, Deep learning, Convolutional Neural Network (CNN), Transfer learning.

1. Introduction

Nowadays, biometric authentication and human identification are critical and reliable methods for identifying objectives. Biometric systems are developing applications that can be utilized in automated processes to uniquely and effectively identify persons and better alternatives to conventional methods like passwords.

Biometrics is an automatic technique that is depending upon a physical or behavioral characteristic to authenticate an individual. Physical properties, including the face, voice, fingerprints, even iris, behavioral properties are features that will be developed or gained, like keystroke dynamics, dynamic signature identification, and speaker identification [1]. For all the particular physical characteristics, iris' biometrics is considered highly reliable and difficult to duplicate and replicate [2]. The iris is a visionable structure, yet preserved. Iris always does not vary over the years; therefore, it is ideal & effective for encrypting identity [3].

Iris feature offers a set of significant advantages over all different biometric features (like a fingerprint, face), making it widely adopted for use in high efficiency and reliable biometric methods. First of all, the iris feature describes the eye's circular region around the black pupil & the white sclera; therefore, it is fully safe from different external conditions. Secondly, the iris pattern is
considered to provide a great extent of individuality, so randomness and no pair iris patterns may remain the same, even irises for similar twins, or a person's right & left eyes. Thirdly, each iris characteristic offers a higher level of stabilization during a human's life from one year of age till death. Finally, this is viewed as the best reliable biometric function toward deceitful approaches and hacking threats via an imposter. All efforts to vary its patterns and surgery are a strong risk, whereas the fingerprint feature is considerably simpler to manipulate [4].

Despite these benefits, the iris identification method's implementation is a difficult issue according to the iris acquisition process that may acquire insignificant parts like eyelashes, eyelids, pupils, and specular reversal, which can have a significant impact on iris segmentation & recognition results.

The major contribution of this research is to study and evaluate the current approaches to the identification of iris features and suggest enhancements to the production of iris features, create a practical and robust iris recognition system, classification algorithms that could improve the accuracy of conventional iris classification, and Development and execution of a robust iris biometric feature system, obtained from the same person's right and left eyes.

The most effective and widely accessible iris recognition method was proposed by Daugman [5]. The model was utilizing an integro differential operator to locate the circular iris & pupil part. Then the iris pattern is transferred to a normalized shape using the rubber sheet method of Daugman. This is continued to follow by using a two-dimension Gabor filter to obtain their iris features with the Hamming distance to decision making. Daugman's method's main drawback is that it needs a high_resolution camera for taking the iris image. In the form of recognition utilizing deep learning, Du al. [6] designed a system that automatically identifies left against right iris images using a convolutional neural network. This approach will be applied to various datasets, obtaining a classification performance of 97.5%. The authors in [7] described a methodology of iris segmentation and recognition dependent on several convolutional neural networks (CNN) with multiple networks operating in various resolutions. The method provided accuracy of 95.63%, 99.41%, and 93.17% on the following databases: CASIA-Iris-Thousand, UBIRIS.v2, and LG2200, respectively. The author in [8] suggested multi_patch deep feature extraction utilizing deep coarse filters to construct a robust smartphone iris authentication method utilizing the visible spectrum. This approach is tested on the MICHE_I database where the researchers registered an Equal Error Rate (EER) lower than 2 %. The authors in [9] proposed improving iris recognition efficiency within non_frontal pictures by multi_spectral combination of the iris pattern with scleral texture. In [10], the authors developed a recent iris dataset containing NIR with VL iris images collected concurrently and executed studies to enhance cross sensor & cross_spectral iris matching. Most recently, the proposed framework in [11] is developed by fine-tuning a pre-trained convolutional neural network (VGG-16) for feature extraction and classification. The author in [12] provided a comparative review of the performance of three deep learning models for iris recognition; Alex net, Vgg16, and Vgg19, and compare them with a traditional algorithm (Masek). Three deep learning models' accuracy was 100%, 97.88%, and 97.5%, respectively. The authors in [13] redesign the implementation of iris recognition via utilizing real attributes. In which they tried the interaction between two iris constructs by uses of Hamming detachment. The accuracy of the system is achieved on the dataset CASIA-V4 Lamp dataset gives 98.34 %.

The paper's remnant is arranged as follows: The iris recognition system's achievement in section 2. The results and discussion of the implemented system are described in section 3. Conclusions and future work are mentioned in section 4.

2. PROPOSED IRIS RECOGNITION SYSTEM
A brief overview of the suggested deep learning method is provided in this section that utilizes two techniques of discriminative learning: a CNN with transfer learning.
2.1 Convolutional Neural Network

The Convolutional Neural Networks (CNN) is deep-learning network architecture, and it can be automatically learning representations of image features. Also, it outperformed other traditional handcrafted technical features. The Convolutional Neural Network (CNN) architecture generally consists of 5 layers: (an input, a convolutional, a pooling, a fully connected, and a SoftMax) layer [14]. CNN tends to be structured in two parts. The first part is named the feature extraction, which uses convolution and pooling layer groupings. The second part is called classification uses a fully connected layer. The architecture of Convolutional Neural Networks (CNN) provides three properties that make it powerful and efficient in recognition tasks: local receptive fields, downsampling operations, and weight sharing.

In CNN, only a small area of input neurons is linked to the neurons in a hidden layer called this region, the local receptive field. The local receptive has the same Convolutional filter size. To create a feature map, the local receptive field is translated over an image. In convolutional & downsampling layers, a local receptive field is used. In CNN, the weight in all neurons in a given layer is the same. In CNN, the sharing of weights is added to the convolutional layer to monitor the capability and reduce the model's complication. Nonlinear downsampling is utilized in the downsampling layers to minimize the image's dimensional size and reduce the model's number of free parameters. These layers are:

1. **Input Layer**
   It is the entry of the neural net and contains each of the kernel pixels of the training image.

2. **Convolutional Layer**
   This layer executed the input image convolutional process using a collection of filters called the kernel. The feature map is the convolution operation output. Figure 1 displays the first convolutional layer comprising six filters, which generated six feature maps organized together. The gray squares refer to the feature maps, while the green squares refer to the convolution filter. Those cross lines between these last two layers refer to their neurons that are fully connected. The process of Convolutional [15] is mentioned in equation (1).

\[
y^i = \sum_j (b^i_j + w^i_j \ast x^j) \tag{1}
\]

Here \(x^j\) is the input of the previous layer, \(w^i_j\) represents the weight across the \(x^j\) & \(y^i\) while \(b^i_j\) is a neuron bias via the input channel & the output channel, \(y^i\) is a neuron in the \(i\) output(*) convolutional operation.

![Figure 1. A description of the architecture of the Convolutional Neural Networks[4].](image)
3. The Rectified Linear Unit (ReLU) layer.
It has been used as the popular activation function after each Conv layers and FC layer (except the latter fully connected layer) in the CNN architecture.

4. Pooling Layer
This layer is used to decrease the convolution layer's output while preserving the most important information found in the input layer. The pooling operation may be (average or max), the max-pooling being the most popular pooling process utilized in this research.

5. Fully Connected Layers
These layers utilized the features extracted within the preceding layers to perform the classification function [16]. The final convolutional or pooling layer's output is supplied to the fully connected layers as in a main neural network.

3. The Proposed System
In general, an iris recognition system consists of multiple stages: image acquisition, segmentation, normalization, feature extraction, and classification, as seen in Figure 2.

At first, a photo is taken from the iris; a preprocessing technique is performed based on applying an effective & automated iris segmentation to closely identify the iris area from the surroundings, including pupil, eyelids, sclera, eyelashes, & specular reflections. Each iris area has been converted to a normalized model with fixed dimensions after detection to directly compare two iris images with different original sizes.

Furthermore, this normalized iris image is utilized to supply reliable and unique iris features when using the CNN and SoftMax classifier as an automated feature extractor. Next, the extracted CNN feature vector is subsequently inserted into the multi_class (SVM) classification module. Then distinguish it into one of N class. Finally, the corresponding scores of each right & left iris images are combined to determine the identification of the person whose iris images are being examined. Through each training process, CNN implementations being trained on the training set & tested on the validation set to get the finest and the least error that we name (IrisConvNet).

![Figure 2. The implemented multi_biometriciris recognition system](image)

3.1 Iris Segmentation
It is an important process to distinguish the iris portion from the other parts of the eye and determine the iris’ external and internal boundaries [17]. However, this process is more complicated when portions of the iris are hidden by eyelashes & eyelids. Additionally, changes in lighting conditions may reduce the quality of that extracted iris area through the acquisition process and then change the iris’ segmentation and the result of recognition. An automated algorithm is suggested to detect the iris's inner & outer boundaries, as shown in Figure 3. An automated algorithm is suggested to detect the inner & outer boundaries of the iris. The inner & outer edges are detected using an effective improvement method depending on the two-dimension Gaussian filter with a histogram equalization process to decrease the mathematical difficulty of the Circular Hough Transformation (CHT), which
smooth the image of the eyes & improve the contrast between the iris & the sclera area. Then a robust CHT is implemented to determine both center coordinates & the radius of each pupil & iris circles.

![Figure 3. Iris segmentation](image)

### 3.2 Iris Normalization

Eye images might be of different sizes because they have been taken from various persons and different environments. Therefore, images should be normalized from different sizes into the same size to achieve more accurate recognition irises. This process is done by transforming the region of the iris into a rectangular region. Daugman’s Rubber Sheet model is widely used for normalization [18]. The rubber sheet algorithm remains every pixel in the segmented iris region in the Cartesian (x,y) coordinates to the polar coordinates (r,θ) Where r is at the [0,1] interval, and where (θ) is at the [0,2π] angle, as shown in Fig.4. The mapping can be described below.

\[
I(x(r, θ), y(r, θ)) \rightarrow I(r, θ)
\]

\[
x(r, θ) = (1 - r)x_p(θ)rx_i(θ)
\]

\[
y(r, θ) = (1 - r)y_p(θ)ry_i(θ)
\]

In which I(x,y) is the region of the iris, (x,y) is the main Cartesian coordinates, (r,θ) is the conformable polar coordinates x_p, y_p & x_i, y_i are the pupil & iris region coordinates along θ the direction[19].

![Figure 4. Daugman’s rubber sheet algorithm to convert the iris area in the Cartesian coordinates into the polar coordinates.](image)

### 3.3 Feature Extraction using deep learning

After iris normalization has been created, extraction features and classification are implemented utilizing a deep learning method incorporating a CNN with transfer learning. The design of the suggested CNN structure is shown in Figure 5. It’s made up of seven layers. CNN uses a fixed size input image, so each training sample is repositioned to 64x360 px. CNN layers are partition into 2 sections, the whole first four layers are used for the extraction of features, and the following three hidden layers are liable for the classification of features.
The first layer (L1) is a convolutional layer with six filters, each with a size of 3x3. This layer implements the convolutional operation by convoluting the filters with the given input 6@64x360. The output features map is obtained nine workable weights with a workable bias; Thus, the layer contains 60 ((6 x 9) +6) trainable parameters. With Max pooling dependent on non-overlapping, including 2x2 kernel sizes in x and y distances generates six feature maps, each of size 179x32.

Layer 2 (L2) is the convolutional layer also. It includes32 kernels with the input image and 32@30x177 and the output features map is obtained nine workable weights with a workable bias, thus, the layer contains 1760 trainable parameters. Max pooling dependent on non-overlapping including 2x2 kernel sizes in x and y distances generates 32 feature images of 15x88 each.

Layer 3 (L3) comprises of two layers like in (L2). Employing a 3x3 bit in the convolutional layer, (64) feature of every dimension 13x86 are made. This is also decreased to the size of 6x43 in the maxpooling layer and 64 feature maps get of the layer (L3). We have 18,496 trainable parameters within the layer.

Layer 4(L4) comprises of two layers like (L2&L3). Employing a 3x3 bit in the convolutional layer, 256 features of every dimension 4x41 a are made. We have 147,712 trainable parameters within the layer.

The next three possible layers are necessary for the classification of features. The hidden layer one was made up of 1024 neurons. The hidden layer two has been arranged, 512 neurons. The hidden layer three has been contained 80 neurons showing the number of classes of the input data.

The system implemented has been programmed to utilize ADAM optimizer with (0.0001) learning rate, 0.0005 decay, and 32 batch size. Methodological cross-entropy is often utilized to measure the loss function, and the network variable has been modified to reduce the loss of prediction. The detailed definition of CNN layers is shown in Table 1.

| Layer (type)                  | Kernel | Kernel Size | Output Shape     | Param |
|-------------------------------|--------|-------------|------------------|-------|
| conv2d_1 (Conv2D)             | 6      | 3x3         | (64,358,6)       | 60    |
| max_pooling2d_1 MaxPooling2   | 2x2    |             | (32,179,6)       | 0     |
| conv2d_2 (Conv2D)             | 32     | 3x3         | (30,177,32)      | 1760  |
| max_pooling2d_2 MaxPooling2   | 2x2    |             | (15,88,32)       | 0     |
In this stage, we use transfer learning, a machine learning technique where information acquired through training in one type of problem is utilized for training in another. Support Vector Machine algorithm is one of the most efficient classification algorithm used in this research.

These are three major factors that have largely effect on CNN’s performance that requires to be examined. Which involve: Training Methodology, Input Image Size, and Training Strategies.

3.3.1 Training Methodology

Through this study, most of the experiments were performed utilizing (80%) manually selected samples for training, while the remaining (20%) for testing provided a specific set of sample data. The main steps of the suggested method of training are defined in algorithm 1.

**Algorithm 1**: CNN and the SVM algorithm

//**Input**: The input(normalization images)
//**Output**: The recognition accuracy
1. Divide the set of images into two sets: Training & Validation.
2. Use the training set to train the CNN configuration.
3. Test configuration of CNN using a set of validations.
4. Repeat steps 3 to 4 using N epochs.
5. Extract Features from the first hidden layers.
6. Take training labels via the training set.
7. Using the training features to train a multi_class(SVM) classifier.
8. Extract features via test set.
9. Using the trained classifier to assess each label for the check set.
10. Offer the known labels to check set.
11. Display the mean accuracy.

3.3.2 Input Image Size

The input image size is one of the active-parameters within CNN which has a major effect on neural network speed and precision. In this study, any effect of image size input is examined utilizing image sizes (64x360).

3.3.3 Training Strategies

1. Data Augmentation: it is widely acknowledged that DNNs want to be trained on a huge number of training samples to accomplish effective prediction so prevent overfitting [20]. Data augmentation is a
basic & widely utilized method to artificially increase the dataset via randomized crops, horizontal flipping, and intensity changes.

2. ADAM-Optimizer: is an adaptive learning rate approach discovered by Kingma and Ba, and has become one of the most common neural network step-size methods\[21\].

3. The ReLU activation function: to add non-linearity to the network, it is added to the top of the convolutional & fully connected layer. In comparison with other activation functions, like tangent and sigmoid \[22\], the ReLU $f(x)=\max(0,x)$ was discovered to be important for learning while using DNNs, specifically for CNNs. Furthermore, it results in training of the neural network many times faster than for other activation functions, without having a major change to generalization precision.

4. Weight Decay: is utilized within each learning operation like an extra term in determining the cost function & updating the weights. Now, the weight decay parameter is assigned to(0.0005) as in \[23\]. Table 2 shows the training parameters of the suggested system.

| Parameter          | Description                  |
|--------------------|------------------------------|
| Optimizer          | Adam (adaptive learning rate) |
| Maximum number of epoch | 200-300                      |
| MiniBatch Size     | 32                           |
| Learning rate      | 0.001                        |
| Weight decay       | 0.0005                       |

3.4 The Classification Stage
Classification is a mechanism of calculating the similarity between two iris templates which have been produced in the feature extraction process.

3.4.1 SoftMax Regression Classifier
A classifier performed within the fully connected portion of the model is a SoftMax regression classifier, a simplified version of the binary regression classifier in multi class classification purposes\[24\]. Each output vector variable relates to the approximate probability for every class label dependent upon the input feature. The probability vector may be described as:

$$ h_\theta(x_i) = \left[ \begin{array}{c} p(y_i = 1|x_i; \theta) \\ p(y_i = 2|x_i; \theta) \\ \vdots \\ p(y_i = k|x_i; \theta) \end{array} \right] = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x_i}} \left[ e^{\theta_1^T x_i} \\ e^{\theta_2^T x_i} \\ \vdots \\ e^{\theta_k^T x_i} \right] $$

Here, $\theta_1, \theta_2, \ldots, \theta_k$ are the variables to be randomly produced & learned via the back_propagation algorithm. K classes with n learning objects named $\{(x_1, y_1), \ldots, (x_n, y_k)\}$, whereas $x_i \in \mathbb{R}^m$ is the learning objects and $y_i \in \{1, \ldots, K\}$ is the label of class $x_i$, $h_\theta(x_i)$ is the strategy for predicting the probability vector of every label. The cost function utilized in the classifier SoftMax is referred to as the categorical cross_entropy loss function.
3.4.2 Support Vector Machine (SVM classification)
In this study, the obtained CNN feature vector is then fed to the classification algorithm. We utilize a common multi-class Support Vector Machine (SVM) [25] because of image classification efficiency and popularity. A short specification of multiclass SVM is as adopts: Given that a given training data set \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\). So it is required to split the set into two classes in which \(x_i \in \mathbb{R}^d\) is the feature vector while \(y_i \in \{-1, +1\}\) is the class of the label. The two classes and a hyperplane \(x + b = 0\) become linearly separable.

For a set of data with \(N\) classes, the multi-class SVM may be implemented. We must train \(N\) binary classifiers that may differentiate every class from all the remaining classes, thus choose the class that classifies each test sample against the maximum margin (one-vs-all).

4. Results and Discussion
4.1 Experimental Results
The implemented system has been tested on two popular iris datasets: the first provided by the Indian Institute of Technology Delhi (IITD) and the second provided by (Chinese Academy of Sciences - Institute of Automation (CASIA-Iris-V3 Interval). The implemented system gets an accuracy of 99% for both left & right IITD iris datasets while gets an accuracy of (94% and 93%) for the left and right CASIA-iris-V3 Interval datasets, respectively, after training as described in Figure 6. Both iris images are taken in these datasets under various situations of pupil expansion, eyelids & eyelashes occlusion, and a slight shadow. The specifications of these datasets are described in Table 3. The recognition accuracy of the selected iris image datasets shows in Table 4. The iris recognition experiments have been carried out on PC Laptop in which the properties are shown in Table 5 with the Jupyter notebook layout and the TensorFlow python libraries.

### Table 3
The characteristics of the used datasets

| Property                   | CASIA-Iris-V3 | IITD   |
|----------------------------|---------------|--------|
| Number of classes          | 120           | 224    |
| Samples per subject        | 6 right and 6 left | 5 right 5 left |
| Number of images           | 720 left 720 right | 1120 left 1120 right |
| Image size                 | (320 × 280) pixels | (320 × 240) pixels |
| Image format               | JPEG          | BMP    |

### Table 4
The recognition accuracy of the selected iris image datasets

| Dataset       | No. of epoch | Accuracy (%) Train CNN alone | Accuracy (%) CNN with transfer learning | Time (minutes) |
|---------------|--------------|------------------------------|----------------------------------------|---------------|
| IITD left     | 200          | 96.25%                       | 99%                                    | 21 minutes    |
| IITD right    | 200          | 96.26%                       | 99%                                    | 29 minutes    |
| interval left | 300          | 85.19%                       | 94%                                    | 32 minutes    |
| interval right| 300          | 88.89%                       | 93%                                    | 36 minutes    |

### Table 5
The properties of the computer utilized for implemented the system
| Operating system       | Windows 10 pro     |
|-----------------------|--------------------|
| Installed memory      | 8.00 GB            |
| Processor             | Intel® Core(TM) i5 6006U CPU 2.40GHz |
| System type           | 64-bit Operating system, x64-based processor |

![Training and Validation accuracy of (a) IITD dataset (b) CASIA interval dataset)](image)

**Figure 6** (Training and validation accuracy of (a) IITD dataset (b) CASIA interval dataset) before transfer earning.

### 4.2 Feature Visualization

Feature Extracting or visualizing is utilized to view the image features that are generated from various hidden convolutional layers in the design network [26]. It also offers a summary of how the input has been analyzed in the different filters that the network has trained. Figure 7 displays the visualizes maps of features of all layers utilizing in the implemented design with an input picture (normalization image). The resulting figure illustrates feature maps for each layer:

1. The first layer feature maps (Conv2d_1) retain more of the image details and generally operate as edge detectors.
2. When we move deeper through the network, the feature maps are generated least like the main image and quite like an abstract description of it. As you can show in Conv2d_3, but it appears...
unrecognizable afterward. This is because deeper feature maps represent ideas of a high level and feature maps of a lower level identify basic edges and shapes.

3. When we move deeper the filters recognize fewer features. This explains appropriate since the first-layer filters recognize basic shapes, and each image involves these. But when we move deeper we begin to search for more complicated stuff, and do not show in each image (e.g., Conv2d_4 to Dence_3).

![Diagram of feature visualization](image)

**Figure 7.** Features Visualization (Extraction)

### 5. Conclusions

In this study, a powerful and efficient multimodal biometric system has been implemented to determine the person’s identity by building a deep learning-dependent approach of the same person’s right & left irises. The current system includes segmentation (to identify iris area using CHT algorithm), normalization image (utilizing a rubber sheet model), feature extraction, and classification stage depending on the Convolutional Neural Network (CNN) with transfer learning to generate special features supplied to a multi_class SVM algorithm to achieve iris recognition. Typically, the creation and execution of a multimodal biometric framework is a difficult challenge, and a variety of aspects that have a significant effect on total performance want to be addressed, involving cost, accuracy, and biometric features.

In the future, this design could be modified to utilize colored iris images, which will help improve the accuracy of identifying iris boundaries. The performance of the current IrisConvNet model could be examined in solving the challenge of heterogeneous iris detection.

### References

[1] Akshay A. Pawle and Vrushsen P. Pawar, “A Study of Different Biometric Authentication Techniques in Cloud Computing,” *Int. J. Eng. Res.*, vol. V6, no. 05, pp. 944–951, 2017.

[2] S. Punnoose and J. S. J. Kumar, “Iris Recognition for Security & Safety of Automobiles,” *Int. J. Innov. Sci. Eng. Technol.*, vol. 2, no. 4, pp. 961–966, 2015.

[3] R. Hentati, M. Hentati, and M. Abid, “Development a New Algorithm for Iris Biometric Recognition,” *Int. J. Comput. Commun. Eng.*, vol. 1, no. 3, pp. 283–286, 2012.

[4] A. Das and R. Parekh, “Iris Recognition using a Scalar based Template in Eigen-space,” *Int. J. Comput. Sci. Telecommun.*, vol. 3, no. 5, pp. 3–8, 2012.

[5] John Daugman,” How iris recognition works”, The essential guide to image processing, pp.715-739, 2009.

[6] Du Y, Bourlai T, Dawson J (2016) Automated classification of mislabeled near-infrared left and right iris images using convolutional neural networks. In: 2016 IEEE 8th international conference on biometrics theory, applications and systems, pp 1–6

[7] Tobji R, Di W, and Ayoub N (2019) FMnet “Iris segmentation and recognition by using fully and multi-scale CNN for biometric security”. In Applied Sciences (Switzerland), pp. 1–17. 2016.
[8] K. B. Raja, R. Raghavendra, S. Venkatesh, and C. Busch, “Multi-patch deep sparse histograms for iris recognition in visible spectrum using collaborative subspace for robust verification,” Pattern Recognit. Lett., vol. 91, pp. 27–36, 2017.

[9] M. A. M. Abdullah, R. R. Al-Nima, S. S. Dlay, W. L. Woo, and J. A. Chambers, “Cross-Spectral Iris Matching for Surveillance Applications,” pp. 105–125, 2018.

[10] P. R. Nalla and A. Kumar, “Toward more accurate iris recognition using cross-spectral matching,” IEEE Trans. Image Process., vol. 26, no. 1, pp. 208–221, 2017.

[11] M. G Alaslni and L. A. Elrefaei, “Transfer Learning with Convolutional Neural Networks for IRIS Recognition,” Int. J. Artif. Intell. Appl., vol. 10, no. 5, pp. 49–66, 2019.

[12] R. Soliman, E. M. El-rabaie, N. Ismail, and F. E. A. El-samie, “Noisy Iris Recognition Based on Deep Neural Network Noisy Iris Recognition Based on Deep Neural Network,” no. July, 2020.

[13] B. Hassan, R. Ahmed, T. Hassan, and N. Werghi, “SIP-SegNet: A Deep Convolutional Encoder-Decoder Network for Joint Semantic Segmentation and Extraction of Sclera, Iris and Pupil based on Periocular Region Suppression.”

[14] Lin, G and Shen, W (2018) “Research on convolutional neural network based on improved Relu piecewise activation function”, Procedia Comput. Sci., vol. 131, pp. 977–984.

[15] Wang, K. and Kumar .A.: “ Cross-spectral iris recognition using CNN and supervised discrete hashing”. In Pattern Recognit, pp. 85–98, (2019).https://doi.org/10.1016/j.patcog.2018.08.010.

[16] Basha .S. H. S, Dubey. S. R., Pulabaigari .V, and Mukherjee .S (2020) “Impact of fully connected layers on performance of convolutional neural networks for image classification”. In Neurocomputing, pp. 112–119. https://doi: 10.1016/j.neucom.2019.10.008.

[17] A. S. Al-Waisy, R. Qahwaji, S. Ipson, and S. Al-Fahdawi, “A fast and accurate iris localization technique for healthcare security system,” Proc. - 15th IEEE Int. Conf. Comput. Inf. Technol. CIT 2015, 14th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2015, 13th IEEE Int. Conf. Dependable, Auton. Se., pp. 1028–1034, 2015.

[18] Daugman. J (2007) “New methods in iris recognition”. In IEEE Transactions on System Man and Cybernetics Part B, pp. 1167–75. https://doi: 10.1109/TSMCB.2007.903540.

[19] Choraś, R. S, Iris recognition (2009) “Advances in Intelligent and Soft Computing”, pp. 593–600.

[20] Al-Waisy .A. S, Qahwaji, R. Ipson. S., and Al-Fahdawi. S (2018) “A multimodal deep learning framework using local feature representations for face recognition”. In Machine Vision and Applications pp. 35–54. https://doi.org/10.1007/s00138-017-0870-2.

[21] Madhusudhan .M. V, Udayarani .V, and Hegde. C (2020) “Finger Vein Based Authentication using Deep Learning Techniques”, Int. J. Recent Technol. Eng., pp. 5403–5408.

[22] Gonzalez. T. F (2007) Handbook of approximation algorithms and metaheuristics, Handb. Approx. Algorithms Metaheuristics, pp. 1–1432.https://doi.org/10.1201/9781420010749.

[23] Turchenko V and Luczak A.: “Creation of a deep convolutional auto-encoder in Caffe”. Proc. IEEE 9th Int. Conf. Intell. Data Acquis. Adv. Comput. Syst. Technol, pp. 651–659, (2017). https://doi:10.1109/DAACS.2017.8095172.

[24] Wani, a,Ahmad,F,Saduf .b et al.: Advanced Applied Deep Learning.https://doi.org/10.1007/978-981-13-6794-6.

[25] Kumar, Geethu S and Devi, C Jyothirmati (2014) “A MultimodalSVM Approach for Fused Biometric Recognition”. In International Journal of Computer Science and Information Technologies:3327-3330.

[26] Muthana J. Mohammed, Emad A. Mohammed, Mohammed S. Jarjees. "Recognition of multfont English electronic prescribing based on convolution neural network algorithm", Bio-Algorithms and Med-Systems, 2020