Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Distant Iris Recognition Through Machine Learning Models with Deep Features Transfer for Human Identification

Md. Azizur Rahman[1*], Lasker Ershad Ali[2], Saikat Kumar Mistry[3], Arnab Mukherjee[4], Md. Zahidul Islam[5]

[1-5]Mathematics Discipline, Khulna University, Khulna, Bangladesh

[4] Department of Quantitative Sciences (Mathematics), International University of Business Agriculture and Technology, Dhaka, Bangladesh.

[5]School of Engineering, Design and Built Environment, Western Sydney University, Australia.

E-mail: [*1]mdazizur@math.ku.ac.bd

Abstract

Human identification through biometrics has become increasingly popular due to its reliable authentication in automated high-security surveillance systems. Several biometric models based on fingerprint, face detection, and iris recognition have been designed and developed for human identification. Among these biometrics, iris recognition, especially distance-based recognition, remains a significant challenge due to its small imaging target. In this paper, we propose a distant iris-based human identification framework employing a deep extracted feature transfer with machine learning (ML) models. In the first stage, we customized the traditional convolutional neural network (CNN) model and utilized three pre-trained models VGG16, VGG19, and ResNet50 for the extraction of deep features from normalized iris images. Later, we fed these deep features extraction into nine ML models for iris image classification. The proposed framework is validated via several experiments using the CASIA-V4 iris dataset. Experimental results show that the softmax classifier with our customized CNN model outperforms the considered pre-trained deep learning models, achieving top scores in accuracy (93.40%), precision (94.31%), recall (93.40%), F1-score (93.25%), and Cohen's kappa (93.34%). This customized CNN model with a softmax also demonstrates competitive performance when compared with other distance-based iris recognition models.

Keywords: Biometrics Recognition, Iris Features, Transfer Learning, Convolutional Neural Networks, Machine Learning Models, ROC Curves.

1. Introduction

The biometric technique is very important to ensure the safety of the public, law enforcement, and healthcare systems. Conventional biometrics are limited to knowledge, token, SMS, and password-based authentication. These authentication systems are laborious, time-consuming, inaccurate, and subject to human biases [1-3]. Due to technological advancements, the biometric technique has overcome these limitations by enabling fingerprint, eye recognition, face detection, and iris recognition [4-6]. Among these biometric techniques, iris

*** Corresponding author**

DOI: https://doi.org/10.36548/jiip.2025.3.14
Received: 17.07.2025, received in revised form: 19.08.2025, accepted: 04.09.2025, published: 16.09.2025

recognition is the most reliable. It is an internal organ of a human and has a unique feature. The other biometrics are replaceable, but the iris is irreplaceable because the irises of the two eyes of a human are not the same [7]. So, no one can share their iris with others.

Iris patterns exhibit a remarkable level of uniqueness and complexity, retaining their distinctiveness from around age three throughout a person's lifetime, as supported by clinical findings. As a result, research on iris biometrics is gaining great interest in the scientific community day by day. However, the collection of a quality iris image can be challenging because it may be affected by lighting, illumination, and the distance between a camera and an object. The distance is a critical factor for capturing an iris image from a human face. Long-distance iris images under less-controlled conditions may affect the performance metric of any machine learning (ML) as well as deep learning models, compared to short-distance iris images. Therefore, this study proposes a distant iris recognition framework through deep extracted features transfer with ML models in the case of class-imbalanced data. In class-imbalanced datasets, instances belonging to minority classes are typically sparse and unevenly distributed, while the majority classes tend to dominate the feature space. It is crucial to identify a classification model that will be capable of handling the class imbalance problems, minimizing the impact of various noise sources, and improving overall classification accuracy.

This study aims to reliably address current limitations and develop a supervised deep learning algorithm capable of performing distant iris recognition for human identification. To account for the textural complexity of the iris, this paper introduces deep transfer learning, which is designed to strategically capture robust and hierarchical features directly from normalized iris images. The VGG16, VGG19, ResNet50, and a 32-layer customized CNN model are employed to extract deep features from multiple layers of the architecture. After that, these deep extracted features are transferred to several ML techniques to search for an effective classification model. Specifically, random forest (RF), linear discriminant analysis (LDA), K-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), Naïve Bayes, multi-layer perceptron (MLP), extra tree (ET), and softmax classifiers are adopted to analyze classification performance using various benchmark statistical measures. Finally, we perform an optimal search to combine deep transfer features with ML models to improve recognition performance for human identification from distantly acquired iris images. The major contributions are summarized as follows:

- This paper integrates deep feature extraction and transfer techniques with ML models for distant iris recognition. The combinations exhibit acceptable performance in iris recognition for human identification.
- A customized 32-layer CNN model, along with VGG16, VGG19, and ResNet50 deep learning models, is implemented to extract deep features from multiple layers of the architecture. These feature extraction frameworks transfer the extracted deep features to different conventional ML models to find an effective classification model.
- The study investigates the classification performance by employing different combinations of each deep feature model together with nine ML algorithms, such as RF, LDA, KNN, SVM, Naive Bayes, DT, MLP, ET, and softmax for iris image classification. Among these combinations, our customized CNN with a softmax framework generates the highest accuracy, precision, recall, F1-Score, and Cohen's Kappa.
- Finally, the proposed iris recognition approach achieves comparable recognition performance to existing distant iris recognition approaches.

The structure of the remaining sections of this paper is as follows: a brief overview of recent works related to iris classification is given in Section 2. The methodology of the iris classification framework is described in detail in Section 3. The experimental setup with evaluation protocols and results analysis through visual representations are presented in Section 4. The conclusions are presented in Section 5.

2. Related Works

Recent years have seen substantial progress in iris recognition systems, driven by innovations in ML algorithms and deep neural network architectures. This section highlights recent advancements in iris recognition systems.

In [7], Daugman presented the prototype system for automatic human identification. Since then, this biometric has drawn attention from researchers across academia and industry due to its wide-ranging applications. Apart from Daugman, most authors have used wavelet-based filters and distance-based similarity matching to improve biometric systems, with the most notable contributions made by Suvarchala et al. [8], Ahmadi et al. [9], Kaewphaluk et al. [10], Chirchi et al. [11], and Arnab et al. [12]. The limitation of their studies is the reliance on high-resolution cameras for iris image capture. Their accuracy significantly decreases in less controlled environments due to various challenges such as low resolution, off-axis capture angles, blur caused by motion, occlusion, and specular reflections. These issues commonly occur in real-time settings and degrade further processing. To overcome these challenges, the studies emphasize deep transfer learning, which can extract robust, hierarchical features from raw iris images, enabling effective handling of noise, occlusions, and varying imaging conditions. Traditional ML methods struggle with complex, noisy iris patterns due to their reliance on handcrafted features.

Al-Waisy et al. developed the IrisConvNet framework using a CNN architecture that integrates features from both irises to enhance recognition accuracy [4]. Tianming et al. proposed an iris recognition method leveraging Capsule Networks to better capture spatial hierarchies and part-whole relationships in iris textures [13]. Arsalan et al. proposed a deep learning framework for iris segmentation specifically designed for visible light conditions, aiming to improve accuracy in challenging environments characterized by noise and reflections [14]. Ren et al. proposed a multiscale dynamic graph-based method that enhances biometric recognition performance in the presence of occlusions [15]. They employed CNNs to extract deep features, which are subsequently reorganized into a feature graph based on specific subregions. In the feature graph, each node represents a distinct local area of the given iris image, while the edges capture the correlation relationships among non-occluded areas.

Ali et al. developed a hybrid feature extraction method combining Log-Gabor wavelets and Contourlet transform, achieving up to 95.93% accuracy on the CASIA-v4 dataset when fused with CNN and GLAC features [16]. Later, the same authors demonstrated high recognition accuracy by integrating diverse feature descriptors and classifiers, effectively addressing challenges posed by distant iris imaging [17]. However, a specific drawback is the system's sensitivity to variations in lighting conditions and image quality, which can adversely affect performance in uncontrolled environments. Kaur et al. addressed iris recognition under unconstrained environmental conditions by employing polar harmonic transform and Zernike Moments to effectively manage variations caused by rotation, noise, and illumination changes [18, 19]. In [20], Meryem et al. developed a framework that integrates dual CNNs with texture-

based features through decision-level fusion to enhance performance. However, their method involves concatenating face and iris images to improve recognition accuracy.

In 2022, El-Latif et al. integrated an edge-detection strategy with the Hough transform to identify iris boundaries, edges, and semicircular structures, while employing CNNs to capture spatial hierarchies and localized feature patterns [21]. Recently, El-Sofany et al. reported a slight enhancement over the prior hybrid model by El-Latif et al. in [21] with similar settings [22]. Notably, the approach omits iris normalization entirely, which poses challenges for maintaining consistent performance. Both hybrid systems exhibit sensitivity to noise, low contrast, and partial occlusions from eyelashes or eyelids, and they fail to accurately segment iris textures in cases of non-circular iris shapes. Lee et al. were the first to introduce a cGAN approach for augmenting iris datasets, aiming to enhance recognition accuracy and mitigate overfitting caused by training on limited data [23]. However, the technique's effectiveness diminishes when incorporating the periocular region, and challenges such as training instability and mode collapse inherent to GANs may affect the diversity and reliability of the generated data.

Ribeiro et al. conducted a study on CNN-based techniques for iris recognition performance. Their method seeks to reconstruct fine-grained texture details by generating photo-realistic images, leveraging optimization processes guided by both the CNN architecture and its training strategy [24]. The study finds that while CNNs enhance visual quality, improved photo-realism does not inherently lead to superior recognition outcomes. A key drawback is the potential introduction of artifacts that may distort critical iris features, negatively affecting recognition reliability.

As discussed above, distant iris classification remains challenging due to factors such as image processing, appropriate feature extraction, and reliable classifier performance. To address these issues, this paper emphasizes a transfer learning-based iris recognition system that specifically extends the CNN with a softmax-based deep learning classifier. This work is motivated by the ability of CNNs to autonomously learn and retrieve hierarchical patterns directly from low-quality images, enabling effective recognition of complex patterns with minimal manual feature engineering.

3. Methodology

This section presents the deep transfer learning-based iris recognition framework, which adapts the CNN architecture and integrates a machine learning ML classifier within the fully connected layer. This iris recognition framework comprises a training phase and a testing phase. In the training phase, the collected iris images undergo preprocessing, including contrast enhancement, iris segmentation, and iris normalization. Following normalization, the iris textures are passed through several deep learning models to automatically capture the local iris patterns and encode them into feature vectors. Finally, the extracted features are divided into training and validation subsets, which are then used to train nine machine learning (ML) classifiers for building the classification models. Using the training subset, we train the ML classifiers and evaluate model performance based on an independent testing subset during the testing phase. To provide a clearer understanding, a schematic diagram of our proposed recognition framework is displayed in Fig. 1.

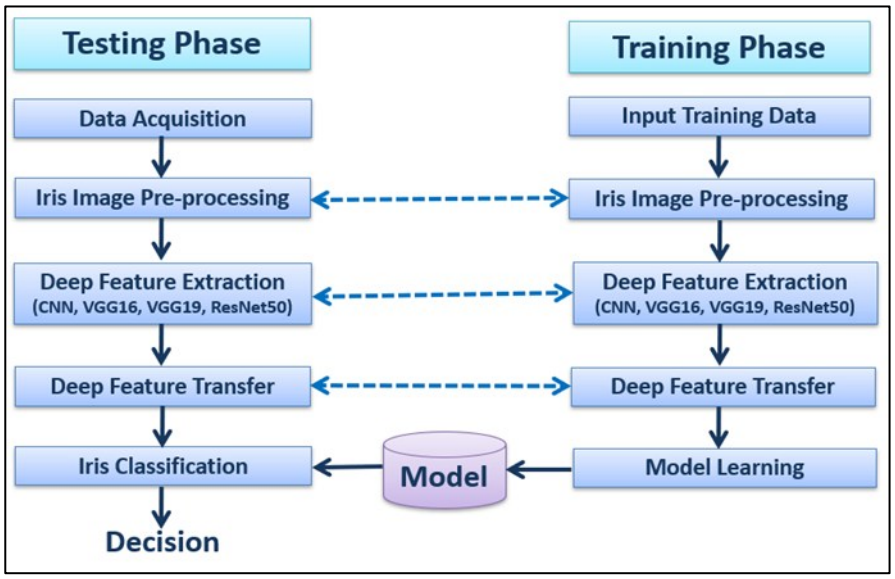

Figure 1. The Schematic Diagram of the Iris Recognition Framework

Actually, in the proposed iris recognition framework, we employ four pre-trained deep feature extraction techniques- CNN, VGG16, VGG19, and ResNet50. Among these, the CNN model is customized before the fully connected layer (FC), and key hyperparameters are tuned to minimize the validation loss. The deep extracted feature spaces are then transferred to nine classical machine learning models (RF, LDA, KNN, SVM, DT, Naïve Bayes, MLP, ET, and softmax) for model training. The complete system workflow, from input acquisition to final classification using the customized CNN features, is illustrated in Fig. 2. The diagram highlights all major components and their interactions, including data preprocessing, CNN feature extraction, train–test data split, optimal hyperparameter selection, model training, and classification. Each of these stages is briefly described in the following subsections.

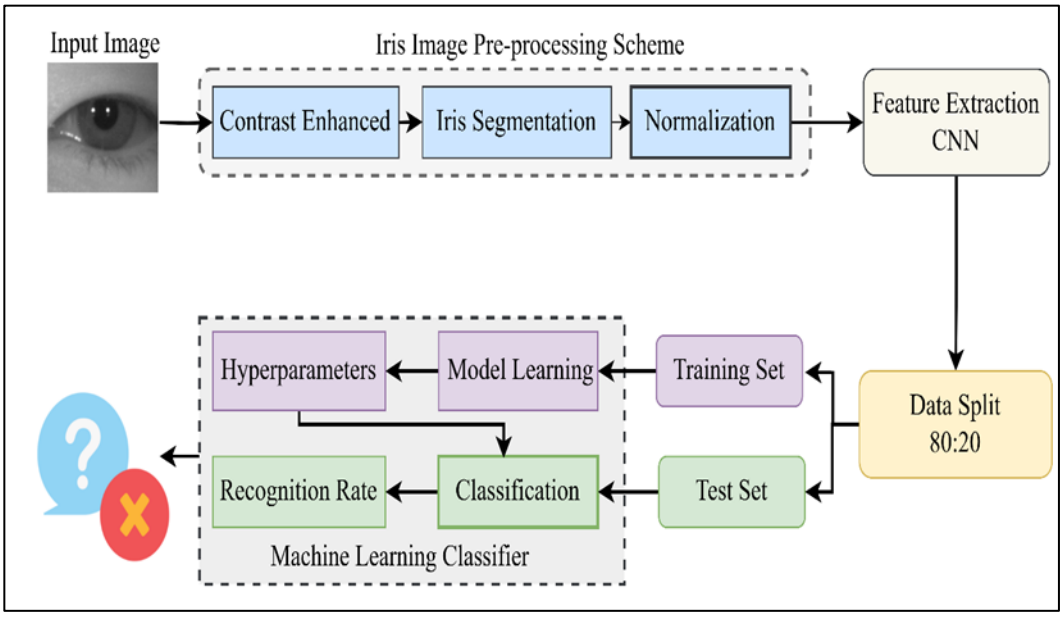

Figure 2. The Complete Workflow of the Proposed Iris Recognition System

3.1 Image Pre-processing

In the proposed approach, we have accomplished several image pre-processing tasks, namely, contrast enhancement, iris segmentation, and normalization, with the help of standard techniques. Indeed, the quality of the acquired eye image is degraded due to being captured from a long distance and under uncontrolled lighting conditions. Therefore, it is necessary to enhance the quality of degraded images for better segmentation of the iris from eye images. For this reason, we have used a well-known Single-Scale Retinex (SSR) algorithm for the contrast enhancement of eye images [25]. The SSR algorithm can be modeled by the following equation

$$R_{lm}(p,q) = \log\left[\frac{Im(p,q)}{G_\tau(p,q) * Im(p,q)}\right] \quad (1)$$

where $Im(p,q)$ indicates a grayscale eye image, and $G_\tau(p,q)$ a Gaussian kernel expressed as $e^{-\frac{(p^2+q^2)}{\tau^2}}$. The convolution is represented by "*", with the standard deviation parameter $\tau = 1.5$.

3.2 Iris segmentation

The enhanced eye images include eyelids, eye shadow, pupils, and sclera, which complicate the process of accurately identifying people through unique iris patterns. It is essential to segment the iris pattern from the enhanced eye images by removing those non-unique patterns. Many image segmentation techniques are available in the literature. In this approach, we follow the best-performing graph-theoretical random walker technique for coarse iris segmentation mentioned in [26].

3.3 Iris normalization

Variations in the scale of segmented iris images occur due to pupil dilation of the eye, variations in lighting, and image capture from a distance. In this paper, we consider Dogman's rubber sheet model to eliminate the differences in the scale of the iris image and bring it within the same range [27]. This method normalizes segmented irises by converting Cartesian coordinates (p,q) to polar coordinates (r,θ), where $r \in [0,1]$ and $\theta \in [0,2\pi]$. This transformation eliminates dimensional disparities and corrects texture distortions for accurate matching. The mapping from the original iris image to the normalized form can be expressed in Eq. (2) and Eq. (3).

$$Im(p(r,\theta), q(r,\theta)) \rightarrow Im(r,\theta)$$

$$p(r,\theta) = (1 - r)p_p u(\theta) r p_b p(\theta) \quad (2)$$

$$q(r,\theta) = (1 - r)q_p u(\theta) r q_b p(\theta) \quad (3)$$

where $Im(p,q)$ is the intensity at the original iris image coordinates (p,q). Also, $p(r,\theta)$ & $q(r,\theta)$ are the coordinates along θ. The interpolation for normalization is usually linear between the pupil coordinates $(p_p u(\theta), q_p u(\theta))$ and iris boundaries $(p_b p(\theta), q_b p(\theta))$ for each θ. Figure 3 sequentially illustrates the results of noise removal, iris segmentation, and normalization from the eye images.

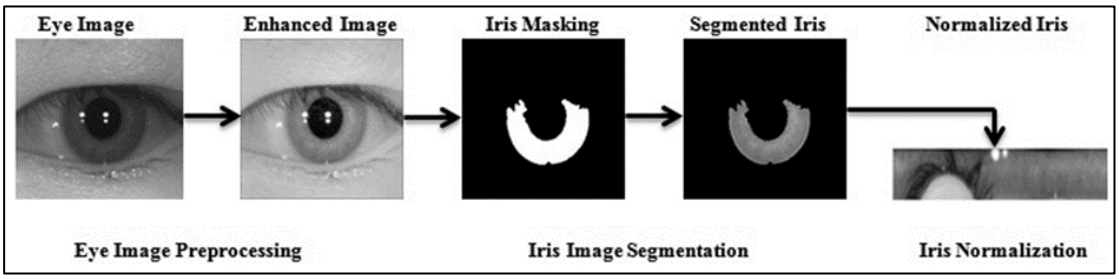

Figure 3. The Framework Outlining the Image Preprocessing Schema [16]

3.4 Customized CNN for Iris Feature Extraction

In recent years, numerous deep learning architectures have been designed and developed to capture informative features from raw input data through multiple hidden layers and to generate class probabilities for accurate target prediction. Among these deep learning models, CNN-based methods have consistently demonstrated superior performance across diverse computer vision tasks [28–31].

In this study, we develop a customized CNN model for multi-class iris image classification (128 classes) aimed at human identification. The proposed CNN is designed to balance feature extraction, enabling hierarchical feature learning from low-level iris patterns. This customized architecture is chosen over deeper pre-trained models like VGG or ResNet to remain computationally feasible for the image dataset. The iris images are preprocessed by resizing them to 224×224 pixels, converting them to grayscale, normalizing the pixel values, and applying one-hot encoding to the labels. The customized model consists of 32 layers with three convolutional blocks (128, 64, and 32 filters). The network is designed with an input layer, two convolutional layers, pooling layers, batch normalization layers, rectified linear unit (ReLU) layers, an FC layer, and a classification layer.

The input layer receives preprocessed iris images, maintaining their original spatial dimensions and channel depth. The convolutional layers employ 16 learnable filters of size 3×3 with a padding of 1, preserving the spatial hierarchies during feature extraction. Pooling layers reduce the spatial dimensions by selecting the maximum feature values, thereby lowering computational costs. Additionally, a batch normalization layer is applied after each convolutional layer to stabilize and accelerate training by normalizing the output activations. The network achieves non-linearity and enables the learning of complex patterns by incorporating the ReLU function, as defined in Eq. (4).

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{4}$$

The flattened output is then passed through a fully connected (FC) layer, which generates a dense representation by connecting all neurons from the previous layers. A softmax activation function converts the FC outputs into class probabilities, producing positive values that sum to one and enabling clear classification. Finally, the classification layer uses these probabilities to predict the target class and compute the loss via cross-entropy. During backpropagation, the cross-entropy loss quantifies the deviation between the predicted outputs and the true classes.

The trainable multistage architectural customized CNN model starts with 128 filters, and then decreases to 32 filters in deeper layers. After flattening, the network transitions to fully connected layers with 256, 128, and 64 neurons before the final 128-neuron classification layer. The largest number of parameters is in the first dense layer (~6 million), while convolutional layers have significantly fewer parameters. The model contains a total of 6,312,576 parameters, including 6,310,784 trainable and 1,792 non-trainable. The model is trained with a batch size of 32 over 64 epochs, using an 80:20 random split between training and validation data. The hyperparameters are optimized with the Adam optimizer, configured with a learning rate of 0.001, β_1=0.9 (first moment decay rate), β_2 = 0.999 (second moment decay rate), and $\varepsilon = 10^{-7}$ (numerical stability term).

3.5 Machine Learning Models

In our framework for distant iris recognition, the extracted feature vectors, obtained from the deep learning models (customized CNN, VGG16, VGG19 and ResNet50), are transferred to train nine classical machine learning models: Softmax classifiers, K-nearest neighbor (KNN), support vector machine (SVM), multi-layer perceptron (MLP), Naïve Bayes, random forest (RF), decision tree (DT), extra trese (ET), and linear discriminant analysis (LDA). These classifiers are evaluated for their ability to achieve accurate and reliable recognition performance. The following subsections provide a brief overview of these ML models.

3.5.1 Softmax Classifier

The softmax classifier uses cross-entropy loss, derived from the softmax function, to optimize the training model. The function converts raw class scores into normalized probabilities, enabling effective loss computation. By mapping scores to positive probabilities, it ensures stable and interpretable outputs. Cross-entropy loss then measures prediction accuracy by comparing these probabilities against true labels, penalizing errors, and refining model confidence. The gradient of the loss function is symmetric with respect to indices (p,q) [32]:

$$\frac{\partial}{\partial j_q}\lambda(p,q) = \lambda(p,q)\left(\eta_{jp} - \lambda(p,q)\right) \tag{5}$$

3.5.2 K-Nearest Neighbor (KNN)

As a distance-based supervised algorithm, KNN predicts the class of a query instance by aggregating the majority class among its K nearest neighbors [1]. The performance of the KNN algorithm can be severely influenced by the choice of K neighbors and distance metric, which relies on feature properties.

3.5.3 Support Vector Machine (SVM)

SVM predicts class labels by finding the optimal separating hyperplane. For a linear case, the hyperplane is defined as follows:

$$g(X) = W^T X + B \quad (6)$$

where W is the weight vector, X represents the feature vector, and B denotes the hyperplane's bias. The equation of a hyperplane is considered as $W^T X + B = 0$, where $\frac{W}{\|W\|}$ is perpendicular to the extricating hyperplane and its distance is always unity. The margin width $d = \frac{2}{\|W\|}$ is maximized between parallel hyperplanes $W^T X + B = \pm 1$. Support vectors are the samples closest to these boundaries that determine the optimal margin [34, 35].

3.5.4 Multi-Layer Perceptron (MLP)

MLP is a neural network with hidden layers that learns nonlinear input-output mappings [35]. Node weights w_{pq} update via:

$$\Delta w_{pq}(\alpha) = -\alpha \frac{\partial \phi(\alpha)}{\partial S_q(\alpha)} x_p(\alpha) \quad (7)$$

where $\phi(\alpha) = \frac{1}{2}\sum_q d_q^2(\alpha)$, $x_p(\alpha)$ is the output of the neuron p from the previous layer (input to the current connection) at iteration α and $S_q(\alpha) = \sum w_{pq}(\alpha)\, x_p(\alpha)$. The activation derivative links output-layer weights to hidden-layer weights.

3.5.5 Naive Bayes Classifier (NB)

Naive Bayes classifies data using Bayes' theorem with a conditional independence assumption between features. Given features $u_1, u_2, \ldots, u_n$ and class v, the posterior probability is computed by the following equations [35]:

$$P(v|u_1, u_2, \ldots, u_n) = \frac{P(v)\prod_{i=1}^{n} P(u_i|v)}{P(u_1, u_2, \ldots, u_n)} \quad (8)$$

The predicted class $\hat{y}$ is the maximum a posteriori (mAP) estimate:

$$\hat{y} = arg \max_v P(v) \prod_{i=1}^{n} P(u_i|v) \quad (9)$$

3.5.6 Decision Tree (DT)

Decision Trees (DTs) are supervised models for classification and regression that build tree structures by recursively splitting data using optimal feature thresholds. For classification, splits minimize Gini impurity, $G = 1 - \sum_{i=1}^{C} p_i^2$, where, p_i^2 denotes the squared probability of class i in a node with C total classes. Trees grow until meeting stopping conditions using max depth [35].

3.5.7 Extra Trees Classifier (ET)

Extra Trees (ET) is an ensemble method using uncorrelated decision trees, similar to Random Forest but with randomized splits. For each node, subsets $K_p^{left}(\alpha)$ (features ≤

threshold T_p) and K_p^{right} (α) (remaining features) are created from training vectors u_h and labels v, the parameter α minimizes the impurity [35].

3.5.8 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) maximizes class separation by optimizing the ratio of between-class scatter (S_B) to within-class scatter (S_w). For two classes Y_1 and Y_2, Fisher's criterion finds the projection w that maximizes $w = \frac{w^T s_B w}{w^T s_w w}$, where $s_B = (m_1 - m_2)(m_1 - m_2)^T$ and $s_w = \sum_{i=1} \sum_{y \in Y_i} (y - m_i)(y - m_i)^T$ quantify the scatter both between and within classes with projection means m_i

3.5.9 Random Forest (RF)

Random Forest (RF) is an ensemble of decision trees trained on random data subsets. For prediction, each tree votes for class n, and the majority wins. The vote is counted by Eq. (10).

$$u_n = \sum_{t=1}^{n} I(\hat{x}_t = n) \tag{10}$$

where $\hat{x}_t$ denotes the t-th prediction of trees for a certain instance, and $I(\cdot)$ is 1 if the t-th tree predicts class n [35]. The final output is $\hat{x} = arg \max_n u_n$.

4. Results and Discussion

The effectiveness of deep transfer learning is evaluated and discussed in this section with respect to distant iris recognition for human identification. Quantitative metrics derived from confusion matrices and ROC curves are used to compare pre-trained deep learning models with fine-tuning on iris datasets. The analysis reveals how transfer learning mitigates data scarcity by leveraging learned features from large-scale datasets, reducing computational costs while maintaining high precision. The following subsections depict the overall results and discussion of the proposed method for this iris recognition framework for human identification.

4.1 Database and Experimental Protocol

For experimental evaluation, we have employed the CASIA-v4 distance database, a large dataset of distantly acquired facial images provided by CASIA [36]. The facial images were captured using a near-infrared camera positioned approximately 3 meters from the subject in less controlled environments. The CASIA-v4 database contains a total of 2,567 facial images collected from 142 subjects. Both eyes were separated from the full-face images, yielding 5,134 eye images used for subject identification in the experimentation. Only regular images are available for the first 14 subjects in this database. To ensure reliable performance, we exclude these 14 subjects from our experiments. From the remaining 128 subjects, we randomly select 80% of the iris images to train the classification models. Similarly, the remaining 20% were designated for model testing.

Google Colab, a cloud-hosted Jupyter Notebook platform that provides free computational resources, was used for all experiments. By choosing the NVIDIA Tesla T4 GPU from the runtime settings, a GPU runtime was enabled to speed up model training and evaluation. The TensorFlow and Keras frameworks, as well as NumPy, Matplotlib, Pandas, Seaborn, and Scikit-learn, were used to implement deep learning models.

Large numeric range features may disproportionately affect learning without normalization, leading to biased or unstable model performance. According to the following formula, min-max normalization thus resolves this problem without causing the deep features V to distort notable feature differences:

$$V = (\vartheta_j - \vartheta_{min})/(\vartheta_{max} - \vartheta_{min}) \quad (11)$$

where ϑ_j represents the jth feature value between ϑ_{max} and ϑ_{min}. These low-dimensional, labeled, and scaled features facilitate effective training of the ML classifiers.

In order to achieve optimal performance, different hyperparameter settings for classification models are tested during experimentation. We use the Adam optimizer to optimize the important hyperparameters for nine machine learning classification models from the various settings; the ideal hyperparameter values are shown in Table 1.

Table 1. Machine Learning Models and Key Optimal Hyperparameters

ML Model	**Key Optimal Hyperparameters**
RF	n_estimators value is 100, gini is set as the criterion, max_features is 'sqrt' with 42 random_state
LDA	The svd is used as the solver, where the tolerance level is 0.0001
KNN	K=5, 'uniform' weights, 'minkowski' as the distance metric
SVM	Kernel function is 'linear', C=1.0, gamma='scale', random_state value=42
Naïve Bayes	var_smoothing=1e-9
DT	The 'gini' is used as the criterion, splitter='best', and the value of random_state is 42
MLP	hidden_layer_sizes=256, max_iter=500, random_state=42, activation='relu', solver='adam'
ET	n_estimators=100, criterion='gini', max_features='sqrt', random_state=42
Softmax	activation='softmax', input_units=128, output_units=128

4.2 Evaluation Metrics

In the experimental evaluation, multiple metrics are used to measure the accuracy and reliability of the classifiers. Since overall accuracy alone may be insufficient, particularly in cases of class imbalance, the evaluation framework also incorporates additional measures

derived from the confusion matrix to assess recognition performance for each class. The confusion matrix is presented in Table 2.

Table 2. The Confusion Matrix

	Predicted positive (p)	Predicted negative (n)
Actual positive (p)	True positives (tp)	False negatives (fn)
Actual negative (n)	False positives (fp)	True negatives(tn)

From the confusion matrix, the average precision, recall, F_1-score, and Cohen's Kappa (κ) for this multi-class iris classification system are defined by the following equations:

$$\text{Average Precision} = \frac{1}{C}\sum_{j=1}^{C}\frac{tp_j}{tp_j+fp_j} \tag{12}$$

$$\text{Average Recall} = \frac{1}{C}\sum_{j=1}^{C}\frac{tp_j}{tp_j+fn_j} \tag{13}$$

$$F_1 - \text{measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{14}$$

$$\text{Cohen's Kappa } \kappa = \frac{1}{C}\sum_{j=1}^{C}\frac{2(tp_j \times tn_j - fp_j \times fn_j)}{(tp_j+fp_j)\times(fp_j+tn_j)+(tp_j+fn_j)\times(fn_j+tn_j)} \tag{15}$$

where C is the total number of classes, indexed by j. The confusion matrix provides the basis for performance measures by comparing predicted and actual classifications for each class [39].

4.3 Experimental Results and Performance Analysis

In this subsection, the obtained results from the various deep extracted features-based ML models are summarized and analyzed sequentially. The results of the VGG19 feature descriptor with nine classification models are presented in Table 3.

Table 3. The Performance of Classification using the VGG19 Feature Descriptor

Classifiers	**Accuracy**	**Precision**	**Recall**	**F_1-Score**	**Cohen's Kappa**
Random Forest	0.6812	0.7156	0.6812	0.6592	0.6783
LDA	0.7388	0.7854	0.7388	0.7288	0.7364
KNN	0.3764	0.6028	0.3764	0.4062	0.3708
SVM	0.8118	0.8452	0.8118	0.8083	0.8101
Decision Tree	0.2121	0.2111	0.2111	0.2030	0.2053
Naïve Bayes	0.4424	0.4957	0.4424	0.4370	0.4375
MLP	0.7669	0.7938	0.7669	0.7605	0.7648
Extra Trees	0.6938	0.7160	0.6938	0.6756	0.6911
Softmax	0.9213	0.9325	0.9213	0.9197	0.9207

With the highest accuracy (0.9213), precision (0.9325), recall (0.9213), F_1-score (0.9197), and Cohen's Kappa (0.9207), softmax outperforms the others, as shown by the ML classified results in Table 3. It can also be demonstrated that the SVM, MLP, and LDA perform exceptionally well regarding balanced precision, recall, and AUC scores. On the other hand, overfitting and data distribution sensitivity cause Decision Tree (DT) and KNN to perform poorly, with low accuracy and agreement scores, respectively. Despite being straightforward, Naive Bayes (NB) performs poorly because of its high independence assumptions, whereas Random Forest and Extra Trees are effective because they utilize ensemble techniques. These findings indicate that the VGG19 descriptor is able to extract intricate features, and the softmax classifier will be superior to conventional classifiers.

Table 4. The Classification Results of the VGG16 Feature Descriptor

Classifiers	**Accuracy**	**Precision**	**Recall**	**F_1-Score**	**Cohen's Kappa**
Random Forest	0.6713	0.6858	0.6713	0.6510	0.6685
LDA	0.7753	0.8199	0.7753	0.7691	0.7733
KNN	0.4368	0.6270	0.4368	0.4587	0.4317
SVM	0.8146	0.8384	0.8146	0.8074	0.8130
Decision Tree	0.2051	0.2355	0.2051	0.2047	0.1984
Naive Bayes	0.4565	0.5497	0.4565	0.4595	0.4517
MLP	0.7837	0.8226	0.7837	0.7854	0.7818
Extra Trees	0.6826	0.6956	0.6826	0.6606	0.6798
Softmax	0.9266	0.9277	0.9266	0.9221	0.9262

Table 4 displays the comparative results of different classifiers in the VGG16 scenario. This table shows that softmax has the highest precision (0.9277), recall (0.9266), F_1-score (00.9221), Cohen's Kappa (0.9262), and accuracy (0.9266) when compared to other classifiers. Furthermore, considering the precision value (0.8384), SVM handles the complexity of data using VGG16 quite well. With an accuracy of 78.37%, the MLP model performs moderately. While LDA (0.7733), Random Forest (0.6685), and Extra Trees (0.6798) offer comparable agreement based on Cohen's kappa, their lower precision and recall compared to softmax suggest that they are more likely to generate classification errors. Perhaps as a result of their poor performance metrics, KNN and Naïve Bayes fall behind. Because it either overfits or underfits the Iris dataset's rich features, Decision Trees perform poorly on it. Softmax is the best classifier, according to Table 5's comparative analysis, with the highest accuracy (0.8441), precision (0.8623), recall (0.8441), F_1-score (0.8337), and Cohen's Kappa (0.8427).

According to the classifier comparison in Table 5, softmax performs better than the others with the highest accuracy (0.8441), precision (0.8623), recall (0.8441), F_1-score (0.8337), and Cohen's Kappa (0.8427). These findings demonstrate softmax's high predictive ability and dependability in the case of ResNet50. With a robust accuracy of 0.8160, SVM also performs well. Softmax, which exhibits balanced metrics, comes in second. Additionally, the LDA attains an accuracy of 0.7542, closely trailed by MLP and Extra Trees. Cohen's Kappa indicates less successful classification, and Decision Tree and KNN perform worse with noticeably low accuracy.

Table 5. The ML Classified Results for the Resnet50 Feature Descriptor

Classifiers	Accuracy	Precision	Recall	F_1-Score	Cohen's Kappa
Random Forest	0.6671	0.6975	0.6671	0.6463	0.6642
LDA	0.7542	0.8007	0.7542	0.7458	0.7520
KNN	0.4621	0.6420	0.4621	0.4742	0.4573
SVM	0.8160	0.8456	0.8160	0.8101	0.8144
Decision Tree	0.1924	0.2115	0.1924	0.1870	0.1856
Naïve Bayes	0.5323	0.5937	0.5323	0.5187	0.5282
MLP	0.7093	0.7593	0.7093	0.7059	0.7067
Extra Trees	0.6868	0.7198	0.6868	0.6685	0.6841
Softmax	0.8441	0.8623	0.8441	0.8337	0.8427

Table 6 displays the ML classified results for the customized CNN feature model. The table demonstrates that, in contrast to the VGG16, VGG19, and ResNet50 models, all ML models unexpectedly outperformed them using customized 32-layer CNN deep features. All eight classifiers, with the exception of Naïve Bayes, yield results that are almost identical for every performance metric. As a result of its ability to fully utilize the deep hierarchical features extracted by the CNN, the softmax classifier achieves the best accuracy (93.40%) in distant iris recognition. Softmax, a probabilistic classifier, offers the best precision, recall, F_1-score, and Cohen's Kappa, resulting in a distinct and accurately calibrated decision boundary.

Table 6. The ML Classified Results for the Customized CNN Feature Descriptor

Classifier	Accuracy	Precision	Recall	F_1-Score	Cohen's Kappa
Random Forest	0.9059	0.9364	0.9058	0.9095	0.9050
LDA	0.8932	0.9280	0.8932	0.8993	0.8923
KNN	0.8960	0.9433	0.8960	0.9068	0.8951
SVM	0.9017	0.9285	0.9017	0.9039	0.9008
Decision Tree	0.8904	0.9181	0.8904	0.8921	0.8894
Naïve Bayes	0.7275	0.8470	0.7275	0.7429	0.7252
MLP	0.9157	0.9300	0.9157	0.9149	0.9150
Extra Trees	0.9227	0.9354	0.9227	0.9215	0.9220
Softmax	0.9340	0.9431	0.9340	0.9325	0.9334

The aforementioned statistical analysis indicates that for distant iris recognition, the customized CNN-based features converted with a softmax classification model perform better than alternative deep feature-based machine learning models. Transfer learning's capacity to efficiently extract iris features through multi-layer feature representation is what gives it its discriminative power. Additionally, we use ROC curves to analyze performance in order to assess the efficacy of ML-based transfer learning as well as the discriminative power of the

feature descriptors. Performance is shown by proximity to the top-left corner of ROC curves, which are plotted using the false positive and true positive rates across various classification thresholds. Figures 4 and 5 show the ROC curves for the four deep feature-based and nine machine learning classifiers.

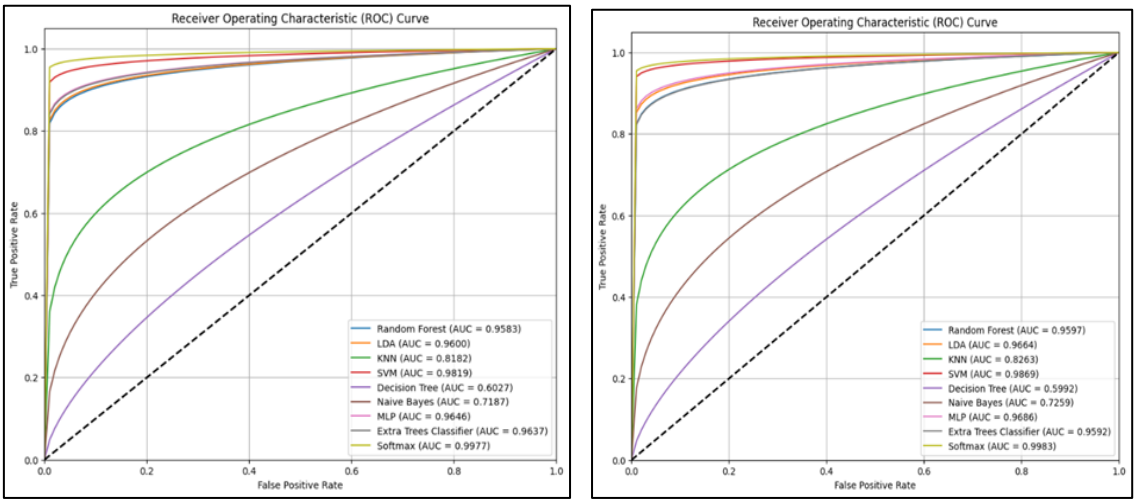

Figure 4. The ROC Curves of ML Models with (a) VGG19 (Left) and (b) VGG16 (Right)

Multiple classifiers with VGG19 are compared in Fig. 4(a) according to their AUC scores, which show how well they can differentiate between classes. With an AUC of 0.9977, the softmax model performs better than any other model, whereas SVM receives an AUC score of 0.9819. Furthermore, with an AUC of 0.96, the MLP, LDA, Extra Trees, and Random Forest models come in close succession. Across the threshold range, these models exhibit strong classification capabilities with low false positive rates and high true positive rates. Although it performs fairly well, the KNN is marginally less efficient than the best-performing models, with an AUC of 0.8182. However, with AUCs of 0.7187 and 0.6027, respectively, the Naive Bayes and Decision Tree classifiers perform noticeably worse, suggesting that their discriminatory power is limited.

The ROC curves based on AUC scores of different competitive ML models with VGG16 are displayed in Fig. 4(b). This figure shows that the softmax provides excellent discrimination capability between classes with an AUC score of 0.9983. The next best performers are SVM (AUC=0.9869), MLP and LDA (AUC≈0.97), and Extra Trees and Random Forest (AUC≈0.96), all demonstrating strong, reliable classification performance. In contrast, KNN demonstrates significantly reduced effectiveness with an AUC of 0.8263, suggesting moderate performance that may be sensitive to dataset characteristics or feature scaling. Naive Bayes performs even less effectively (AUC=0.7259), showing limited predictive capability. Decision Trees prove least effective with an AUC of 0.5992, offering only marginal improvement over random chance. Overall, ensemble and linear models deliver the strongest and most reliable performance, while KNN, Naive Bayes, and Decision Trees show substantially lower classification accuracy on this dataset.

The ROC curve shown in Fig. 5(a) compares the classification performance of multiple ML models for Resnet50. In this figure, softmax and SVM show nearly identical ROC curves with AUC scores approaching 0.99. LDA (AUC=0.9769), MLP (0.9653), Random Forest, and Extra Trees (both 0.9635) also demonstrate excellent performance, as evidenced by their curves

tightly following the top-left boundary characteristic of models with near-perfect classification accuracy. In contrast, the Decision Tree performs poorly (AUC=0.5928), with its curve approaching the diagonal line of random chance (AUC=0.5). While KNN (AUC=0.8504) and Naive Bayes (AUC=0.7641) show moderate performance, both indicate substantial room for improvement.

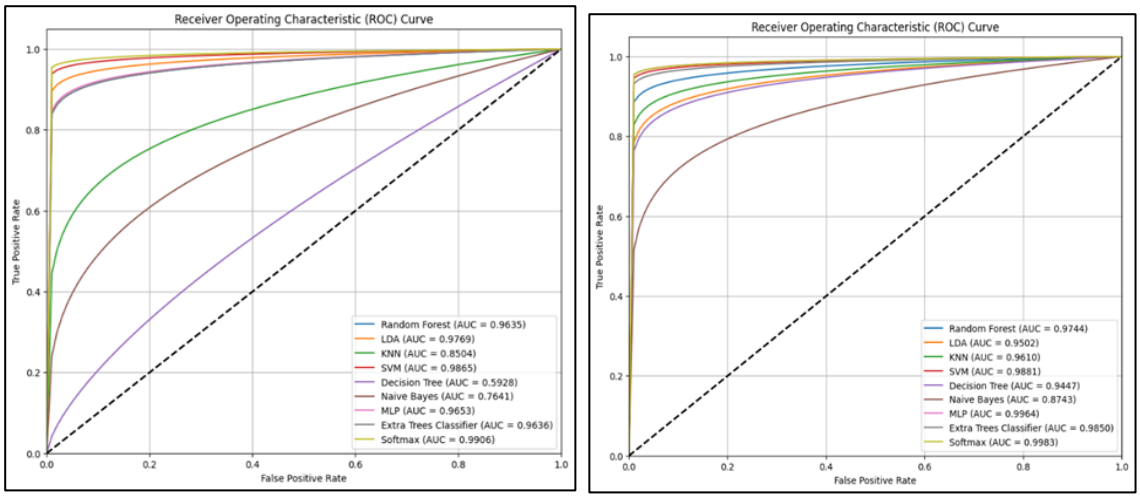

Figure 5. The ROC Curves of ML Models with (a) Resnet50 (Left) and (b) Customized CNN (Right)

Fig. 5(b) shows the ROC curve of the ML classifiers with customized CNN models. The ROC curve analysis reveals a softmax classifier achieving near-perfect discrimination (AUC=0.9983), with its curve showing an immediate steep rise to a 1.0 true positive rate. MLP (AUC=0.9964) and SVM (AUC=0.9881) exhibit similarly sharp initial curve ascents. Extra Trees (AUC=0.9850) and Random Forest (AUC=0.9744) maintain strong top-left curve positioning. Moderately performing models like KNN (AUC=0.9610) show gradual curve progression, while LDA (AUC=0.9502) and Decision Trees (AUC=0.9447) display flatter trajectories, indicating weaker early true positive capture. Naive Bayes (AUC=0.8743) proves least effective, its curve approaching the diagonal random-guessing line. Such performance reflects balanced model accuracy, though it may also suggest possible overfitting if evaluated on training data.

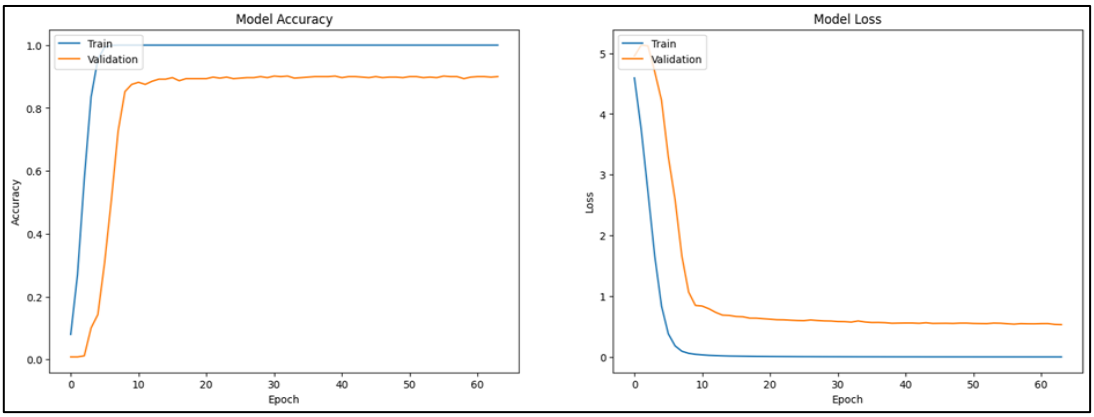

Figure 6. (a) Model Accuracy (Left); (b) Model Loss (Right) of the Softmax Classification with a Customized CNN

In this paper, model validation was conducted through accuracy and loss measurements for both training and validation datasets, as shown in Fig. 6. Specifically, Fig. 6(a) demonstrates the evolution of model accuracy over training epochs for both datasets, while Fig. 6(b) displays the corresponding training loss. The model achieves rapid convergence to near-perfect training accuracy, indicating strong pattern recognition capability. This is supported by the training loss that quickly approaches zero. However, validation loss decreases more gradually before stabilizing at a higher value. The persistent gap between training and validation metrics in both plots suggests the model has high learning capacity, though it may be fitting the training data too closely.

During testing, the CNN shows satisfactory accuracy but also notable limitations. Misclassifications occur mainly among classes with similar textures or intensity patterns, a challenge amplified by the use of grayscale images lacking color cues. Minority classes suffer from lower precision, recall, and F1-score, reflecting bias toward majority classes. Signs of overfitting are evident from the gap between training and validation accuracy in later epochs. Fixed input resizing (224×224) likely causes loss of fine details, reducing performance on subtle variations. Additionally, the deep model demands significant training time and GPU memory, limiting scalability. These issues suggest the need for data augmentation and class rebalancing to improve generalization.

4.4 Performance Comparison

To evaluate the performance of our proposed method, we conduct extensive experiments and compare the results against several baseline approaches, which are presented in Table 7. The table presents classification accuracies achieved using different feature extraction descriptors and classification techniques. Some of these techniques evaluated classification performance across different databases using varying amounts of training and testing images. For example, Tan and Kumar used only the first eight left or right eye images from the CASIA-v4 iris dataset [26], achieving 93.90% accuracy when training on the first 10 subjects and testing on subjects 11–141. Kumar's method obtained 90.43% recognition using 79 training and 961 test images [37].

Table 7. Performance Comparison of the Proposed Approach with the Competitive Approaches

Competitive Approaches	**Accuracy**
Discrete Fast Fourier transform and softmax, Szymkowski et al. [39]	78.70%
Histogram of Oriented Gradients and Canberra distance, Mukherjee et al. [1]	90.55%
Gradient Local Auto-correlation and Correlation distance, Mukherjee et al. [2]	91.84%
Convolutional Neural Network and KELM, Ali et al. [16]	92.00%
Log-Gabor and Hamming Distance, Tan et al. [38]	92.90%
Customized CNN with softmax Classifier (Proposed)	93.40%

Moreover, Tan and Kumar used geometric key-based iris encoding using Hamming distance and obtained 92.90% recognition accuracy on the same database [38]. In 2021,

Szymkowski developed a distant iris recognition framework using discrete fast Fourier transform (DFT) with machine learning algorithms, where the recognition rate was 78.70% on the CASIA-v4 database [39]. It is noteworthy that the authors in [17] reported significant advantages in using multi-feature descriptors. Although the accuracy of 98.17% is the highest among the results, their performance cannot be directly compared to ours. This is because the authors incorporated both iris images and surrounding eye regions such as the pupil, eyelash, eyelid, and sclera. Therefore, directly comparing our findings with those reported by others is not possible. We employ a total of 3,975 images, using 80% for training and the other 20% for testing, while excluding the first 14 subjects with regular eye images. As illustrated in Fig. 3, the images within our investigated dataset exhibit greater complexity and non-linearity compared to those in existing datasets. The top accuracy of 93.40% is achieved from the proposed deep transfer CNN-based softmax model in this domain.

5. Conclusion and Future Work

This study introduces a reliable biometric recognition system that leverages a deep learning architecture to perform bilateral iris analysis for accurate human identification. The proposed customized CNN architecture includes a softmax layer and incorporate diverse training strategies aimed at preventing overfitting and enhancing the neural network's ability to generalize across unseen patterns. Despite achieving the highest possible average precision, recall, F_1-score, Cohen's Kappa, and overall classification accuracy on this database with the lowest noise implications. The proposed transfer learning model continues to face challenges associated with class imbalance. In the future, we will focus on enhancing iris preprocessing to achieve more accurate segmentation by mitigating noise arising from uneven illumination, eyelashes, and eyelid occlusions. Additional enhancements may include implementing feature-level fusion techniques and robust learning models to address class imbalance in the data.

Conflicts of Interest

The authors hereby declare that there are no conflicts of interest concerning the content or publication of this manuscript. This declaration applies to all aspects of the research and its presentation.

Acknowledgement

The authors gratefully acknowledge the financial support provided by the Research and Innovation Centre (RIC) of Khulna University, Bangladesh, under Research Grant No. KURIC-46/2021 for the financial year 2021-2022. The authors also sincerely thank the Institute of Automation, Chinese Academy of Sciences (CASIA) for providing access to the iris image dataset used in this study. Additionally, the authors acknowledge the Laboratory of Mathematics and Applied Mathematics (LMAM), Peking University, China, for supporting the data processing for this research.

References

[1] Mukherjee, Arnab, Zahidul Islam, and Lasker Ershad Ali. "Human Iris Classification through Histogram of Oriented Gradient Features with Various Distance Metrics." Machine Graphics and Vision 33 (2024).

[2] Mukherjee, A., Ripon, K. S. N., Ali, L. E., Islam, M. Z., & Mamun-Al-Imran, G. M. (2022). Image gradient-based iris recognition for distantly acquired face images using distance classifiers. International Conference on Computational Science and Its Applications. Springer: 239–252.

[3] Tan, Chun-Wei, and Ajay Kumar. "Efficient and accurate at-a-distance iris recognition using geometric key-based iris encoding." IEEE Transactions on Information Forensics and Security 9, no. 9 (2014): 1518-1526.

[4] Al-Waisy, A. S., Qahwaji, R., Ipson, S., Al-Fahdawi, S., & Nagem, T. A. M. (2018). A multi-biometric iris recognition system based on a deep learning approach. Pattern Analysis and Applications, 21, 783–802.

[5] Yin, Y., He, S., Zhang, R., Chang, H., & Zhang, J. (2025). Deep learning for iris recognition: A review. Neural Computing and Applications. Advance online publication.

[6] Nguyen, K., Proença, H., & Alonso-Fernandez, F. (2024). Deep learning for iris recognition: A survey. ACM Computing Surveys, 56(9), 1–35.

[7] Daugman, J. G. (2002). High confidence visual recognition of persons by a test of statistical independence. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(11), 1148–1161.

[8] Suvarchala, P. V. L., & Kumar, S. S. (2018). Texture synthesis and modified filter bank in contourlets for improved iris recognition. Pattern Analysis and Applications, 21(4), 1127–1138.

[9] Ahmadi, N., Nilashi, M., Samad, S., Rashid, T. A., & Ahmadi, H. (2019). An intelligent method for iris recognition using supervised machine learning techniques. Optics & Laser Technology, 120, 105701.

[10] Kaewphaluk, K., & Widjaja, J. (2017). Experimental demonstrations of noise-robustness of compression-based joint wavelet transform correlator in retinal recognition. Optik, 142, 168–173.

[11] Chirchi, V., & Waghmare, L. M. (2017). Enhanced isocentric segmentor and wavelet rectangular coder to iris segmentation and recognition. International Journal of Intelligent Engineering and Systems, 10(6), 1–10.

[12] Kim, Jaemin, Seongwon Cho, Jinsu Choi, and Robert J. Marks. "Iris recognition using wavelet features." Journal of VLSI signal processing systems for signal, image and video technology 38, no. 2 (2004): 147-156.

[13] Zhao, T., Liu, Y., Huo, G., & Zhu, X. (2019). A deep learning iris recognition method based on a capsule network architecture. IEEE Access, 7, 49691–49701.

[14] Arsalan, M., Hong, H. G., Naqvi, R. A., Lee, M. B., Kim, M. C., Kim, D. S., Kim, C. S., & Park, K. R. (2017). Deep learning-based iris segmentation for iris recognition in a visible light environment. Symmetry, 9(11), 263.

[15] Ren, M., Wang, Y., Zhu, Y., Zhang, K., & Sun, Z. (2023). Multiscale dynamic graph representation for biometric recognition with occlusions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(12), 15120–15136.

[16] Ali, L. E., Luo, J., & Ma, J. (2017). Effective iris recognition for distant images using log-Gabor wavelet based contourlet transform features. Intelligent Computing Theories and Application: 13th International Conference, ICIC 2017 Springer: 293–303.

[17] Ali, L. E., Luo, J., & Ma, J. (2016). Iris recognition from distant images based on multiple feature descriptors and classifiers. 2016 IEEE 13th International Conference on Signal Processing (ICSP) IEEE: 1357–1362.

[18] Kaur, B., Singh, S., & Kumar, J. (2018). Iris recognition using Zernike moments and polar harmonic transforms. Arabian Journal for Science and Engineering, 43(12), 7209–7218.

[19] Kaur, B., Singh, S., & Kumar, J. (2018). Robust iris recognition using moment invariants. Wireless Personal Communications, 99(2), 799–828.

[20] Şerifi, M., & Şerifi, Ü. (2025). Dual CNN and texture-based face-iris multimodal biometric system via decision-level fusion. Signal, Image and Video Processing, 19(4), 292.

[21] Farouk, R. H., Mohsen, H., & Abd El-Latif, Y. M. (2022). A proposed biometric technique for improving iris recognition. International Journal of Computational Intelligence Systems, 15(1), 79.

[22] El-Sofany, H., Bouallegue, B., & Abd El-Latif, Y. M. (2024). A proposed biometric authentication hybrid approach using iris recognition for improving cloud security. Heliyon, 10(16).

[23] Lee, M. B., Kim, Y. H., & Park, K. R. (2019). Conditional generative adversarial network-based data augmentation for enhancement of iris recognition accuracy. IEEE Access, 7, 122134–122152.

[24] Ribeiro, E., Uhl, A., & Alonso-Fernandez, F. (2019). Iris super-resolution using CNNs: Is photo-realism important to iris recognition? IET Biometrics, 8(1), 69–78.

[25] Tan, C.-W., & Kumar, A. (2012). Unified framework for automated iris segmentation using distantly acquired face images. IEEE Transactions on Image Processing, 21(9), 4068–4079.

[26] Tan, C.-W., & Kumar, A. (2013). Towards online iris and periocular recognition under relaxed imaging constraints. IEEE Transactions on Image Processing, 22(10), 3751–3765.

[27] Daugman, J. (2009). How iris recognition works. In The essential guide to image processing Academic Press: 715–739.

[28] Nguyen, K., Fookes, C., Ross, A., & Sridharan, S. (2018). Iris recognition with off-the-shelf CNN features: A deep learning perspective. IEEE Access, 6, 18848–18855.

[29] Boyd, A., Fang, Z., Czajka, A., & Bowyer, K. W. (2020). Iris presentation attack detection: Where are we now? Pattern Recognition Letters, 138, 483–489.

[30] Minaee, S., Abdolrashidiy, A., & Wang, Y. (2016). An experimental study of deep convolutional features for iris recognition. 2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 1–6.

[31] Zhao, T., Liu, Y., Huo, G., & Zhu, X. (2019). A deep learning iris recognition method based on capsule network architecture. IEEE Access, 7, 49691–49701.

[32] Zeng, R., Wu, J., Shao, Z., Senhadji, L., & Shu, H. (2014). Quaternion softmax classifier. Electronics Letters, 50(25), 1929–1931.

[33] Sun, Zhenshi, Haokun Yang, Ming Fang, Yibo Dai, Dayong Huang, and Chengwei Zhao. "On Bayesian optimization-based CNN-BiLSTM network for multi-class classification in distributed optical fiber vibration sensing systems." IEEE Transactions on Instrumentation and Measurement (2024).

[34] Yao, Jingtao, Yili Li, and Chew Lim Tan. "Option price forecasting using neural networks." Omega 28, no. 4 (2000): 455-466.

[35] Koklu, Murat, and Ilker Ali Ozkan. "Multiclass classification of dry beans using computer vision and machine learning techniques." Computers and Electronics in Agriculture 174 (2020): 105507.

[36] Tan, T., & Sun, Z. (n.d.). CASIA Iris Image Database. [Accessed: 28 Apr 2022]. http://biometrics.idealtest.org/#/datasetDetail/4

[37] Kumar, A., Chan, T.-S., & Tan, C.-W. (2012). Human identification from at-a-distance face images using sparse representation of local iris features. 2012 5th IAPR International Conference on Biometrics (ICB) IEEE: 303–309.

[38] Tan, C.-W., & Kumar, A. (2014). Efficient and accurate at-a-distance iris recognition using geometric key-based iris encoding. IEEE Transactions on Information Forensics and Security, 9(9), 1518-1526.

[39] Szymkowski, M., Jasiński, P., & Saeed, K. (2021). Iris-based human identity recognition with machine learning methods and discrete fast Fourier transform. Innovations in Systems and Software Engineering, 17(3), 309–317