



A Scientific Programming Based Effective and Precise Iris Acknowledgement Algorithm

Rahul Sawhney¹, Shilpi Sharma^{1*}, Vipul Narayan², Swapnita Srivastava³, Shashank Awasthi^{3,4},
Naresh Kumar²

¹ Department of Computer Science and Engineering, Amity University, Noida 201313, India

² Department of Computer Science and Engineering, Galgotias University, Greater Noida 203201, India

³ GL Bajaj Institute of Technology and Management, Greater Noida 203201, India

⁴ Research Management Center, Management and Science University, Shah Alam 40100, Malaysia

Corresponding Author Email: ssharma22@amity.edu

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.410614>

ABSTRACT

Received: 18 September 2023

Revised: 26 March 2024

Accepted: 12 July 2024

Available online: 31 December 2024

Keywords:

deep iris recognition, deep neural networks, deep learning, iris convolutional neural network, GPU, PyTorch

Machine learning is being increasingly used in several computer science specializations. A deep conventional neural network is a great option for machine learning applications requiring a strong visual model. We demonstrate the robustness and dependability of the iris recognition system. The image goes through several preparation procedures, such as having its quality improved, having its centre and radius of the iris and pupil located for iris segmentation, and having its coordinate system converted from Cartesian to polar. Since feature extraction and classification can be performed automatically using IRISNet, it is recommended as a system to utilize. CNN features are extracted and then classified into N classes using a SoftMax layer in IRISNet, with weight updates performed by backpropagation and the learning rate fine-tuned using Adam optimization. The suggested technique was evaluated by using the IITD V1 iris database. When compared to supervised classification models, the proposed method yields better results (SVM, KNN, DT, and NB). Without any adjustments to the images, 96.34% were able to be identified, whereas after normalization, this number dropped to 96.43%.

1. INTRODUCTION

In modern times, technology has become an integral component of almost everyone's daily life. Common practices are inefficient and need to be replaced. Credit cards, identity cards, passports, and similar items are all easy to misplace, hack, forget, or otherwise damage [1]. This has prompted the development of more robust, hack-proof alternatives to the standard practice of storing valuables in a safe deposit box. This safety in recognition and authentication can be achieved using biometrics, as every individual possesses unique features and characteristics that no two people could share. These include, but are not limited to, the patterns of lines in a fingerprint, the pitch and volume of a voice, the size and shape of a person's face, the colour of their eyes, and the shape of their irises, among many others [2].

One of the most secure methods of biometric identification is iris recognition. The colourful ring around the pupil of the eye, known as the iris, is a dependable technique of identifying a person since it is unique to each person and does not change over time. Furthermore, iris recognition systems gather high-quality photos of the iris using sophisticated image processing methods, which further increases the system's dependability. The photos are compared to a reference template that has been saved in a database in order to verify the subject's identification [3]. Extensive testing and evaluation of iris recognition under many settings, such as varied illumination, has shown that it is

very accurate and reliable. Most Iris recognition systems have fairly low False Rejection Rates (FRRs) and False Acceptance Rates (FARs), the two major criteria used to assess the performance of such systems. Because of the iris's uniqueness and immutability, the abundance of data contained within it, and the sophisticated image processing methods used by the system, iris recognition is widely regarded as a secure and reliable means of biometric identification.

To create a successful biometric system, one must adhere to a certain procedure. To begin, an image or video is captured that contains the feature to be recognized, such as an eye or a face. Next, the input image is pre-processed since the camera lens may contain dust, or the subject may have moved slightly during capture, resulting in the image being rotated or distorted. Next, an iris or face must be found and turned into a template that can be put to a database and compared to other human templates for recognition. The feature extraction, classification, and recognition methods used in the construction of a rapid and accurate system that fulfils acceptable standards are crucial. Tasks of this kind are perfectly within Deep Learning's capabilities. Deep learning has been more popular in recent years because of machine learning's inherent capacity to automatically extract features, classify data, and identify patterns. This has prompted a slew of new research into the topic. Face recognition and object identification are two examples of pattern recognition applications that make use of deep learning algorithms [4]. The iris identification system is a

good example of various real-world applications of deep learning. In this system, deep learning has been used to address difficulties with segmentation, classification, and recognition. Here are some examples. After the iris has been manually segmented from the lid, lashes, and sclera by human operators, we propose doing research on an iris identification system that makes use of a deep convolutional neural network to automatically extract information from the segmented iris [5].

Machine learning is a rapidly growing field within computer science that involves the development of algorithms and statistical models that enable computers to learn from data, without being explicitly programmed. It is being increasingly used in several computer science specializations such as:

Computer Vision: Machine learning is used to analyze and interpret visual data, such as images and videos, for tasks such as object recognition, facial recognition, and image classification [6].

Natural Language Processing: Machine learning is used to process and analyze human language, such as text and speech, for tasks such as language translation, text summarization, and sentiment analysis.

Robotics: Machine learning is used to enable robots to learn from their environment and improve their performance over time, such as object detection, grasping and manipulation [7].

Recommender Systems: Machine learning is used to build systems that can make personalized recommendations, such as product recommendations for e-commerce websites or content recommendations for streaming platforms.

Medical Informatics: Machine learning is used to assist in the analysis of medical data, such as images, and improve diagnoses, treatment planning and monitoring of patients.

Finance: Machine learning is used to analyze financial data, such as stock prices and market trends, and make predictions about future financial performance [8].

It is believed that the one-of-a-kind patterns of an individual's iris's texture are created at random during the fetal development of the eye and that these patterns do not alter as the person gets older. Iris patterns are sufficiently unique that they may sometimes be utilised to differentiate between a pair of identical twins. Iris recognition is one of the most secure and dependable forms of biometric identification, which is why it has been widely used in a variety of industries, including banking, border control, mobile phones, and many others. Iris recognition is a non-contact approach that exposes the individual to less germs and pathogens than other popular biometric methods such as face recognition, palm print recognition, and fingerprint scanning. Other frequent biometric methods include palm print recognition and fingerprint scanning. As more people learn about the advantages of iris recognition, more studies have been conducted to determine the best technique to extract characteristics from irises [9, 10]. The most common applications of iris recognition are authentication and identification. In the verification phase of an iris recognition system, a submitted iris image is checked against a library of previously received pictures. In the identification mode, the system compares this iris scan to a database of known identities to answer the query "who is he?"

One form of machine learning model that mimics the human brain in structure is called a deep convolutional neural network (DCNN). It's a network of neurons that has been taught to recognise certain visual or auditory patterns in data. Densely connected networks (DCNNs) have several layers, with many artificial neurons (or nodes) making up each layer. Each successive layer learns a more sophisticated representation of

the data that was ingested by the input layer. Predictions made by the model are output at the last layer of the network. DCNNs excel because they can learn features automatically from the data, without the requirement for human-engineered features. It does this by training the network to modify the weights and biases of its synthetic neurons. Picture and video identification applications including object detection, face recognition, and image classification all benefit greatly from the usage of deep convolutional neural networks (DCNNs) [11]. As a result, they find further use in the field of natural language processing, where they are employed for purposes like translation and analysis of emotional tone. DCNNs' strength as a learning tool lies in their capacity to process and adapt to massive data sets. They have the capacity to analyse high-resolution photos and movies, as well as massive volumes of textual and acoustic information.

The technique that is evaluated using the IITD V1 iris database typically involves the following steps:

Image Acquisition: The first step is to acquire high-quality images of the iris. This is typically done using an iris camera, which captures images of the iris in visible light.

Image Pre-processing: The acquired images are then pre-processed to remove any noise and artifacts, such as reflections or eyelashes, that may be present in the image.

Feature Extraction: The pre-processed images are then used to extract unique features from the iris. These features are typically based on the number, size, and location of furrows and ridges in the iris.

Pre-processed images often come from an iris camera and show the subject's iris. The picture must be pre-processed to get rid of any noise or artefacts before it can be utilised for iris identification. This is significant because background noise and other distortions may reduce the accuracy of an iris recognition system. Dust on the camera lens is a typical cause of artefacts and noise in iris photos. Light specks caused by dust on the lens are often misinterpreted as being a component of the iris [12]. The iris recognition system may therefore make erroneous or incorrect connections as a result. Pre-processing methods, such as image denoising, are used to clean the input picture of dust and other forms of noise. These methods use mathematical algorithms to clean up a picture without losing any of the iris's essential details. Methods used for eliminating noise include median filtering, Gaussian filtering, and Wiener filtering, to name a few. Picture normalisation is another frequent preprocessing technique that is used to modify the brightness and contrast of the image such that the iris area is in the same place in all photographs [13].

There are a number of reasons why deep learning has grown so popular in recent years. Improvements in technology, such as graphics processing units (GPUs), have made it feasible to train huge, complicated deep learning models, which has contributed to deep learning's rising popularity. These models need a great deal of computing power, but with the availability of powerful hardware, they can now be trained in a very short length of time [14]. One of the main reasons for deep learning's success is the abundance of data now available. In order to train their models, deep learning systems need a lot of information, and the proliferation of Internet-connected gadgets and the expansion of the web have made it feasible to amass massive volumes of data on all sorts of topics. Image recognition, language translation, and voice recognition are just few of the areas where deep learning models have surpassed human performance. As a result of this breakthrough, deep learning is now being used in a wider range of fields and

contexts than ever before [15, 16].

The feature extraction, classification, and recognition methods used in the construction of a rapid and accurate biometric system will depend on the specific biometric modality being used. However, some common methods that are used in biometric systems include:

Feature Extraction: This is the process of extracting relevant and unique features from the biometric data. For example, in iris recognition, the unique features of the iris, such as the number, size, and location of furrows and ridges, are extracted from the iris images. Other biometric modalities such as fingerprint, face, and voice have their own unique feature extraction methods.

Classification: Once the features have been extracted, they are used to train a classifier. A classifier is a model that assigns a label or class to an input based on its features. For example, in iris recognition, the classifier will assign an iris image to a specific individual based on its unique features.

Recognition: The final step is recognition, where the trained classifier is used to match an input biometric sample with a template in a database. The matching process compares the features of the input sample to the features of the templates in the database and assigns the most similar match as the recognition result.

Machine Learning Algorithms: To improve the performance of the system, machine learning algorithms such as deep learning, Random Forest, Support Vector Machine (SVM) and others can be used for feature extraction, classification, and recognition [17].

Several procedures must be followed to create a biometric system that is effective and trustworthy. The development of a reliable biometric system typically involves the following stages: First, we must precisely specify what features the biometric system must have. The system's performance objectives, security and privacy concerns, and the biometric modality to be employed (fingerprint, iris, face, etc.) are all factors to be considered. After specifications have been finalised, the next stage is to collect a sizable body of biometric information. To provide accurate results from the biometric system, the dataset used to train it should accurately reflect the target audience. Next, the dataset undergoes pre-processing, which includes feature extraction. Iris recognition software, for instance, can read the iris and determine things like the person's age, gender, and race based on the pattern of furrows and ridges. Once the characteristics have been collected, they are utilised to train a deep learning model, such as a convolutional neural network (CNN), that will be used to identify the biometric information. Cross-validation methods should be used during training and testing of the model [18]. Evaluating the model's accuracy and performance after training is necessary. You may accomplish this by running the model on a new dataset and comparing its performance to the benchmarks you established in the first phase. Finally, the biometric system must be implemented in the actual world and distributed to the intended users.

The primary goal of the initial stage of processing in iris recognition systems is to isolate and authenticate regions of iris texture. This is done to ensure that the system is able to accurately and reliably identify an individual based on the unique features of their iris.

The initial stage of processing typically involves several steps:

Image Segmentation: The first step is to segment the iris region from the rest of the eye in the image. This is done to

remove any extraneous information from the image, such as the sclera, eyelashes, and eyelids, that may be present.

Localization: Once the iris region has been isolated, the next step is to locate the iris within the image. This is done by finding the inner and outer boundaries of the iris, which define the region of iris texture.

Normalization: The iris region is then normalized, which involves adjusting the image so that the iris is in the same position in all images. This helps to ensure that the iris is in the same position and orientation in all images, which can improve the performance of the system.

Authentication: The last step is to ensure that the iris region is authentic, which means that the iris is from a living person and not a fake one. This can be done by analyzing the iris texture and checking for signs of tampering or forgery.

Biometric authentication is now one of the most promising and trustworthy options. Biometrics have the potential to serve as a kind of permanent identity, used in lieu of documents like passports or passwords that need the user to memorise long strings of characters. The use of irises as a form of identification has been gaining traction since it was first used in 1992 [19]. To begin with, it has to be unique enough that the likelihood of two persons sharing the same characteristic is minimal; second, it needs to be stable enough that it doesn't vary significantly over time; and third, it needs to be readily collected without being too invasive or bothersome for users. Iris recognition technology has the potential to solve all of these problems. Between the pupil and the white of the eye, there is a highly textured region called the iris. When compared to alternative facial, fingerprint, or voice biometrics technologies, iris recognition is often considered to be the most reliable and safe option. First of all, it (1) varies between even identical twins and between the eyes of the same person. The eye's physiological response to light acts as a natural test against counterfeit irises [2, 3], and (2) it is a protected internal organ. The appearance of one's visual texture is fixed from the time of conception [4]. Further (3) iris-based recognition is generally accepted since it poses no security risks to its users. Also, these days it just takes a single glance at the iris capture gadget to unlock the door. You'll save so much time using it. Here are some of the points that will be further upon in the article's remaining sections: Part 2 introduces the problem, Part 3 explains several approaches to solving it, and Part 4 summarises the findings. The results and recommendations for further study are discussed in Section 5.

Our study's primary objectives are to leverage deep CNNs for iris recognition, investigate preprocessing techniques, compare the proposed approach with traditional models, and evaluate the impact of normalization, ultimately contributing to advancements in the field of visual-based machine learning and biometric identification systems.

- The study aims to demonstrate the effectiveness of utilizing deep CNNs, specifically the IRISNet framework, for iris recognition applications.
- It explores various preprocessing techniques such as image quality enhancement, iris segmentation, and feature extraction to improve recognition accuracy.
- The research compares the proposed CNN-based approach with traditional supervised classification models (SVM, KNN, DT, NB) to showcase its superiority.
- The impact of normalization on system performance is evaluated to understand its role in enhancing identification rates.
- The study's contributions lie in providing a robust and

dependable iris recognition system capable of real-time identification with high accuracy, thereby advancing the field's capabilities and applications.

2. LITERATURE REVIEW

Ophthalmologist Frank Birch proposed utilizing an individual's iris as a unique identifier for the first time in 1936. Because of the uniqueness of everyone's iris, the therapeutic theories of American ophthalmologists Aran Safir and Leonard Flum, who accepted Birch's notion, were unable to be refined. Dougman is credited with introducing the first iris recognition system, which he also improved upon by applying it to many databases at once. You can extract iris features using a Gabor filter, and then utilize the Hamming distance for matching [7]. To determine the characteristics of an iris, Bilos employs Zero crossing Wavelet Transform (WT). The iris picture is first normalized so that it has the same amount of data points, and only then are the characteristics extracted.

Specifically, a dissimilarity function was employed for the matching process [8]. The LAMASTER neural network is utilized by Homayon to partition the database and facilitate iris recognition [9]. Classification is one area where deep learning has shown its worth, and it has been widely employed in applications that do not need human interaction. Deep conventional neural networks (DCNNs) have an advantage over regular neural networks in that they can learn characteristics in the same manner that humans can. Several deep conventional neural network (DCNN) approaches have been developed for the purpose of iris recognition. VGGNet are only a few instances of such methods. Each of these approaches was developed by a different group of researchers. On the other hand, the DeepIrisNet-B model substitutes inception modules for the last two blocks in the previous sentence [8].

There has been a recent surge in interest in the study of iris identification techniques that make use of deep learning. Commonly, the iris image is converted into a set of feature

vectors, and the distance between them is calculated, same as in the conventional method. It's possible that deep CNN works well as a feature extractor in these scenarios. The researchers designed a deep convolutional neural network model to ease the collecting of iris features. For us, it was important to not force things to work. Networks like AlexNet and VGG-net, which have been trained on other large-scale image datasets, may be easily applied to the task of iris texture feature extraction due to their better encoding capability [11].

It wasn't until 1987 that the world saw its very first completely automated iris recognition system. According to my understanding, Flom and Safir were awarded the first patent for an iris recognition system for their abstract proposal of such a system. This is the only instance in which I have knowledge of this happening. In the wake of this discovery, a number of methods were developed for the extraction and categorization of iris textures. Traditional, hand-crafted feature engineering approaches and more current, cutting-edge deep learning techniques are the two broad categories that may be used to classify these methods. Both categories can be broken down further into subcategories [10].

In iris identification, convolutional neural networks (CNNs) have garnered interest. CNN-based algorithms perform well and automatically extract features, but they need more training data and computational complexity than older methods. This study trains a unique condensed 2-channel (2-ch) CNN with minimal training data for efficient and reliable iris detection and verification. The first high-performance basic iris classifier is a multi-branch CNN with three well-designed online augmentation techniques and radial attention layers [13]. The model's weight distribution then prunes branches and channels. Fast finetuning, which reduces computational load and improves pruned CNN performance, is optional. We also examine 2-ch CNN encoding and offer an effective iris recognition approach for big database applications [13]. Gradient-based analysis shows that the method is resilient against picture contaminations. Our method performed well for real-time iris recognition on three public iris datasets.

Table 1. Literature review to provide context and justification

Reference	Description	Contribution to Iris Recognition
[7]	Introduced the first iris recognition system and improved it for use across multiple databases.	Pioneered the application of iris recognition in automated systems and improved its scalability for large-scale deployment.
[8]	Developed a dissimilarity function for iris matching, and LAMASTER neural network for database partitioning and iris recognition.	Contributed to the advancement of matching algorithms and database organization techniques for improved iris recognition accuracy.
[10]	Awarded the first patent for an automated iris recognition system, leading to the development of various iris texture extraction and categorization methods.	Set the foundation for automated iris recognition systems and spurred research into iris texture analysis and classification techniques.
[11]	Proposed the use of deep learning, specifically deep convolutional neural networks (DCNNs), for iris feature extraction, leveraging networks like AlexNet and VGG-net.	Contributed to the adoption of deep learning in iris recognition, enhancing feature extraction capabilities and recognition accuracy.
[13]	Developed a unique condensed 2-channel CNN with online augmentation techniques and radial attention layers for efficient and reliable iris detection and verification.	Introduced novel CNN architectures and optimization strategies for efficient iris detection and verification, addressing real-time performance challenges.
[14]	Utilized a Multiobjective Artificial Bee Colony (MABC) approach for feature reduction and classification error rate in iris recognition, achieving high accuracy on public datasets.	Advanced optimization techniques for iris feature extraction and classification, demonstrating high accuracy and performance on standard iris datasets.
[15]	Identified challenges in existing deep learning-based iris recognition algorithms, such as sensitivity to image pollution and lack of hyperparameter calibration.	Highlighted areas for improvement in deep learning-based iris recognition systems, focusing on mitigating sensitivity issues and optimizing model hyperparameters for enhanced performance.

Iris recognition has been a reliable biometric paradigm for person recognition for decades. Criminal-to-commercial items, citizen verification, and border control are examples. The integrated model uses deep learning for precise iris detection and identification. Eye pictures from CASIA and IIT Delhi v1.0 datasets are first evaluated. Daugman's method and Circular Hough Transform properly segment the iris (CHT). DTCWT, Gabor filter, LBP, and GLCM are used to extract hybrid features from segmented iris regions [14]. A Multiobjective Artificial Bee Colony (MABC) approach estimates consistent information to reduce noisy and duplicate feature vectors. MABC method has two multi-objective functions: feature reduction and classification error rate. Autoencoder classification for iris recognition takes active feature vectors. MABC-autoencoder model achieved 99.67% and 98.73% accuracy on CASIA-Iris and IIT Delhi v1.0 iris datasets. Accuracy, specificity, Critical Success Index (CSI), sensitivity, Fowlkes Mallows (FM) index, and Mathews Correlation Coefficient (MCC) determine performance (MCC) [14].

While existing deep learning-based algorithms have shown promise for fully automated end-to-end iris feature extraction and classification, several obstacles remain. As an example, 2-ch techniques, which need a lot of computing power, have only been successfully used to the iris verification issue. The deep learning model's sensitivity to both the degree of image pollution and the amount of the training data further complicates real-time iris recognition. There is also a lack of calibration in the CNN architecture's hyperparameters, such as the number of layers and kernel sizes [15]. Table 1 provides literature review to provide context and justification.

3. METHODOLOGY

3.1 Deep convolutional neural networks

Conversion networks, or ConvNets, are a kind of convolutional neural network (CNN) or deep convolutional common applications include picture recognition and categorization. The approach had been dormant since the 1990s, but in 2012 it saw a resurgence and is now widely used in several subfields of computer vision. Besides its numerous hidden layers, DCNN mimics the brain's image-recognition capabilities. It employs several locally linked layers capable of automated feature identification and several completely connected layers for classification, allowing for feature extraction rather than human feature creation [20]. There are specialized neural networks that perform the extraction function. To be determined by means of exercise and subsequent weight updates. DCNN automates the formerly laborious process of manually extracting features. Layers of a DCNN are responsible for different tasks inside the network [21]. There will be examples of the most typical layers' structure and function:

As its name suggests, the convolutional layer (conv) is built from a series of filters that may be trained individually. The weights were chosen at random and then trained using a backpropagation technique. The feature map is the output of a filter, the learnt weight, which is convolved with the whole picture [6]. Finding out what makes the original input picture special is what the features map is for. The equation is used to calculate the feature map:

$$Z^s = f(\sum_{t=1}^q W_t^s * X^i + b_s) \quad (1)$$

In deep convolutional neural networks (DCNNs), the Rectified Linear function (ReLU) is often used [22]. Such layers include max-pooling layers, which seek the maximum value in a constrained area of the output units of a convolutional feature map (2 2 or 3 3). This leads to a more condensed portrayal of the features. Using a pooling layer in a CNN may help you save time and memory by discarding features that aren't vital to the model. It is possible to express the maximum pooling operation:

$$y_{j,k}^i = \max_{0 \leq m, n < s} (x_{j,s+m, k, s+n}^i) \quad (2)$$

where, $y_{j,k}^i$ is the i^{th} neuron's activation map output across a $a_{(s,s)}$ non-overlapping local area, and $x_{j,k}^i$ is the i^{th} neuron's input map. The output of one layer of pooling or convolution feeds into the input of the following fully connected layer, just as it does in a traditional neural network. The last layer of a deep convolutional neural network (DCNN) is a SoftMax layer, which is responsible for classifying the features retrieved in earlier layers. Typically, the probabilities of belonging to a class are produced as a sum of logits, which is then used as an indication of membership. To convert a vector of random real numbers into a set of probabilities, the SoftMax function is used.

$$z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} \begin{cases} 1.1 \rightarrow \\ 2.2 \rightarrow \\ 0.2 \rightarrow \\ -1.7 \rightarrow \end{cases} s_i = \frac{e^{x_i}}{\sum_i e^{x_i}} \begin{cases} \rightarrow 0.224 \\ \rightarrow 0.672 \\ \rightarrow 0.091 \\ \rightarrow 0.013 \end{cases} \quad (3)$$

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}$$

By using an exponential function in the above calculation, we know that the results cannot be negative. The denominator normalization term causes the derived numbers to add up to 1. In addition, every number is between zero and one. The rank order of the input values is maintained, which is a useful characteristic of the SoftMax function:

$$-1.7 < 0.2 < 1.1 < 2.2 \rightarrow 0.013 < 0.091 < 0.224 < 0.672 \quad (4)$$

In mathematical terms, the SoftMax function is what's known as a "vector function," meaning that it takes a vector as input and returns another vector as output:

$$\text{softmax} : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (5)$$

Therefore, we refer to the SoftMax function's Jacobian matrix, which contains all the first-order partial derivatives, when discussing its derivative:

$$J_{\text{softmax}} = \begin{pmatrix} \frac{\partial s_1}{\partial z_1} & \frac{\partial s_1}{\partial z_2} & \dots & \frac{\partial s_1}{\partial z_n} \\ \frac{\partial s_2}{\partial z_1} & \frac{\partial s_2}{\partial z_2} & \dots & \frac{\partial s_2}{\partial z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial s_n}{\partial z_1} & \frac{\partial s_n}{\partial z_2} & \dots & \frac{\partial s_n}{\partial z_n} \end{pmatrix} \quad (6)$$

Here,

$$s_i = \frac{e^{z_i}}{\sum_{l=1}^n e^{z_l}}, \quad \forall i = 1, \dots, n \quad (7)$$

To see how the SoftMax function's outputs are contingent on the input values, just look at the SoftMax's inputs and output (due to the denominator). Therefore, the Jacobian's off-diagonal elements aren't equal to zero. Since the SoftMax function returns only positive numbers, we may use the following method to drastically shorten the derivation: We compute the logarithmic derivative (also known as the "partial derivative of the logarithm of the output") rather than the "partial derivative of the output".

$$\frac{\partial}{\partial z_j} \log(s_i) = \frac{1}{s_i} \cdot \frac{\partial s_i}{\partial z_j} \quad (8)$$

In this case, the right-hand phrase is a natural consequence of the chain rule. We then shuffle the top formula around to get:

$$\frac{\partial s_i}{\partial z_j} = s_i \cdot \frac{\partial}{\partial z_j} \log(s_i) \quad (9)$$

The term on the left is the desired partial derivative. Because of the right-hand side, we may calculate the derivative without resorting to the quotient rule of derivatives, as we shall show in a moment. The logarithm of s is a necessary initial step:

$$\log s_i = \log \left(\frac{e^{z_i}}{\sum_{l=1}^n e^{z_l}} \right) = z_i - \log \left(\sum_{l=1}^n e^{z_l} \right) \quad (10)$$

The partial derivative of the resulting expression is:

$$\frac{\partial}{\partial z_j} \log s_i = \frac{\partial z_i}{\partial z_j} - \frac{\partial}{\partial z_j} \log \left(\sum_{l=1}^n e^{z_l} \right) \quad (11)$$

Let's have a look at the first term on the right-hand side:

$$\frac{\partial z_i}{\partial z_j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where, 1 is an indicator function that allows for compact expression. If its input is true, the indicator function returns 1, and else it returns 0. You may use the chain rule to determine the value of the second term on the right-hand side:

$$\frac{\partial}{\partial z_j} \log s_i = 1\{i = j\} - \frac{1}{\sum_{l=1}^n e^{z_l}} \cdot \left(\frac{\partial}{\partial z_j} \sum_{l=1}^n e^{z_l} \right) \quad (13)$$

For the preceding process, we tapped into the natural logarithm's derivative:

$$\frac{d}{dx} \log(x) = \frac{1}{x} \quad (14)$$

Obtaining the partial derivative of the sum is trivial:

$$\frac{\partial}{\partial z_j} \sum_{l=1}^n e^{z_l} = \frac{\partial}{\partial z_j} [e^{z_1} + e^{z_2} + \dots + e^{z_j} + \dots + e^{z_n}] = \frac{\partial}{\partial z_j} [e^{z_j}] = e^{z_j} \quad (15)$$

Plugging the result into the formula yields:

$$\frac{\partial}{\partial z_j} \log s_i = 1\{i = j\} - \frac{e^{z_j}}{\sum_{l=1}^n e^{z_l}} = 1\{i = j\} - s_j \quad (16)$$

When all is said and done, the upper expression must be multiplied by s, as was demonstrated at the outset of this subsection:

$$\frac{\partial s_i}{\partial z_j} = s_i \cdot \frac{\partial}{\partial z_j} \log(s_i) = s_i \cdot (1\{i = j\} - s_j) \quad (17)$$

Our deduction process is finished at this point. As a result, we can now calculate any part of the Jacobi matrix given its other parts. When we restrict to the case where n is 4, we obtain [23, 24]:

$$J_{\text{softmax}} = \begin{pmatrix} s_1 \cdot (1 - s_1) & -s_1 \cdot s_2 & -s_1 \cdot s_3 & -s_1 \cdot s_4 \\ -s_2 \cdot s_1 & s_2 \cdot (1 - s_2) & -s_2 \cdot s_3 & -s_2 \cdot s_4 \\ -s_3 \cdot s_1 & -s_3 \cdot s_2 & s_3 \cdot (1 - s_3) & -s_3 \cdot s_4 \\ -s_4 \cdot s_1 & -s_4 \cdot s_2 & -s_4 \cdot s_3 & s_4 \cdot (1 - s_4) \end{pmatrix} \quad (18)$$

See how the diagonal elements differ from the off-diagonal elements.

3.2 IRIS recognition system

The iris recognition system's full name will be a compilation of its numerous individual approaches, which include but are not limited to the following: The iris may be used as an image input device (Not isolation from pupil); the ability to read irises and pupils; Extraction of features for categorization and Harmonizing. The following are the components (as shown in Figure 1) that would make up the iris recognition system that is being proposed: Initially, the input photo is processed so that the iris can be distinguished from the rest of the eye. Subsequently, the feature is extracted and categorised. Iris scanning, also known as iris recognition, involves capturing a high-contrast image of an individual's iris with the use of both visible and near-infrared light. Similar to fingerprinting and facial recognition, it is a kind of biometric technology. Supporters of iris scanning technology say it helps police identify criminals by comparing suspects' irises to a database of known identities. It's simpler to hide or change fingerprints than it is to change one's eyes, therefore iris scans are more secure and faster, they say.

There are serious privacy and civil rights issues that arise with iris scanning. Iris scanning may be feasible at a distance or even while the subject is in motion, opening the door to covert data collection without the subject's awareness or agreement. There are also privacy and security issues to consider; for example, if a database containing biometric information is stolen or compromised, a new pair of eyes cannot be supplied in the same way that a stolen credit card number can. Furthermore, third-party providers often acquire

and retain iris biometrics, which further exacerbates the aforementioned security issue.

The irises, the colourful circles in the back of each person's eye, are scanned in order to determine their individual patterns. Scanners that use biometric iris recognition shine infrared light on the eye, lighting the iris so that it can detect the distinctive patterns inside the iris that are otherwise imperceptible to the human sight. Iris scanners are able to identify and filter out potential obstructions to the iris, such as eyelashes, eyelids, and specular reflections. In the end, we get a collection of pixels that exclusively represent the iris. Then, a bit pattern representing the information in the iris is deduced by analysing the eye's underlying structure of lines and colours. To confirm its authenticity (by a one-to-one template match) or to identify it, this bit pattern is digitalized and compared to a database of templates (one-to-many template matching). Iris scanning cameras may be either stationary (placed on a wall) or mobile (carried about). The long-range scanners being developed by Carnegie Mellon University researchers might be used to take photos invisibly from as far as 40 feet away.

3.3 Data pre-processing

The primary goal of this initial stage of processing is to isolate authentic regions of iris texture from a backdrop consisting of noise. Iris segmentation's primary objective is to isolate the iris from the rest of the eye picture, including the lid and the lashes, so that it may be used as input for feature extraction. This allows the iris to be evaluated on its own merits. A median filter and histogram equalization [13] are used to extract the iris and the pupil, respectively. There is a possibility that blurring the picture using a disc filter and then performing gamma correction will be of assistance in detecting the inner border (iris-pupil). After the picture has been turned into binary with the help of Otsu global thresholding, the CHT method may be used to locate the centre and the perimeter of the iris [17]. The CHT algorithm is used to extract the iris' radius from the pre-processed photos, and the Canny edge detection algorithm is used to find the iris' edge. The iris-sclera boundary may thus be seen more clearly because of this.

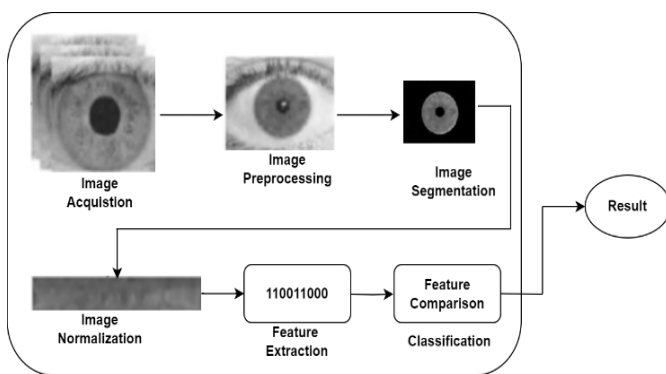


Figure 1. Typical stages of Iris acknowledgement system

The methodology for building an iris recognition system using deep convolutional neural networks (CNNs) involves designing a CNN architecture with convolutional layers, ReLU activation, pooling layers, fully connected layers, and a SoftMax layer for classification. Hyperparameters such as learning_rate =0.00, batch_size =32, epochs =50 are set, and the training process includes dataset preparation, model initialization, loss function selection (cross-entropy), optimizer

usage (Adam), and monitoring validation loss for overfitting. Data augmentation strategies like rotation_range=20, width_shift_range=0.1, height_shift_range=0.1, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, vertical_flip=True, fill_mode='nearest' are employed to increase dataset diversity.

3.4 Dataset description

This database is built upon ITT Delhi 0.1. Except for people 1-13, 27, 55, and 65, who only have left eye photographs, the database contains a total of 10 images for everyone. Persons between the ages of 14 and 15. 167 men and 48 females make up the total population. The photographs are bitmap format, 320×240 in size, and were captured in a secure location away from prying eyes. Figure 2 depicts a database sample.

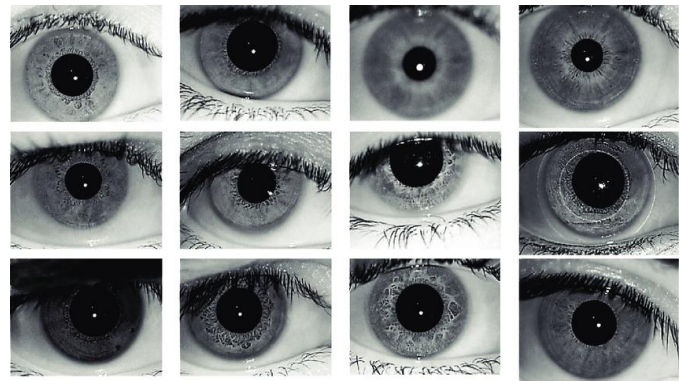


Figure 2. Samples of the IRIS database

3.5 Model' s architecture

In this research, we provide a method for automatically identifying and categorizing iris characteristics. The proposed model's structure is comprised of 4 Convolution layers, 6 activation ReLU levels, 3 Pooling layers, and 2 completely connected layers; these components work together to automatically categorize pictures and extract features without requiring domain expertise. The IRISNet architecture consists of 18 separate levels, as seen in Figure 3. Some recommended layer orderings include Convolution, Rule, Pooling, FC, and SoftMax. To prevent overfitting, a dropout layer is added at the end of each fully connected (FC) layer. Thus, there is a blockade on global updates to neuronal weights. With this layer in place, neurons won't all be firing in the same direction. A separate "development" (D) set, and "evaluation" (T) set were created from the original data (T set). Training and validation are now two separate aspects of the design process. Some of the data used in the design process is kept aside for the sole purpose of monitoring its performance via the use of dedicated training and validation sets of data. Design phase (D set) photographs may be utilised to guide test phase (T set) photo categorization once training is complete (D set). CNN's default weights are a random selection. It's plain to see that it leads nowhere good. When training a deep neural network, one of the most important principles is to begin with a network that is not very effective and work our way up to one that is very precise. After all training is done, the loss function should be as little as possible. The weight is updated based on the learning rate. Different strategies are utilized to find the best possible loss function optimizations. Possible or unlikely application of gradients in the relevant algorithms. Adam is

shorthand for an extremely simple gradient-based method [23] as reported by study [24]. In order to train the network, we use the backpropagation technique [25] and update the weights using Adam. Weights are updated using mini-batch 20; this technique divides the training data into 20-element "mini-batches" and computes errors in the SoftMax layer before back-propagating them to the lower layers. Any number of epochs may be selected at random, and their internal iteration counts can be determined with the help of the formula shown below [26, 27]:

$$num_{iterations} \cong \frac{number\ of\ training\ labels}{size\ of\ the\ mini\ batch} \quad (19)$$

After each iteration through an epoch, the current configuration's efficacy was evaluated in relation to the validation set. Once training is complete, the testing set is used to evaluate the network's performance. Training of the convolutional layer is shown in Figure 3.

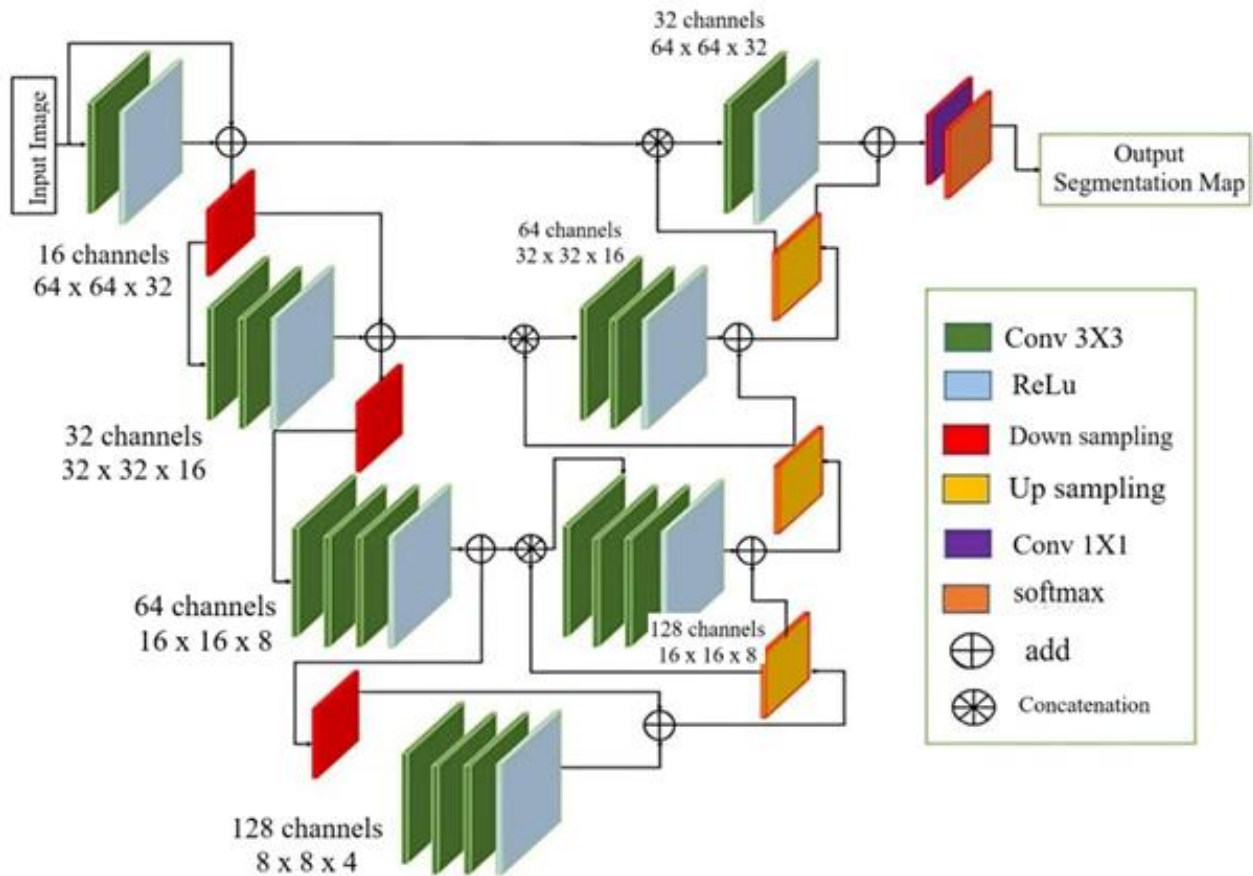


Figure 3. Model's architecture

3.6 Model's optimizer and LR scheduler

In our paper, we used modified version of Adam optimizer. The choice of an optimizer is one of the most crucial choices while developing a deep learning model. It can immensely affect the learning of the model and can estimate whether we get to see positive outcomes in a matter of minutes, hours, or days. The Adam optimizer is undoubtedly one of the most used and well-known optimization algorithms in the field of deep learning. It can be viewed as an extension of stochastic gradient descent and was developed by Diederik Kingma and Jimmy Ba from OpenAI and University of Toronto respectively. Adam optimizer offers several advantages over its counterparts: straightforward to implement; requires very little memory; works well on problems with noisy and sparse gradients; works well with huge amounts of data; often requires very less hyperparameter tuning. Adam combines the benefits of two other variants of stochastic gradient descent namely, the AdaGrad algorithm and the RMSProp algorithm where $params$ = Iterable of parameters to optimize; lr = learning rate; $betas$ =

coefficients used to calculate running averages of gradients and its square; $epsilon$ = denominator which we must improve; $weight\ decay$ = commonly known as L2 penalty.

In addition to this, we have made use of Lambda's Learning Rate scheduler. A learning rate schedule is a predetermined framework that modifies the learning rate as the training continues between epochs or iterations. The two most popular methods for learning a rate schedule are: Constant learning rate: it implies, throughout training we initialize a learning rate and do not adjust it and Decay in learning rate: we choose a starting learning rate, then progressively lower it in line with a scheduler. The most fundamental and often used methods for reducing the learning rate during training are those that gradually slow down learning. These offer the benefit of adjusting weights significantly early in the training phase when larger learning rate values are being employed, and then reducing the learning rate such that smaller training updates are applied to weights later in the training process when the learning rate is reduced. This causes one to pick up good weights immediately and then fine-tune them afterwards. The

following are two well-liked and simple learning rate schedules: The learning rate should be gradually reduced dependent on the epoch, with huge, punctuated dips occurring at epochs.

$$lr_{epoch} = lr_{initial} * Lambda(epoch) \quad (20)$$

Algorithm1: Modified Adam Algorithm
(params, lr, betas, epsilon, weight_decay)

```

state ←HashMap()
group ←HashMap()
if lr ≤0.0 then
    lr ←1e-3
    group['lr'] =lr
end if
if betas[0] ≤0.0 or betas[1] ≤0.0 then
    betas[0] ← 0.9
    betas[1] ← 0.999
end if
group['betas'][0]= betas[0]
group['betas'][1]= betas[1]
if epsilon ≤0.0 then
    epsilon ←1e-8
end if
group['epsilon'] =epsilon
if weight_decay ≤0.0 then
    weight_decay ←0
end if
group['weight_decay'] =weight_decay
loss ←NaN
for pm in params do
    gradient =pm.gradient.data
    state =state[pm]
    if state.length() ==0 then
        state['step'] ←0
        state['exp_average']←ensor(pm.data)
        state['exp_average_sq']←tensor(pm.data)
    end if
    exp_average = state['exp_average']
    exp_average_sq=state['exp_average_sq']
    beta_one =group['betas'][0]
    beta_two =group['betas'][1]
    state ['step']← state['step']+1
    if group['weight_decay']≠0 then
        gradient←(group['weight_decay'], pm.data)
    end if
    exp_average=element_wise_multiplication(exp_average,
beta_one)+(1-beta_one)*gradient
    exp_average_sq=element_wise_multiplication(exp_averag
e_sq, beta_two)+(1-beta_two)*(gradient*gradient)
    denoms=exp_average_sq.sqrt()+group['epsilon']
    bias_correction_one=1/(1-beta_one**state['step'])
    bias_correction_two=1/(1-beta_two**state['step'])
    adapted_learning_rate=group['lr']*bias_correction_one /
    bias_correction_two.sqrt()
    pm.data=pm.data-
    adapted_learning_rate*exp_average / denoms
end for
return loss
end procedure

```

3.7 Model’s criterion

Criteria or loss functions are basically used to optimize the

model during training. The categorical cross entropy loss function, which is employed in classification issues with two or more classes, is what we used in our research. Cross-entropy is a popular loss function in machine learning. Building on entropy, the information theory measurement known as cross-entropy essentially calculates the distinction between 2 or more probability distributions. Cross-entropy can be thought of as calculating the total entropy linking the distributions, whereas KL divergence calculates the relative entropy linking two statistically probable distributions. The two are closely related but distinct from one another. This term makes sense if we think of a goal or fundamental probability distribution A and a distribution that approximates it as B. The cross-entropy of A from B is the number of additional bits needed to record an event using B rather than A. Formally, the cross-entropy between 2 probability distributions, like B from A, is expressed as: $H(A, B)$. Here, A might be the target distribution, B is an estimate of the target distribution, and $H()$ is the cross-entropy function.

The probability of the occurrences from A and B can be used to determine cross-entropy as follows:

$$H(A, B) = - \sum x \ln XA(x) * \log(B(x)) \quad (21)$$

where, $A(x)$ is the probability of event x occurring in A , $B(x)$ is the probability of x occurring in N , and \log is the base 2 log, the results are in bits. Although a comparable calculation may be made for continuous probability distributions by using the integral across the occurrences rather than the sum, this computation is for discrete probability distributions. If the two probability distributions are the same, the outcome will be a positive number expressed in bits that fit for the entropy of the distribution.

Let *Frobenius product* be $a : b = a^T b$ the *Hadamard product* be $a \odot b$ and *Hadamard division* be $\frac{a}{b}$ here \log function will be applied *elementwise*, and our modified *criterion* be:

$$L = -mJ \quad (22)$$

The *differential* and *gradient* of L will be calculated as:

$$L = y : \log(\hat{y}) + (1 - y) : \log(1 - \hat{y}) \quad (23)$$

$$dL = y : d \log(\hat{y}) + (1 - y) : d \log(1 - \hat{y}) \quad (24)$$

$$dL = \frac{y}{\hat{y}} : d\hat{y} + \frac{1 - y}{1 - \hat{y}} : d(1 - \hat{y}) \quad (25)$$

$$dL = \left(\frac{y}{\hat{y}} - \frac{1 - y}{1 - \hat{y}} \right) : d\hat{y} \quad (26)$$

$$dL = \left(\frac{y - \hat{y}}{\hat{y} - \hat{y} \odot \hat{y}} \right) : d\hat{y} \quad (27)$$

$$\frac{\partial L}{\partial \hat{y}} = \left(\frac{y - \hat{y}}{\hat{y} - \hat{y} \odot \hat{y}} \right) \quad (28)$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{y}{\hat{y}} : d\hat{y} + \frac{1 - y}{1 - \hat{y}} : d(1 - \hat{y}) \quad (29)$$

$$\frac{\partial L}{\partial \hat{y}} = \left(\frac{y}{\hat{y}} - \frac{1 - y}{1 - \hat{y}} \right) : d\hat{y} \quad (30)$$

$$\frac{\partial L}{\partial \hat{y}} = \left(\frac{y - \hat{y}}{\hat{y} - \hat{y} \odot \hat{y}} \right) : d\hat{y} \quad (31)$$

$$\frac{\partial L}{\partial \hat{y}} = \left(\frac{y - \hat{y}}{\hat{y} - \hat{y} \odot \hat{y}} \right) \quad (32)$$

Thus, the gradient of the original criterion will be:

$$\frac{\partial J}{\partial \hat{y}} = -\frac{1}{m} \frac{\partial L}{\partial \hat{y}} = \frac{\hat{y} - y}{m(\hat{y} - \hat{y} \odot \hat{y})} \quad (33)$$

4. RESULTS

The accuracy of a machine learning model is the yardstick by which its superiority in spotting correlations and patterns between variables in a dataset based on the input data, also known as training data, can be measured. Greater accurate predictions and insights mean more value for the organization, and this is only possible if the model can generalize to 'unseen' data successfully. Companies employ machine learning models to make better, more informed decisions in light of the increased realism the models provide. The cost of mistakes may be quite high, but the cost can be reduced by improving the accuracy of the model. There is, of course, a threshold of diminishing returns at which the value of constructing a more accurate model will not result in a proportionate gain in profit, but in many cases, it is helpful to the whole organization. For instance, a hospital and a patient will both incur expenses associated with a false positive cancer diagnosis. The advantages of enhancing model accuracy assist prevent wasting significant amounts of time and money while also relieving unnecessary tension. The concept of "good accuracy" in machine learning is open to interpretation. But according to our standards, a fantastic performance for a model is anything that is more than 70%. An accuracy measurement of anything between 70 and 90%, inclusive, is not only desirable but also attainable.

The sensitivity of a machine learning model is defined as its ability to identify high-quality instances. The "true positive rate" (or recall) is a term used in specific contexts. Sensitivity is an important metric for evaluating a model's efficacy since it shows us the proportion of true positives it identified. Models with high sensitivity also tend to miss some positive events since they have a low false negative rate. Alternatively stated, sensitivity is a model's ability to detect positive instances with precision. This is crucial since accurate forecasting relies on our models being able to identify every possible positive scenario. The sum of sensitivity, also known as the rate of true positives, and specificity, sometimes known as the rate of false negatives, would equal 1. The greater the genuine positive rate, the more accurately the model can detect positive situations; conversely, a lower rate indicates that the model is less accurate. Let's try to comprehend this using the model that is used for determining whether a person is afflicted with the ailment. The sensitivity rate, also known as the true positive rate, is a measurement that determines the percentage of persons who are afflicted with the illness who were accurately anticipated as being the ones afflicted with the condition. To put it another way, the person who is unwell and has a good attitude was in fact correct in their prediction that they are unhealthy. A high sensitivity indicates that the model is successfully detecting most positive findings, while a low sensitivity indicates that the model is failing to recognise a significant number of positive

results.

It is common practice to do a comparison between sensitivity and specificity when assessing the performance of a model. The fraction of genuine negatives that are accurately detected by the model is what we mean when we talk about specificity. This indicates that there will be another fraction of true negatives that were forecasted as positives and may be referred to as false positives. This is because of the previous point. Additionally, one may refer to this fraction as a True Negative Rate (TNR). The total of the specificity, or the rate of genuine negatives, and the sensitivity, or the rate of false positives, would always equal 1. A high specificity indicates that the model is accurately recognizing most of the outcomes that were expected to be negative, while a low specificity indicates that the model is incorrectly classifying a significant number of expected negative findings as expected positive ones. Let's try to comprehend this using the model that is used for determining whether a person is afflicted with the ailment. Specificity is a measure of the percentage of individuals who are not afflicted with the illness who were accurately anticipated as being individuals who are not afflicted with the disease. To put it another way, specificity refers to the percentage of people who truly match the health status that was predicted for them.

Recall and Precision are also important metric because, when we are constructing any machine learning model, the first thing that comes to our minds is how we can construct an accurate and good fit model and what the challenges are that will come up during the process. Both questions are important because they help us anticipate what will happen next. Precision and Recall are two of the most fundamental ideas in Machine Learning, although they may be difficult to understand. Precision and recall are performance measures that are used in machine learning for the purposes of pattern identification and categorization. These ideas are fundamental to the process of developing a perfect model for machine learning, which yields outcomes that are more exact and accurate. The machine learning models may be broken down into two categories: those that demand higher accuracy and those that require more recall. Therefore, it is essential to have a good understanding of the trade-off between accuracy and recall, sometimes known as the precision-recall balance.

$$Precision = \frac{True\ Positive}{Total\ Predicted\ Positives} \quad (34)$$

$$Recall = \frac{True\ Positive}{Total\ Actual\ Positives} \quad (35)$$

In mathematics, the value that is referred to as the "mean" of a group of numbers is the value that is "central" or "average" among the numbers in the group. In a similar manner, the geometric mean is the value that is in the middle of a collection of numbers and may be calculated by finding the root of the nth degree of the product of each of the n values in the set. Since calculating the geometric mean entails finding the nth root of a number, this method can only be used to sets of positive integers. This is a crucial fact to keep in mind. Following is the mathematical definition of the geometric mean: where n is equal to the total number of values in the set, and x is a specific number from inside the set

$$\left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \dots x_n} \quad (36)$$

The F-measure presents a method for combining accuracy and recall into a single measure that can capture both features. When taken by themselves, neither accuracy nor memory reveal the complete story. It's possible for us to have horrible accuracy but fantastic memory, or it's also possible for us to have excellent recall but poor precision. The F-measure offers a method for expressing both issues via the use of a single score. After the scores for accuracy and recall have been determined for a binary or multiclass classification issue, it is possible to combine the two values to get the F-measure for the problem. The two fractions have been averaged to get this harmonic mean. This is referred to as the F-score or the F1-score on occasion, and it is possible that it is the measure that is used the most often for unbalanced categorization issues. When learning from data that is unbalanced, the option that is employed the most often is the F1-measure, which gives equal weight to both accuracy and recall. When working on classification models in which your data set is unbalanced, such as in IRIS segmentation, the F1-score becomes extremely useful because of its increased predictive power.

$$F1\ Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (37)$$

Table 2 presents the performance metrics of different classifiers for iris recognition, including accuracy, sensitivity, specificity, precision, recall, G-mean, and F1-score. Here's a discussion of these metrics:

Accuracy: This metric measures the overall correctness of the classifier's predictions. A higher accuracy indicates that the classifier is making more correct predictions overall. Among the classifiers listed, IRISNet has the highest accuracy of 0.9634, indicating that it has the best overall performance in terms of correct predictions as shown in Figure 4.

Sensitivity: Also known as True Positive Rate or Recall, sensitivity measures the proportion of actual positive cases correctly identified by the classifier. In this case, all classifiers

have a sensitivity of 1.0000, indicating that they correctly identify all positive cases, which is a desirable trait, especially in biometric recognition systems.

Specificity: Specificity measures the proportion of actual negative cases correctly identified by the classifier. A high specificity indicates that the classifier can effectively distinguish negative cases. IRISNet and Support Vector Machine (SVM) have the highest specificity among the listed classifiers.

Precision: Precision measures the proportion of true positive predictions among all positive predictions made by the classifier. It reflects the classifier's ability to avoid false positives. IRISNet has the highest precision among the listed classifiers, indicating that it makes fewer false positive predictions.

G-Mean: The geometric mean (G-mean) combines sensitivity and specificity into a single metric. It is particularly useful for imbalanced datasets where the number of negative cases outweighs the positive cases. IRISNet has the highest G-mean, indicating a good balance between sensitivity and specificity.

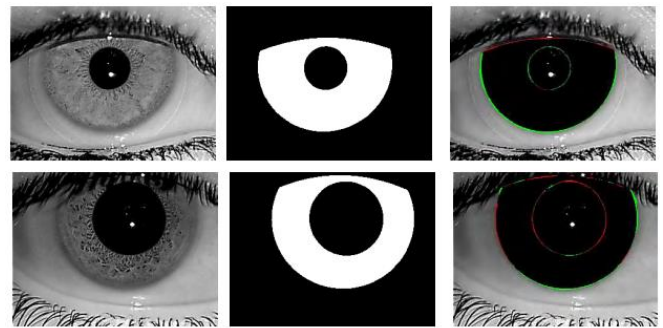


Figure 4. Segmentation results: - Left: Original image, Middle: Image mask, Right: Segmented image

Table 2. Evaluation report

Classifier	Model Accuracy	Model Sensitivity	Model Specificity	Model Precision	Model Recall	Model G-Mean	Model F1-Score
IRISNet	0.9634	1.0000	0.9583	0.1247	1.0000	0.9727	0.2217
K-Nearest Neighbour	0.9287	1.0000	0.9226	0.0623	1.0000	0.9657	0.1172
Decision Tree	0.4128	1.0000	0.4037	0.0168	1.0000	0.6438	0.0330
Support Vector Machine	0.9026	1.0000	0.9023	0.0423	1.0000	0.9486	0.0811
Naïve Bayesian Algorithm	0.9128	1.0000	0.9103	0.0512	1.0000	0.9521	0.0974
[16]	0.9593	0.9736	0.9145	NA	NA	NA	NA

F1-Score: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is useful when there is an uneven class distribution. IRISNet has the highest F1-score, indicating a good balance between precision and recall compared to other classifiers.

An error analysis of the iris recognition models presented in the table is crucial for understanding their limitations and performance gaps. Starting with IRISNet, despite achieving a high overall accuracy of 96.34%, the model's precision is notably low at 12.47%, indicating a high false positive rate. This suggests that while the model correctly identifies most positive cases (sensitivity of 100%), it also misclassifies a significant number of negative cases as positive. On the other hand, K-Nearest Neighbour demonstrates a lower overall accuracy of 92.87% but maintains a higher precision of 6.23%. However, the decision tree model exhibits significantly lower

performance across all metrics, with an accuracy of 41.28% and a precision of 1.68%. Support Vector Machine and Naïve Bayesian Algorithm show better overall accuracy and precision compared to the decision tree but still lag behind IRISNet and K-Nearest Neighbour. An in-depth error analysis would delve into the specific cases where these models fail, such as misclassifications between similar iris patterns or outliers that challenge the classifiers' generalization abilities.

The results of the classifier performance analysis for iris recognition highlight several key insights and challenges. Starting with IRISNet, the model exhibits a high level of accuracy and sensitivity, indicating its ability to correctly identify positive cases with a low rate of false negatives. However, the notably low precision of IRISNet raises concerns about its false positive rate, suggesting potential misclassifications of negative cases as positive. This trade-off

between sensitivity and precision is a common challenge in biometric recognition systems and underscores the need for models that can strike a better balance between correctly identifying positives while minimizing false positives.

In contrast, the K-Nearest Neighbour model demonstrates a higher precision compared to IRISNet, indicating a reduced false positive rate at the cost of a slightly lower overall accuracy. This trade-off suggests that K-Nearest Neighbour may be more suitable for applications where minimizing false positives is crucial, such as security systems. However, its performance could still be improved to achieve a better balance between sensitivity and precision.

The decision tree model, on the other hand, shows significant limitations across all metrics, with poor accuracy, specificity, precision, and F1-score. This indicates challenges in correctly classifying iris patterns, potentially due to the model's simplistic decision-making process or lack of robustness in handling complex data patterns.

Both Support Vector Machine (SVM) and Naïve Bayesian Algorithm exhibit moderate overall performance but still struggle with relatively low precision, indicating room for improvement in reducing false positives. These models may benefit from further optimization or feature engineering to enhance their classification capabilities.

The comparison of these classifiers aligns with existing literature, highlighting the ongoing challenges in achieving a balance between sensitivity and precision in iris recognition systems. Future research efforts could focus on developing hybrid models or incorporating advanced techniques like deep learning to improve classification accuracy and mitigate false positives effectively. Additionally, addressing dataset imbalances and refining evaluation metrics will be essential for advancing the performance and reliability of iris recognition models in real-world applications.

Limitations and Challenges:

- **Imbalanced Datasets:** The presence of imbalanced datasets can skew performance metrics, especially for precision and specificity.
- **Model Complexity:** Certain classifiers may exhibit varying performance based on dataset complexity and feature representation.
- **Evaluation Metrics:** While the selected metrics provide valuable insights, additional metrics like ROC-AUC could offer a more comprehensive assessment.
- **Generalization:** The study's findings may be specific to the iris recognition dataset used and may not generalize universally across all iris recognition scenarios.

5. DISCUSSION AND CONCLUSION

One kind of AI that is becoming dominant across fields is the deep convolutional neural network (DCNN). Features are automatically learned via deep learning. Deep learning algorithms are widely used in pattern recognition applications, some examples include:

- 1) **Image Recognition:** Deep learning algorithms are used to train models that can recognize objects, people, and scenes in images and videos. For example, a deep learning model trained to recognize objects in images can be used in applications such as self-driving cars, to detect pedestrians and other vehicles on the road.

- 2) **Speech Recognition:** Deep learning algorithms are used to train models that can recognize and transcribe speech. These models are used in applications such as voice assistants, speech-to-text dictation, and speech-enabled devices like smart speakers.
- 3) **Handwriting Recognition:** Deep learning algorithms are used to train models that can recognize and transcribe handwritten text. These models can be used in applications such as digital ink recognition, signature verification, and document analysis.
- 4) **Object Detection:** Deep learning algorithms are used to train models that can detect and classify objects in images and videos. These models can be used in applications such as surveillance systems, self-driving cars, and robotics.
- 5) **Face Recognition:** Deep learning algorithms are used to train models that can recognize and identify people in images and videos. These models can be used in applications such as security systems, social media, and mobile devices.
- 6) **Natural Language Processing:** Deep learning algorithms are used to train models that can understand, interpret, and generate human language, such as text and speech. These models can be used in applications such as language translation, text summarization, and sentiment analysis.

These are just a few examples of the many pattern recognition applications that make use of deep learning algorithms. As the technology continues to evolve, deep learning is expected to be used in an increasing number of applications in the future. Therefore, give a thorough grasp of the data without relying on the extraction of created characteristics. There are several forms of biometric identification that are the most secure and dependable:

- 1) **Iris recognition:** Iris recognition is considered to be one of the most robust and dependable forms of biometric identification. The iris, the colored ring around the pupil of the eye, is unique to each individual and does not change over time, making it a reliable method for identifying a person.
- 2) **Facial recognition:** Facial recognition uses a 3D or 2D image of a person's face to identify them. It is considered a secure form of biometric identification because the face is a unique and unchanging feature, and facial recognition systems can also take into account the facial structure, skin texture and other features.
- 3) **Fingerprint recognition:** Fingerprint recognition is based on the unique patterns of ridges and valleys on the fingerprints. It is considered a reliable form of biometric identification because fingerprints are unique to each individual and do not change over time.
- 4) **DNA recognition:** DNA recognition is based on the unique genetic code of an individual. It is considered to be the most secure form of biometric identification, as DNA is unique to each individual and does not change over time.

All these biometric methods have been extensively tested and evaluated and have been found to be highly accurate and dependable. However, it's important to note that no single biometric method is 100% foolproof, and it is always recommended to combine multiple biometric methods for more secure identification. In summary, iris recognition, facial

recognition, fingerprint recognition, and DNA recognition are some of the most secure and dependable forms of biometric identification. They are based on unique and unchanging physical characteristics and have been extensively tested and evaluated for their accuracy and dependability. In this study, we introduce a powerful network for extracting features from and classifying iris images. The iris is first isolated by pre-processing, and then the Convolution, Pooling, ReLU, Dropout, Fully Connected, and SoftMax layers of a DCNN are used to extract features useful for IITD iris classification. The performance of the trained IRISNet is tracked over time using a specific set of evaluation criteria. The use of deep learning techniques, and convolutional neural networks (CNN), has proven very effective in many computer vision (CV) applications. CNN's capacity to automatically learn relevant features from adequate training data has made it superior to handcrafted feature extraction methods. CNN's potential for use in iris image processing has been explored in recent years, leading to improvements in iris segmentation, identification, and false iris detection. Iris identification is a good example of a real-world application of deep learning for several reasons:

- 1) Image Processing: Iris identification systems use deep learning to process and analyze images of the iris. This is done by using convolutional neural networks (CNN) to extract features from the iris images, such as the number, size, and location of furrows and ridges. This is a common approach in deep learning, CNN are good at identifying patterns in images and videos.
- 2) High-Security Applications: Iris identification systems are used in high-security applications such as border control, access control, and law enforcement. Iris recognition is considered to be one of the most secure forms of biometric identification because the iris is unique to each individual and does not change over time.
- 3) Handling Large Amounts of Data: Iris identification systems can handle large amounts of data, which is a characteristic of deep learning. Iris recognition systems can process large numbers of iris images and extract unique features from them.
- 4) High Accuracy: Deep learning models have been shown to achieve high accuracy in iris recognition, which makes it a reliable method for identification. This is due to the ability of deep learning models to automatically learn features from the data without manual feature engineering.
- 5) Real-time Processing: Deep learning models can be optimized to perform in real-time, which makes iris recognition systems suitable for real-world applications. This allows the system to quickly identify an individual, making it useful in high-security applications such as border control.

In summary, iris identification system is a good example of a real-world application of deep learning because it uses deep learning to process and analyze images of the iris, it's used in high-security applications, it can handle large amounts of data, it achieves high accuracy, and it can perform in real-time.

REFERENCES

- [1] Boles, W.W., Boashash, B. (1998). A human identification technique using images of the iris and wavelet transform. *IEEE Transactions on Signal Processing*, 46(4): 1185-1188. <https://doi.org/10.1109/78.668573>
- [2] Ciregan, D., Meier, U., Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, pp. 3642-3649. <https://doi.org/10.1109/CVPR.2012.6248110>
- [3] Shoniregun, C.A., Crosier, S. (2008). Securing biometrics applications. In: *Securing Biometrics Applications*, Springer, Boston, MA, pp. 113-142. https://doi.org/10.1007/978-0-387-69933-2_4
- [4] Daugman, J. (2009). How iris recognition works. In *The Essential Guide to Image Processing*, Academic Press, pp. 715-739. <https://doi.org/10.1016/B978-0-12-374457-9.00025-1>
- [5] El-Rahiem, B.A., Ahmed, M.A.O., Reyad, O., El-Rahaman, H.A., Amin, M., El-Samie, F.A. (2020). An efficient deep convolutional neural network for visual image classification. In the International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019) Springer International Publishing. Springer, Cham, 4: 23-31. https://doi.org/10.1007/978-3-030-14118-9_3
- [6] Gad, A.F., Gad, A.F., John, S. (2018). *Practical Computer Vision Applications Using Deep Learning with CNNs*. Berkeley: Apress.
- [7] Gangwar, A., Joshi, A. (2016). DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition. In 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, pp. 2301-2305. <https://doi.org/10.1109/ICIP.2016.7532769>
- [8] Gonzalez, R.C., Woods, R.E. (2007). *Digital Image Processing*. [https://books.google.co.in/books?hl=en&lr=&id=a62xQ2r_f8wC&oi=fnd&pg=PA19&dq=%5B8%5D%09Gonzalez+,+R.C.,+Woods,+R.E.+\(2007\).+Image+processing.+Digital+Image+Processing,+2:+1.&ots=3B1yQ7gI2H&sig=1gxOersqswId0oLRh9WxXr172FQ&redir_esc=y#v=onepage&q&f=false](https://books.google.co.in/books?hl=en&lr=&id=a62xQ2r_f8wC&oi=fnd&pg=PA19&dq=%5B8%5D%09Gonzalez+,+R.C.,+Woods,+R.E.+(2007).+Image+processing.+Digital+Image+Processing,+2:+1.&ots=3B1yQ7gI2H&sig=1gxOersqswId0oLRh9WxXr172FQ&redir_esc=y#v=onepage&q&f=false).
- [9] Johar, T., Kaushik, P. (2015). Iris segmentation and normalization using Daugman's rubber sheet model. *International Journal of Scientific and Technical Advancements*, 1(1): 11-14.
- [10] Kim, P. (2017). *MATLAB Deep Learning. With Machine Learning, Neural Networks and Artificial Intelligence*. Apress Berkeley, CA. <https://doi.org/10.1007/978-1-4842-2845-6>
- [11] Kingma, D.P. (2014). Adam: A method for stochastic optimization. *arXiv Preprint arXiv: 1412.6980*. <https://doi.org/10.48550/arXiv.1412.6980>
- [12] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.
- [13] Manvi, R.S.C., Singh, M. (2012). Image contrast enhancement using histogram equalization. *International Journal of Computing & Business Research*.
- [14] Minaee, S., Abdolrashidiy, A., Wang, Y. (2016). An experimental study of deep convolutional features for iris recognition. In 2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), Philadelphia, PA, USA, pp. 1-6.

- <https://doi.org/10.1109/SPMB.2016.7846859>
- [15] Moons, B., Bankman, D., Verhelst, M. (2018). *Embedded Deep Learning: Algorithms, Architectures and Circuits for Always-On Neural Network Processing*. Springer.
- [16] Nagi, J., Ducatelle, F., Di Caro, G.A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., Gambardella, L.M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, pp. 342-347. <https://doi.org/10.1109/ICSIPA.2011.6144164>
- [17] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1): 62-66.
- [18] Boles, W.W., Boashash, B. (1998). A human identification technique using images of the iris and wavelet transform. *IEEE Transactions on Signal Processing*, 46(4): 1185-1188. <https://doi.org/10.1109/78.668573>
- [19] Ciregan, D., Meier, U., Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, pp. 3642-3649. <https://doi.org/10.1109/CVPR.2012.6248110>
- [20] Shoniregun, C.A., Crosier, S. (2008). *Securing Biometrics Applications*. Springer, Boston, MA., pp. 113-142. https://doi.org/10.1007/978-0-387-69933-2_4
- [21] Daugman, J. (2009). How iris recognition works. In *the Essential Guide to Image Processing*. Academic Press, pp. 715-739. <https://doi.org/10.1016/B978-0-12-374457-9.00025-1>
- [22] Agarap, A.F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375. <https://doi.org/10.48550/arXiv.1803.08375>
- [23] Narayan, V., Daniel, A.K. (2022). CHOP: Maximum coverage optimization and resolve hole healing problem using sleep and wake-up technique for WSN. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 11(2): 159-178.
- [24] Narayan, V., Daniel, A.K. (2022). Energy efficient protocol for lifetime prediction of wireless sensor network using multivariate polynomial regression model. *Journal of Scientific & Industrial Research*, 81(12): 1297-1309. <https://doi.org/10.56042/jsir.v81i12.54908>
- [25] Narayan, V., Daniel, A.K. (2022). CHHP: Coverage optimization and hole healing protocol using sleep and wake-up concept for wireless sensor network. *International Journal of System Assurance Engineering and Management*, 13(Suppl 1): 546-556. <https://doi.org/10.1007/s13198-021-01538-5>
- [26] Narayan, V., Daniel, A.K. (2021). A novel approach for cluster head selection using trust function in WSN. *Scalable Computing: Practice and Experience*, 22(1): 1-13. <https://doi.org/10.12694/scpe.v22i1.1808>
- [27] Bharath, B.V., Vilas, A.S., Manikantan, K., Ramachandran, S. (2014). Iris recognition using radon transform thresholding based feature extraction with Gradient-based Isolation as a pre-processing technique. In 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, India, pp. 1-8. <https://doi.org/10.1109/ICIINFS.2014.7036572>