Understanding Iris Biometrics: Analysis of iris Feature patterns with CNN and Generalized Structure Tensor

Sandip S. Thorat*[1], Pravin V. Dhole[2], Santosh K. Gaikwad[3], Prof. K.V. Kale[*4]

[1*]Sandipthorat16@Gmail.Com [1]Department of Computer Science and Information Technology, [2]Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Chhatrapati sambhajinagar-431004 (MS), India
[2]pravindhole07@Gmail.Com [2]Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Chhatrapati sambhajinagar-431004 (MS), India
[3]Associate Professor, School Of Basic And Applied Science, JSPM University Pune
[4*]Vice Chancellor Dr. Babasaheb Ambedkar Technological University, Raigad Lonere

***Corresponding Authors:** Sandip S. Thorat, *Prof. K.V. Kale*
[*]Sandipthorat16@Gmail.Com [1]Department of Computer Science and Information, Dr. Babasaheb Ambedkar Marathwada University, Chhatrapatisambhajinagar-431004 (MS), India
[*]Vice Chancellor [4]Dr. Babasaheb Ambedkar Technological University, Raigad Lonere

ABSTRACT:
The field of iris recognition is widely studied due to its broad use in security contexts, such as airports and border control. This paper presents a descriptive study of an iris feature extraction approaches that utilize Convolutional Neural Networks (CNNs) and Generalized Structure Tensor (GST). We investigate how CNNs and GST techniques extract and analyze iris characteristics, comparing their effectiveness and potential combinations. The objective of our research is to gain a deeper understanding of the capabilities of these technologies and their potential impact on iris recognition systems. The experiments were carried out on our own generated KVK-R iris recognition dataset, with limited training images per class. The results look good; our models worked better than the previous ones. We also present a visualization method for identifying important regions in iris images that might significantly affect identification results. We think that a lot of other biometrics can be recognized using this method. CNN, on the other hand provide a robust approach to iris pattern recognition. By training CNN model on iris dataset, it automatically learning and detecting complicated patterns within iris images. The proposed iris recognition method was tested on two public datasets: KVK-R and CASIA-Iris-Thousand. The approach produced excellent results with high accuracy rates. Using trained models for feature extraction allowed for the recognition of both closed and open sets. The best results yet to be achieved with 97% for KVK-R and 89% for CASIA-Iris thousand are achieving state-of-the-art results.

Keywords: Iris features pattern, CNN, morphological operation, GST (Generalized structure *Tensor), Iris Recognition*

1. INTRODUCTION:

Biometric security systems use the iris to identify people, and it's important to get the right features from the iris. One of the most accurate and reliable ways to identify someone biometrically is through their iris. The iris is the colored ring around the pupil of the eye. This method uses the iris's unique patterns, which are different for each person and remains the same over time [1]. Iris recognition is well-suited for a range of security applications, such as airport screenings, border controls, and secure access systems, due to its non-invasive nature and high level of precision [2]. Improving eye recognition systems starts with accurately identifying features. Traditional methods often required segmenting and extracting features by hand, which could be time-consuming and lead to mistakes [3]. Recently, automatic feature extraction methods have become more popular by using deep learning algorithms. For iris recognition, convolutional neural networks (CNNs) work especially well because they can learn and extract complex patterns from large amounts of iris images. CNNs can easily find unique features of the iris by training on a variety of datasets. This makes recognition extremely accurate even in difficult situations [4].

In parallel, Generalized Structure Tensor (GST) approaches provide a mathematical foundation for assessing local structures in images. GST approaches can provide precise information regarding the orientation, coherence, and intensity of patterns in the iris [5]. This structural study enhances CNNs' feature extraction capabilities, providing a deeper understanding of the hidden characteristics that distinguish each iris. Integrating GST techniques with CNNs can improve the performance of iris identification systems by combining the capabilities of both approaches [6].

A Several different algorithms and implementations for iris recognition processes have been presented over the years Obtaining a high-quality iris image and precisely segmenting the iris region are two of the key challenges for standard processing operations [7,8]. Many algorithms have been developed for iris segmentation employing a variety of methods, including an integrodifferential operator and the circular Hough transform (CHT) [9, 10]. Using normal methods, the Hough transform (HT) method was first used to separate the eye into its parts. Bakshi et al. [11] and Raffei et al. [12] used an iris segmentation phase with three parts: localizing the iris region, eyelid occlusion, and eyelash occlusion. Next To find the center of the circle and get rid of the effect of eyelashes and eyebrows, the CHT method proposed by Davies was used to localize the iris and find the lower half circles [13]. As part of our study, we compare the CNN and GST methods by looking at how well they describe iris patterns respectively. For training and testing, we use iris datasets that are open to the public. Our results show that both methods work well. CNNs are great at recognizing patterns in general, and GST gives us the specific information we need for feature extraction [5]. We also look at how CNN and GST methods can be combined to make the most of each one's strengths. Putting these ideas together should help iris recognition systems work better and be more dependable. Overall, our research employing state-of-the-art techniques like CNNs and GST, our research expands on the understanding of iris pattern recognition. We demonstrate how these methods can enhance iris recognition systems and open the way for more secure and reliable biometric identification methods in various applications [14]. Since deep learning has been so successful recently, studies on iris identification have adopted numerous types of Convolutional neural networks (CNNs) to replace various phases of conventional pipelines, like segmentation and representation [15].

2. DEEP LEARNING METHODS FOR IRIS RECOGNITION:

In the research, iris recognition was examined. For many years, iris recognition technology has been researched; which has been studied for decades, and it was carried out by Daugman et al. [2]. Recent developments in image processing have made it possible for researchers to successfully increase the effectiveness of iris identification systems by utilizing a deep learning-based approach. To extract iris image features for recognition [16] used five convolutional neural network (CNN) models that had already been trained. These models included the Visual Geometry Group (VGG), AlexNet, DenseNet, ResNet, and Google Inception models.

They demonstrated how deep features perform better for iris detection than manually generated image features. [17,18] used CNN to compare and match iris images taken under different situations for heterogeneous iris verification/recognition.[19]

developed a CNN-based multi-model iris identification using left and right iris images to increase accuracy. The recognition and iris localization tasks use CNN. 20] suggested using CNN to detect pupil and iris boundaries, which are essential to iris identification. This makes workplace iris recognition systems efficient and reliable. Over the years, many iris segmentation methods have been proposed. Pixel-based and boundary-based segmentation methods exist [21]. Boundary-based methods isolate iris texture areas and focus on pupillary, limbic, and eyelid boundaries. Pixel-based methods quickly distinguish iris pixels from non-iris pixels based on pixel appearance. The Daugman integro differential operator and Wilde circular Hough transform are the most famous boundary-based algorithms. Both methods assumed circular pupil and limbic boundaries, which was most significant [22]. Few researchers have focused on applying Convolutional Neural Networks (CNNs) for iris segmentation. Liu et al. [23] addressed intra-class variation in heterogeneous iris images using their DeepIris model. Gangwar et al. [17] created DeepIrisNet to demonstrate iris recognition across several sensors. In the field of iris segmentation with CNNs, Liu et al. [18] used fully convolutional networks (FCN) to precisely detect iris boundaries in difficult, non-cooperative situations. Alaa S. Ai Wisy[24] and colleagues developed the IrisConvNet architecture, which uses a convolutional neural network and a softmax classifier to automatically extract unique features from iris images. Furthermore, deep learning techniques seek to improve iris identification by replacing classic Gabor filters with ones developed through deep learning. Minnae et al. [25] collected features from the VGG-Net and used Support Vector Machines (SVM) for further analysis. A network known as UniNet, which is built on fully convolutional networks, was presented by Zhao and Kumar. In order to concentrate on the bit-shifting and noniris masking operations in the matching stage, they created a loss function that corresponds to a variation of the Triplet Loss [7].

3. METHODOLOGY

3.1 Proposed methods

A diagram of the proposed system is shown in Fig.1 first iris image acquisition process has been done using Iscan sensor. Then the preprocessing step has done it is essential for enhancing raw data and offering accurate identification, particularly in the area of iris recognition.

Then the GST methods used for iris segmentation to get the coarse iris boundaries. After the iris has been coarsely segmented, specular reflections are eliminated, and then precise iris segmentation is carried out to eliminate the eyelids and produce a mask that is more accurate.

At last, we use the newly segmented iris mask to carry out the pupil detection. Finally feature extraction and matching process has been done from unwraps iris region from the input image based on polar coordinates. In the end, the iris region of the input image was unwrapped using polar coordinates for easier feature extraction and matching [26].

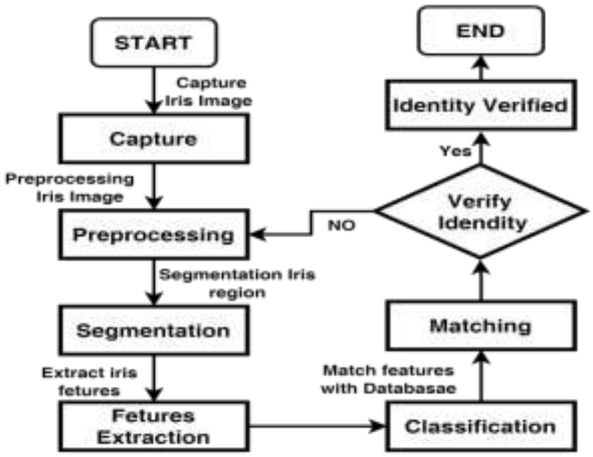

Fig.1 Diagram of proposed Iris recognition work

3.1.1 Pre-processing

3.1.1.1 Specular reflection removal:

Specular reflections had significant impacts on the quality of the image. The goal of this experiment is to remove specular reflections from images by combining adaptive thresholding with morphological closing to the image processing. By converting the image to binary (black and white), adaptive thresholding also makes it easier to identify specular reflections from the background of the image. In order to enhance the image quality and remove a few small bright spots that are directly related to specular reflection, morphological closure is also employed during the process [6].

The following is the formula for adaptive thresholding:

$$threshold(x,y) = \{ 0^{maxValue\ if\ I(x,y) > block\_size - c} \quad otherwise \qquad (1)$$

Where

- $I(x, y)$ represent the input pixel intensity from the coordinate (x, y)
- $blocksize$ Declare the size of neighbourhood area, if $I(x, y)$ is greater than $blocksize - c$, set the maximum value for a pixel . If not assign it the value 0.
- c Represent constant subtracted from mean.

Finally, highlight the specular reflection-free areas for more clearly which is more useful for further processing [19].

3.1.1.2 Eyelash removal:

When eyelashes are present in iris images, feature extraction may be disturbed by and hence reduce the overall recognition accuracy. A p-rank filter and morphological closure are used to target eyelashes, producing an iris image clear of eyelashes. Firstly, the p-rank filter helps find areas with high-frequency content that might be related to eyelashes. It is possible to measure the local median frequency in a certain area. This is important as eyelashes are frequently seen as high-frequency elements inside iris images. When eyelash identification is achieved, an eyelash mask is mostly generated using a user-specified threshold and a p-rank filter that is computed using the median_filter function with a given kernel size (ksize)[26].

Here Each pixel I(x,y) in the image has its p-rank filter P(x,y) calculated for it using the median filter:

P(x, y) = median(neighborhood(I, x, y)) (2) Where the (I, x, y)*indicate* the neighbourhood of pixel I(x,y).

- I(x,y) indicate the pixel intensity of (x,y) coordinate.
- P(x,y) indicate the value of p-rank filter of (x,y) coordinate.

Each pixel in this mask has a binary value (1 or 0) indicating whether it might be an eyelash area. Potential eyelash regions are identified as pixels with frequencies under a specific threshold multiplied by the p-rank filter value as follows:

$$\mathbf{M(\mathcal{X}, \mathcal{Y}) = \{ 1 \text{ if } F(\mathcal{X}, \mathcal{Y}) < P(\mathcal{X}, \mathcal{Y}) \times T \text{ otherwise } 0} \qquad (3)$$

Where,

- M($\mathcal{X}$, $\mathcal{Y}$) is eyelash mask obtain from threshold T.
- F($\mathcal{X}$, $\mathcal{Y}$) is the image frequency of (x,y) coordinate.
- P($\mathcal{X}$, $\mathcal{Y}$) is P-rank filter

Following the segmentation of the eyelash region, a 5×5 kernel morphological closing procedure is conducted. This process makes sure that isolated pixels are eliminated and that eyelash regions are well-connected by utilising cv2.morphologyEx with cv2.MORPH_CLOSE. Lastly, as illustrated in Fig.2, an inverted mask is applied to the original image to produce an image without eyelashes [27,28].

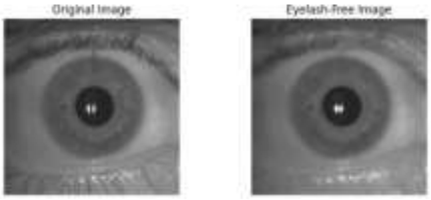

Fig.2 shows eyelash region remove from original images

3.1.2 Segmentation

3.1.2.1 Iris Boundary Detection:

To identify the iris boundary, use the GST-based method. The iris centre and radius used in the demonstration are placeholders; the real GST-based method would take their place. An essential stage in iris recognition systems is the identification of iris boundaries. The Generalised Structure Tensor (GST) method is used to determine the iris boundary. The generalised structural tensor is a mathematical framework used in image processing for segmentation and texture analysis. It detects localised texture differences in images [9]. The equation for the generalized structural tensor is as follows:

The structural tensor elements Sij are defined based on an image I(x,y), where x and y denote the spatial coordinates.

$$\boldsymbol{Sij} = \sum \boldsymbol{x},\boldsymbol{y}\ \ \boldsymbol{w}(\boldsymbol{x},\boldsymbol{y}).\nabla\ \boldsymbol{Ii}(\boldsymbol{x},\boldsymbol{y}).\nabla\ \boldsymbol{Ij}(\boldsymbol{x},\boldsymbol{y}) \qquad (4)$$

In this context

- $\boldsymbol{i}$ and $\boldsymbol{j}$ represent the gradient components, which typically denoted as $\boldsymbol{i},\boldsymbol{j} \in \{\boldsymbol{x},\boldsymbol{y}\}$.
- The weight function $\boldsymbol{w}(\boldsymbol{x},\boldsymbol{y})$ define the specific region of interest. In iris segmentation, A circular or elliptical window centred on the pupil is frequently employed in iris segmentation to highlight the iris.
- The partial derivatives of the function $\boldsymbol{i}$ with respect to the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are denoted as $\nabla\ \boldsymbol{Ii}(\boldsymbol{x},\boldsymbol{y})$ and $\nabla\ \boldsymbol{Ij}(\boldsymbol{x},\boldsymbol{y})$The symbols $\nabla\ \boldsymbol{Ij}(\boldsymbol{x},\boldsymbol{y})$represent the partial derivatives of the image intensity *I* with respect to the spatial coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$.

In practice, the pixel values inside the specified region of interest (e.g.the iris area) are added together to generate the structural tensor. The weight function is used to weight each pixel's contribution, and gradient information is computed in both the x and y directions [5].

- The iris center is specified as (x,y) with coordinates (W/2,H/2) to indicate its central position in the image.
- The iris radius RR is set to the minimum value between W/3 and H/3 to maintain image dimensions.
- After reducing the iris radius by iris_radius_fraction, the final radius is R' = R × iris_radius_fraction.

Here the size of the image is 400 pixels high and 600 pixels wide. At first, the radius of the predicted iris centre is 200 pixels, or one-third of the height, and it is (300, 200). If iris_radius_fraction = 0.8, then the final radius is equal to (0.8 * 200) = 160 pixels. Examine the original image to see the detected iris center and radius, and then compare it to the real iris boundary to determine the accuracy which is shown in fig.3[9].

3.1.2.2 Pupil segmentation:

A pupil localization process involves finding the pupil's center and radius, while an iris perimeter recognition process involves precisely finding the edge between the iris and the pupil. To find pupil boundary function tries to find the edge of the pupil in the picture of the iris. To move on with iris identification, it is important to correctly set the pupil border. To find the pupil radius in the experiment, the center and radius of the eye were taken from images [29]. The starting pupil radius was found by dividing the iris radius by three. The goal of this step was to make the pupil border smaller so that it would be easier to see where the edges are. The iris radius, *iris_radius,* was used to find the pupil radius, *pupil_radius*. The pupil radius is equal to iris_radius//3. This division by three made the pupil border smaller, which made it easier to find. The Fig.3 shows the detection of iris and pupil boundaries [30].

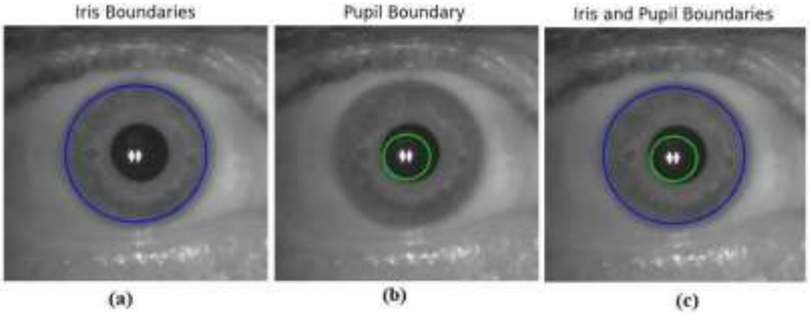

Fig.3. Detection of iris boundaries and pupil boundaries from eye images

3.1.3 Hough Transform for edge detection

The Hough Transform is a frequently utilised technique in computer vision. Finding lines, circles, or other shapes with parameters can be done with it [31]. Since the edges of the pupil and the eye are both rounded, the edge of the iris area can be found using a circular Hough transform. When a circle appears in the image, the circular Hough transform is explained as follows:

$$(x-a)2 +(y-b)-r2 =G(xk, ykr) \qquad (5)$$

where the radius (r) and the center's coordinates (a,b) are located. This is followed by transforming random edge points into the (a,b,r) parametric form of a circular parameter space. All of the picture points will intersect at point (a,b,r) if they are all orientated on the circle. Here, intensity thresholding and derivatives of the ocular picture are used to first derive the edge map of the image [32].Next, the Hough transform for the given contour is optimized using the edge map. As is the notation for the Hough transform.

n
$(xk , ykr)=$□$h(x j , y j ,xk , ykr)$ Where (6)
$j=1$
$h(xj, yj,xk , ykr)=$□□1 $ifG(xk , ykr) = 0$
□0 *otherwise*

Since eyelids are usually oriented horizontally, vertical edge information can be used to identify the limbus and horizontal edge information can be used to identify the eyelids.

3.1.4 Normalization iris regions:

Normalization is the process of converting the iris area from Cartesian to polar coordinates. This unwraps the iris into a rectangular shape, allowing for more efficient future processing and rotational invariance. After proper iris segmentation, a normalized template is created. Iris images taken many times may vary in size. The identical iris image can be taken numerous times; however the size will vary owing to various circumstances. These considerations include the fact that the distance between the iris and the acquisition device will never be exactly the same, the camera or head will rotate, and ambient illumination will change pupil size [33]. In strong light, our pupils shrink, and in dim light, they grow. To account for these and other issues, a rectangle normalized picture is constructed to compare two iris samples. Iris normalization procedures include Daugman's Rubber Sheet Model, Wildes' Image Registration, Boles' Method, and others [2,34]. The iris region is usually translated to polar coordinates as follows:

3.1.4.1 Polar Coordinate Transformation:

To unwrap the iris, the code use polar coordinates. Polar coordinates are made up of an angle (θ) and a radius (r). To convert polar coordinates (x, y) to Cartesian coordinates we use:

$$\boldsymbol{x = center[0] + r.\cos[\theta]} \qquad (7)$$
$$\boldsymbol{y = center[1] + r.sin[\theta]}$$

Here, center[0] and center[1] represent the x and y coordinates of the circular iris region's center[35].

3.1.4.2 Adjusting Angle and Radial Scaling:

Adjusting the angle $\boldsymbol{\theta}$ to span a range of 1.7 times π (pi) in order to map the iris into a rectangular shape:

$$\theta = 1.7\pi.\frac{x}{output\_size[1]} \qquad (8)$$

In this case, x represents the current x-coordinate in the output image, and output_size[1] is the width of the output image. The radial scaling r is set to radially map the iris onto the rectangle space:

$$\mathrm{r} = \frac{1-radius-y}{output\_size[0]} \qquad (9)\ (9)$$

In this context, the variable "radius" represents the radius of the iris. The variable "y" denotes the current y-coordinate in the output image, while "output_size[0]" refers to the height of the output image[36].

3.1.4.3 Unwrapping iris:

For every pixel in the output rectangular image:

- Determine the relative polar coordinates (θ,r) according to the current pixel's position (x,y) in the output image using the formulas for adjusted angle and radial scaling.
- Converting the polar coordinates to the Cartesian coordinates look likes:

$$\mathbf{src\_}\ \boldsymbol{x} = \mathbf{center}\ [\mathbf{0}] + \mathbf{r} \cdot \mathbf{cos}(\boldsymbol{\theta}) \qquad (10)$$
$$\mathbf{src\_y} = \mathbf{center}[\mathbf{1}] + \mathbf{r} \cdot \mathbf{sin}(\boldsymbol{\theta})$$

Verify that the Cartesian coordinates that were calculated ($\mathbf{src\_}\ \boldsymbol{x}$, $\mathbf{src\_y}$) fit inside the boundaries of the original iris image.

If the Cartesian coordinates are correct, obtain the pixel intensity from the original image at the location indicated by ($\mathbf{src\_}\ \boldsymbol{x}$, $\mathbf{src\_y}$), and then assign that value to the current pixel in the image that has been unwrapped [37.]

These processes convert polar iris coordinates to Cartesian ones, unwrap the iris so that it takes on a rectangular shape, and finally crop the image so that the top portion is no longer visible which is shown in the fig.5(b)[38].

3.1.5 Iris feature extraction:

Firstly, a horizontal Sobel filter extracts the furrow pattern from an unwrapped iris image, and then the gradient peaks are identified. The Sobel operator convolutionally computes gradient magnitude at each pixel on normalized picture with sobel kernel. The Sobel kernel detects xdirection horizontal intensity changes.

Mathematical operation of convolving an image I with the Sobel kernel Kx in the x -direction expressed as: $Gx=I*Kx$ Gx denotes the gradient in x direction. The variable Gx represents the dimension of horizontal edges, or furrows, in this context. For identification and visualization, a mask is constructed with white furrow places representing furrow features[39].

Second, precise edge detection and morphological processes extract and enhance crypt patterns. Create a binary edge image using furrow characteristics via canny edge detection. Morphological closing smooth and close edge gaps. Finally, edges are flipped for crypt focus. A bitwise NOT operation on binary edges provides this. Then we take furrow and crypt features and combine them using element-wise addition.

$$\boldsymbol{combined\_features}(\boldsymbol{i},\boldsymbol{j}) = \boldsymbol{furrow\_features}(\boldsymbol{i},\boldsymbol{j}) + \boldsymbol{crypts\_features}(\boldsymbol{i},\boldsymbol{j}) \qquad (11)$$

After that matching procedure utilizes ORB then extracts key points and descriptions from input photos. Matching descriptors with Brute-Force Matcher and Hamming distance [40]. Finally, furrow and crypt matches are shown separately which is shown in fig.6. Next, extract *ciliary_zone* features from the unwrapped iris image. Additionally, morphological processes create a binary image that highlights the ciliary zone feature from iris images shown in fig.7. The Brute-Force Matcher with Hamming distance matches iris ciliary zone features to a cropped unwrapped iris. This matching is important for iris recognition [41].

4. CONVOLUTIONAL NEURAL NETWORK:

A Convolutional Neural Network (CNN) is a feed-forward multilayer neural network that is different from traditional fully connected neural networks because it has some locally connected layers that are used for automatic feature recognition and then some fully connected layers that are used for classification. The CNN architecture as illustrated in Fig.4 consists of multiple separate layers, such as one or more fully connected layers, subsampling layers called pooling layers, and sets of locally connected convolutional layers (each with a certain number of distinct learnable kernels) [42]. Three architectural elements are combined in the CNN's internal structure to enable success in several domains, including speech recognition, NLP, image processing, and pattern recognition. There are two main parts of a CNN: fully connected output classification layers and feature extractors, convolutional layers, and max pooling layers. Put all the feature responses from the whole picture into fully connected layers to get the final result, which is generally pretty small [43].

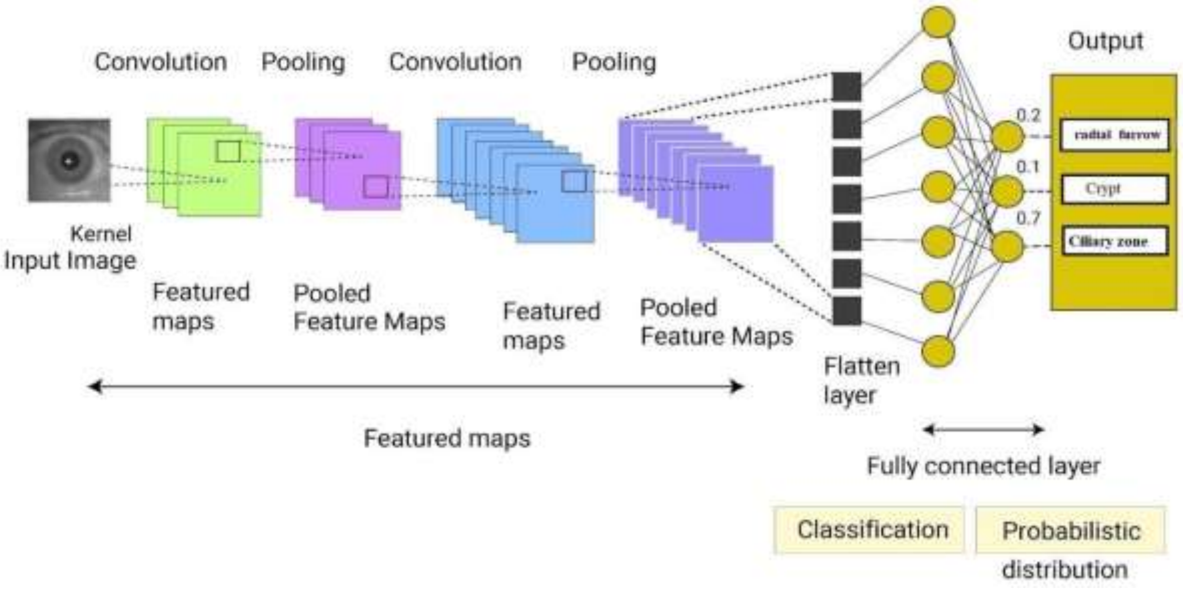

Fig.4…Block Diagram of Convolutional Neural Network

- ***Convolutional Layers:*** Using various kernels from a three-dimensional array of convolutional layers, whose size corresponds to a two-dimensional feature shape, the major feature maps of the complete image are produced.

$M(l-1)$

$$_j^{(l)} = f\left(\sum \quad x_i^{(l-1)} * K_{ij}^{(l)} + b_j^{(l)} \right)$$

x (12)

i=1

There are two sets of outputs, xj(l) and xi(l−1) . K(ijl) is kernel for the current layers, M(l − 1) is used to select the feature maps by layer which is l-1. * denotes two dimensional convolutional operator, b(jl) given as a bias parameter and $f(\cdot)$ provide a nonlinear activation function i.e. to be a rectified linear unit (ReLU*) f(x) = max(0,x)*[44]**.**

- ***Max Pooling layers**:* pooling layers are used to reshape the matrix, with 2*2 filters being a popular practice. Down sampling approaches are used extensively. The procedure of applying a stride to a matrix includes determining the maximum value and using max pooling layers. The most common strategy is to reduce the dimensions of the input matrix array, hence minimizing the number of parameters to be learned during training. This allows us to identify the pixel with the highest intensity in the filter, which is subsequently mapped to the output. Reducing a matrix's size reduces the memory and computational burden of succeeding layers while retaining spatial invariance.
- ***Fully connected layers:*** The Fully Connected Layer is the last layer in a Convolutional Neural Network (CNN). To make additional processing easier, this layer takes the output from the layers above and converts it into a one-dimensional representation. This could

be used as a source of information for the next step. Within the specified layer, each node is linked to the next node by a coefficient. Among the several accessible classes, the top probability pixel that is chosen follows a probability distribution [45].

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Database:

Researchers have worked hard to create many iris image databases, including CASIA [46], UBIRIS [47], MICHEDB [48], and IITD Unfortunately, it was hard to find a database that would work for our study because they all either only have single eye images, or it is hard to find the right one for our study. In this work we have developed own KVK-R certain modalities database acquired using high quality commercial sensors followed by Standard procedure. In a laboratory environment, the ISCAN sensor for iris image data enrolment has been utilized to capture iris images with the least amount of light illumination. Left iris and right iris are acquired in the following order: 10 attempts for each eye at a distance of 10 cm which contain 5000 iris images from 250 distinct subjects. The resolution qualities of the images are 480 x 480 pixels with 96 DPI. Eyeglasses are removing during the acquisition time. Our proposed method has been evaluated using own generated KVK-R database as well as CASIA iris database. The 80/20 patterns are used for validation and training purpose.

5.2 Experimental Setup:

The dataset consists of gray scale images of irises, resized to 128×128 pixels. The images are split into two sets training set: used to train the cnn model and testing set: used to evaluate the model performance the left and right irises of every subject are treated as two different classes. Thus, the KVK-R dataset and the CASIA-Iris-Thousand have randomly chosen 80% of the data for training and the rest of 20% for testing. For CNN classification we implemented our approach using TensorFlow with Python. It is an open-source machine learning framework developed by Google.

5.3 Iris feature extraction:

Iris feature extraction involves multiple processes to effectively identify and analyze various iris patterns, such as radial furrows, crypts, and ciliary zone characteristics. The procedure begins with extracting radial furrow and crypt features from an unwrapped iris image which is shown in fig.6 Initially, a horizontal Sobel filter is applied to the unwrapped iris to enhance the furrow patterns, highlighting the radial furrows. This is followed by the extraction of crypt features, which include edge detection using the Canny edge detector on furrow-enhanced features. Further morphological processes enhance the crypt characteristics by smoothing the edges.

Afterwards, the furrow features, crypt features, and cropped unwrapped iris are used for feature matching. Keypoints and descriptors inside these features are identified by the ORB (Oriented FAST and Rotated BRIEF) detector once it is setup. Then, to make it easier to compare iris features, the BFMatcher (Brute Force Matcher) is used to match the keypoints and descriptors shown in the fig.7[49]. In addition, ciliary zone characteristics are collected from the cropped, unwrapped iris. A Gaussian blur is used to smooth the image and eliminate noise, hence improving keypoint clarity. Keypoint coordinates are then retrieved, providing additional unique properties for iris recognition. These integrated methods ensure an accurate evaluation of different iris patterns, which improves the accuracy and reliability of iris recognition systems. The approach efficiently captures and compares complicated iris features by utilizing advanced techniques such as Sobel filtering, Canny edge detection, morphological operations, and ORB-based feature for matching [50].

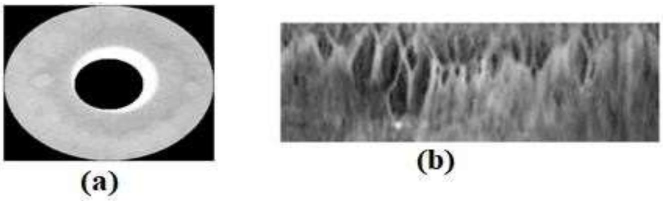

Fig.5 Crop Iris circle from the image(a) and unwraps the iris into a rectangular shape(b).

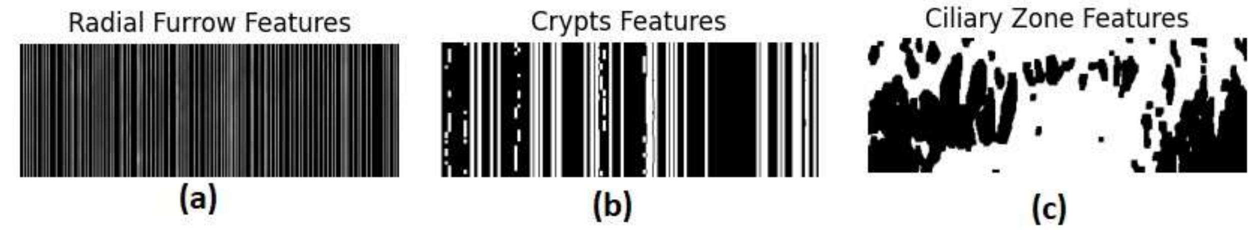

Fig.6 shown effectively identify various iris patterns, such as (a) radial furrows, (b)crypts, and (c)ciliary zone characteristics.

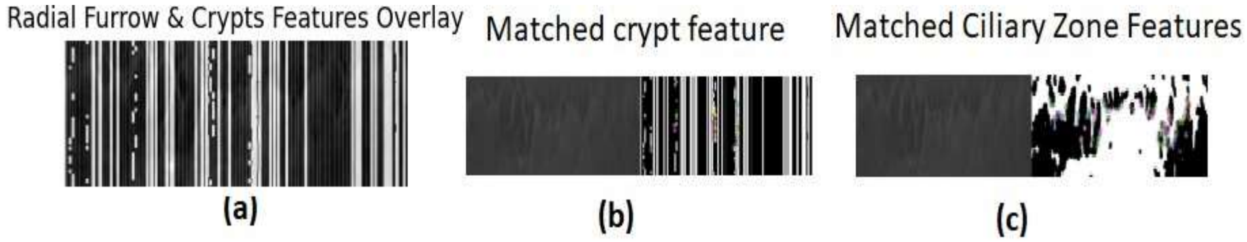
Radial Furrow & Crypts Features Overlay
(a)
Matched crypt feature
(b)
Matched Ciliary Zone Features
(c)

Fig.7 shows the comparison and matching of the furrow features(a), crypt features(b), and ciliary zone(c). The Keypoints and descriptors are indentified inside in features.

5.4 Model Training:

We describe the Convolutional Neural Network (CNN) model's architecture that was designed for iris recognition. The CNN model was trained on own generated KVK-R database for the training and testing purpose. A training set had 4000 images for right and 4000 for left eye images and test set had 2000 thousand images of same left and right eye.

Convolutional Neural Network (CNN) model for iris recognition with an input shape of (128,128,1) evaluates 128x128 grayscale images. The model consists of three convolutional layers, which utilizing ReLU activation. The first layer has 32 filters of size 3x3, and the second layer has 64 filters of the same size. Max pooling layers, which lower the spatial dimensions by a factor of two, come after these layers and produce output forms of (63,63,32) and (30,30,64), respectively. After the feature maps are flattened into the 1D vector depicted in fig., overfitting is avoided by preserving smaller, more general weights in a dense layer with 64 units and L2 regularization (factor 0.001). To further reduce overfitting during training, a dropout layer with a rate of 0.5 is added, which randomly sets half of the input units to zero. The final step in the binary classification process is a single-unit output layer with sigmoid activation [51].

By utilizing convolutional layers for feature extraction, pooling layers for dimensionality reduction, and regularization approaches to improve generalization, this model architecture effectively achieves a balance between complexity and robustness, making it a good fit for iris recognition tasks. The total parameter count is 3,704,345, all of which are trainable.

5.5 Model Evaluation:

The performance of the above model was evaluated using several common measures, including as accuracy, precision, recall, and F1 score, for iris recognition. These measurements offer a complete understanding of how well the model recognizes the iris region. The dataset was split into training, validation, and test sets in order to evaluate the model. In order to prevent overfitting, the model was trained on the training set and then stopped early and its hyper parameters were adjusted based on the model's performance on the validation set. To ensure unbiased assessment, the test set was used for the final evaluation.

A confusion matrix was created and the predicted labels and actual test labels were compared in order to assess how well the CNN model performed on the test dataset. The confusion matrix (CM) is a 2x2 matrix that measures the results of the categorization in the following plot in Fig.8:

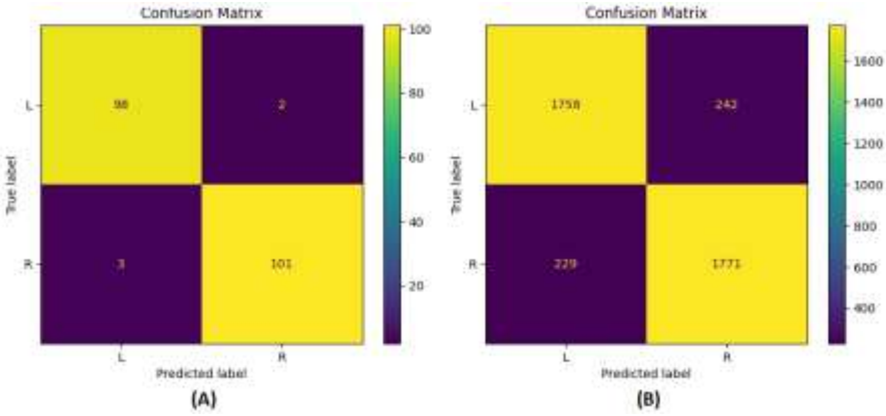

Fig.8 shows Confusion matrix of (A)KVK-R and (B)CASIA-Iris dataset respectively

To enhance clarity, the display labels "L" denoted the left eye and "R" the right eye. The confusion matrix plot, which shows the proportion of accurate and inaccurate predictions for each class, offers a clear visual representation of the model's performance.

6. PERFORMANCE ANALYSIS:

As previously mentioned, the various layers represent varying degrees of visual content. After utilizing each layer's output as a feature vector to represent the iris, we assess the recognition accuracy in order to examine the performance caused by each layer. For the two datasets, KVK-R and CASIA-Iris thousand, the recognition accuracies are shown in fig.9. The KVK-R iris database and the CASIA iris database were the two datasets used to assess the CNN model for iris classification and recognition. With a recognition accuracy of 97%, the self-generated KVK-R database demonstrated exceptional performance and high precision in iris pattern identification which shown in fig.9(a). On the other hand, the model achieved 89% recognition accuracy on the CASIA-iris database the accuracy plot shown in fig.9(b). The variation in accuracy draws attention to a number of possible causes, including as variations in picture circumstances, iris pattern diversity across the two datasets, and quality variation in the datasets. The better performance on the KVK-R database suggests that the model works especially well with the features of this dataset [24]. On the other hand, the slightly lower accuracy on the CASIA dataset suggests that the model needs to be fine-tuned and possibly improved in order to handle a wider range of iris variations more effectively.

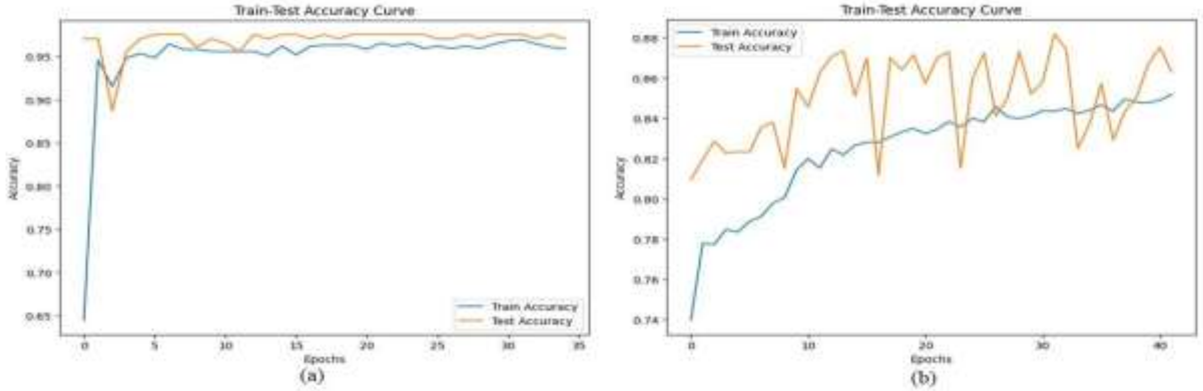

Fig.9 Recognition accuracy of proposed CNN model on two datasets (a)KVK-R dataset and (b) CASIA-Iris Datasets.

7. CONCLUSION:

In this paper we have approached the task of iris recognition using convolution neural network and generalized structure tensor view. The GST method was used to estimate the iris boundary. After preprocessing the input image, the iris center and radius are calculated, with the radius adjusted based on the specified fraction. Our experiment have shown that iris feature patterns, iris segmentation and classification process have used the integration of Sobel filtering, Canny edge detection, morphological operations, and ORB-based feature matching techniques offers a thorough and efficient way for extracting iris features. This multi-step procedure improves the ciliary zone characteristics, crypts, and radial furrows' clarity and distinctiveness, which raises the accuracy and reliability of iris recognition systems. Afterword the CNN model performed well on the KVK-R iris database with 97% accuracy, but decreased to 89% on the CASIA database. High sensitivity and specificity were found in the confusion matrix, however false positives and negatives might be reduced. The study emphasizes dataset quality and diversity in model training. To improve model performance and flexibility, future study should include more iris images and explore other regularization methods.

In order to improve model performance, future research should concentrate on increasing the diversity of datasets and exploring new regularization strategies. Overall, in this work shows how CNNs could help make biometric identification systems get better.

ACKNOWLEDGEMENT

The authors would like to thank the KVK-R database and the Multimodal Biometric Research Laboratory at the Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Chhatrapati Sambhajinagar. This work was funded under the UGC-SAP-II (DRS-II) scheme.

REFERENCES:

1. Nguyen, K., Fookes, C., Ross, A., & Sridharan, S. (2017). Iris recognition with off-the-shelf CNN features: A deep learning perspective. IEEE Access, 6, 18848-18855.
2. Daugman, J. (2004). How Iris works‖ IEEE Transaction on circuit and systems for Video Technology.
3. Minaee, Shervin, and Amirali Abdolrashidi. "Deepiris: Iris recognition using a deep learning approach." *arXiv preprint arXiv:1907.09380* (2019).
4. Al-Waisy, Alaa S., et al. "A multi-biometric iris recognition system based on a deep learning approach." *Pattern Analysis and Applications* 21 (2018): 783-802.

5. Kumaresan, S. Joshua, J. Raja Paul Perinbam, and D. Ebenezer. "Person Identification with Iris Recognition Based on Generalized Structure Tensor." (2015).
6. Li, Yung-Hui, Po-Jen Huang, and Yun Juan. "An efficient and robust iris segmentation algorithm using deep learning." *Mobile Information Systems* 2019 (2019).
7. Daugman, John. "How iris recognition works." *The essential guide to image processing*. Academic Press, 2009. 715-739.
8. Poursaberi, Ahmad, and Babak N. Araabi. "A novel iris recognition system using morphological edge detector and wavelet phase features." *ICGST International Journal on Graphics, Vision and Image Processing* 5.6 (2005): 9-15.
9. Alonso-Fernandez, Fernando, and Josef Bigun. "Iris boundaries segmentation using the generalized structure tensor. A study on the effects of image degradation." *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2012.
10. Gangwar, Abhishek, et al. "IrisSeg: A fast and robust iris segmentation framework for non-ideal iris images." *2016 international conference on biometrics (ICB)*. IEEE, 2016.
11. Bakshi, Kavitha Amit, B. G. Prasad, and K. Sneha. "An efficient iris code storing and searching technique for Iris Recognition using non-homogeneous Kd tree." *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*. IEEE, 2015.
12. Raffei, Anis Farihan Mat, et al. "Fusing the line intensity profile and support vector machine for removing reflections in frontal RGB color eye images." *Information Sciences* 276 (2014): 104-122.
13. Xu, Li, et al. "Structure extraction from texture via relative total variation." *ACM transactions on graphics (TOG)* 31.6 (2012): 1-10.
14. Farouk, Rahmatallah Hossam, Heba Mohsen, and Yasser M. Abd El-Latif. "A proposed biometric technique for improving iris recognition." *International Journal of Computational Intelligence Systems* 15.1 (2022): 79.
15. Yin, Yimin, et al. "Deep learning for iris recognition: a review." *arXiv preprint arXiv:2303.08514* (2023).
16. Nguyen, Kien, et al. "Iris recognition with off-the-shelf CNN features: A deep learning perspective." *IEEE Access* 6 (2017): 18848-18855.
17. Liu, Nianfeng, et al. "DeepIris: Learning pairwise filter bank for heterogeneous iris verification." *Pattern Recognition Letters* 82 (2016): 154-161.
18. Gangwar, Abhishek, and Akanksha Joshi. "DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition." *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016.
19. Al-Waisy, Alaa S., et al. "A multi-biometric iris recognition system based on a deep learning approach." *Pattern Analysis and Applications* 21 (2018): 783-802.
20. Arsalan, Muhammad, et al. "Deep learning-based iris segmentation for iris recognition in visible light environment." *Symmetry* 9.11 (2017): 263.
21. Liu, Nianfeng, et al. "Accurate iris segmentation in non-cooperative environments using fully convolutional networks." *2016 International Conference on Biometrics (ICB)*. IEEE, 2016.
22. Daugman, John G. "High confidence visual recognition of persons by a test of statistical independence." *IEEE transactions on pattern analysis and machine intelligence* 15.11 (1993): 1148-1161.
23. Chai, Tong-Yuen, et al. "Local chan-vese segmentation for non-ideal visible wavelength iris images." *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, 2015.
24. Alaa, S., et al. "A multi-biometric iris recognition system based on a deep learning approach." *Pattern. Anal. Appl* 21 (2018): 783-802
25. Minaee, Shervin, Amirali Abdolrashidiy, and Yao Wang. "An experimental study of deep convolutional features for iris recognition." *2016 IEEE signal processing in medicine and biology symposium (SPMB)*. IEEE, 2016.
26. Fuentes-Hurtado, Félix, et al. "A hybrid method for accurate iris segmentation on at-a-distance visiblewavelength images." *EURASIP Journal on Image and Video Processing* 2019 (2019): 1-14.
27. https://www.geeksforgeeks.org/python-morphological-operations-in-image-processing-closing-set-2/
28. Zhu, Wenyao, Zemao Zhao, and Yufeng Wu. "An algorithm of eyelashes detection for iris recognition." *International Journal of Security and Its Applications* 10.7 (2016): 195-202.
29. Alonso-Fernandez, Fernando, and Josef Bigun. "Iris segmentation using the generalized structure tensor." *SSBA Symposium 2012 (SSBA2012), Stockholm, Sweden, 8-9 mars, 2012*. 2012.
30. Simpson, Michael J. "Scaling the retinal image of the wide-angle eye using the nodal point." *Photonics*. Vol. 8. No. 7. Multidisciplinary Digital Publishing Institute, 2021.
31. Houston, Caroline. "Iris Segmentation and Recognition Using Circular Hough Transform and Wavelet Features." *Rochester Institute of Technology* (2010).
32. Cherabit, Noureddine, Fatma Zohra Chelali, and Amar Djeradi. "Circular hough transform for iris localization." *Science and Technology* 2.5 (2012): 114-121.
33. Bonyani, Mahdi, Maryam Ghanbari, and Ahmad Rad. "Different gaze direction (DGNet) collaborative learning for iris segmentation." *Available at SSRN 4237124* (2022).
34. Tomeo-Reyes, Inmaculada, et al. "A biomechanical approach to iris normalization." *2015 International Conference on Biometrics (ICB)*. IEEE, 2015.
35. How to change an image from Cartesian to Polar coordinates in Matlab? - Stack Overflow
36. Polar coordinates mapping - Math Insight
37. Polar and Cartesian Coordinates (mathsisfun.com)

38. Färberböck, Peter, et al. "Transforming rectangular and polar iris images to enable cancelable biometrics." *International Conference Image Analysis and Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
39. Bogdan, Victor, Cosmin Bonchiş, and Ciprian Orhei. "Custom extended sobel filters." *arXiv preprint arXiv:1910.00138* (2019).
40. Canny, John. "A computational approach to edge detection." *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986): 679-698.
41. Liu, Qing, and Cheng-yu Lai. "Edge detection based on mathematical morphology theory." *2011 International Conference on Image Analysis and Signal Processing*. IEEE, 2011.
42. Karpathy, Andrej. "Convolutional neural networks for visual recognition." *Notes accompany the Stanford CS class CS231* (2017).
43. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." *Communications of the ACM* 60.6 (2017): 84-90.
44. Saleem, Muhammad Asif, et al. "Comparative Analysis of Recent Architecture of Convolutional Neural Network." *Mathematical Problems in Engineering* (2022).
45. Yamashita, Rikiya, et al. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging* 9 (2018): 611-629.
46. CASIA iris databases. http://biometrics.idealtest.org/. Accessed 06 Sept2017.
47. UBIRIS iris database. http://iris.di.ubi.pt. Accessed 06 Sept 2017.
48. MICHE iris database. http://biplab.unisa.it/MICHE/. Accessed 06 Sept 2017.
49. Liu, Chang, and Shuwen Dang. "The Research and Application of Improved ORB Feature Matching Algorithm." *International Conference on Guidance, Navigation and Control*. Singapore: Springer Nature Singapore, 2022.
50. Zhou, Dong, et al. "Orb-based template matching through convolutional features map." *2019 Chinese Automation Congress (CAC)*. IEEE, 2019.
51. Winston, J. Jenkin, et al. "Hybrid deep convolutional neural models for iris image recognition." *Multimedia Tools and Applications* (2022): 1-23.