LEX LOCALIS-JOURNAL OF LOCAL SELF-GOVERNMENT
ISSN:1581-5374 E-ISSN:1855-363X
VOL. 23, NO. 11(2025)

ENHANCED FACE RECOGNITION FOR SECURE DRIVER'S LICENSE DATA RETRIEVAL USING MULTI-LEVEL RED DEER CASCADE CONVOLUTIONAL NEURAL NETWORK

Bhoomireddy Venkata Haripratap Reddy[1], Dr. S. P. Vijayaragavan[2*]
Prof. Dr. B. Karthik[3]

[1]Ph.D.Scholar, Department of ECE, Bharath Institute of Higher Education and Research, Chennai, India
[2]Professor, Department of EEE, Bharath Institute of Higher Education and Research, Chennai, India
[3]Professor, Department of EEE, Bharath Institute of Higher Education and Research, Chennai, India

[*]Corresponding Author E-mail: vijayaragavan.eee@bharathuniv.ac.in[2*]

Abstract
Extracting driver's license information using facial recognition presents significant challenges, including handling diverse lighting conditions, face orientations, and occlusions while ensuring data security. Traditional identity verification methods that rely on physical documents or manual checks are prone to inaccuracies and security vulnerabilities. This study addresses the challenge by proposing a robust, automated system for face detection and recognition, aimed at secure retrieval of driver's license data. To tackle the issues of varying illumination and facial appearances, they apply a pre-processing step that utilizes the Integral Normalized Gradient Image (INGI) method, transforming images into an illumination-insensitive format. Our system leverages a Multi-level Cascade Convolutional Neural Network (MLC-CNN) for face recognition. The CNN parameters are optimized using an enhanced Red Deer Optimization (RDO) algorithm to ensure high accuracy. The driver's license details are then retrieved from the database upon successful facial recognition. Our approach demonstrates superior performance compared to existing methods, achieving enhanced accuracy and reliability under challenging conditions. This work provides a secure, efficient solution for facial recognition-based data retrieval, contributing to safer identity verification systems for driver's licenses. The proposed method achieved outstanding performance with an accuracy of 98.9%, precision of 99.4%, recall of 98.4%, and an F-measure of 98.9%, surpassing other approaches in terms of overall effectiveness.

Keywords: driver's license, Integral Normalized Gradient Image, Multi-level Cascade Convolutional Neural Network, and enhanced Red Deer Optimization.

1. Introduction

One of the most important and quickly evolving artificial intelligence (AI) technologies now accessible for security and law enforcement applications is biometric driving license face recognition [1].Facial expression recognition (FER) can benefit from study findings on the intriguing issue of human emotion identification from photographs. Facial recognition is a fascinating field that presents one of the most difficult problems in image retrieval and computer vision. It may be used in many different contexts, including ATMs, healthcare systems, systems for obtaining a driver's license, reservations for trains, surveillance operations, and passport authentication[2]. Facial recognition is employed for secure data retrieval due to its high accuracy, security, and speed in verifying identities. It enhances safety in applications like driver's license systems by ensuring reliable authentication. However, challenges include variations in lighting, facial expressions, and occlusions, which can impact recognition performance.

Any facial recognition system's ability to function depends on the feature extraction procedure. It significantly affects the system's overall operational efficiency. SVM, STISM, STIP, and SIFT are a few of the several feature extractor models that have been created. In addition, adaptive

cruise control (ACC), which the writers first suggested in [3], Anti-lock Braking System (ABS), alcohol ignition interlock devices [4], Traffic Sign Recognition (TSR) [10], driver drowsiness detection [6], Electronic Stability Control (ESC) [7], Forward Collision Warnings (FCW) [8], automotive night vision, collision avoidance systems [5], Lane Departure Warning System (LDWS) [9], and. While feature extraction models like SIFT, SVM, STIP, and STISM improve face recognition accuracy, they can be computationally intensive and sensitive to variations in lighting, pose, and occlusions, affecting real-time performance.

DCNNs are able to identify faces in a wide range of poor quality images. A common training set for these algorithms is millions of facial images from thousands of individuals [11]. A limitation of DCNNs in face recognition is their vulnerability to adversarial attacks and bias, leading to performance degradation on underrepresented groups or distorted images. The creation of network designs also benefits deeply learned features; exemplary CNN models include VGG and AlexNet[12]as well as GoogleNet. Deeper architectures like DenseNet and ResNet were created to improve learning capacity in response to the massive growth in the volume and complexity of training data [13]. A limitation of deeper structures like ResNet and DenseNet is their high computational cost, making them resource-intensive and difficult to deploy on edge devices. To address these issues, this study proposes aMLC-CNN, with CNN parameters optimally selected using an enhanced RDO algorithm. The key contribution of proposed method is listed below;

- Creation of a comprehensive driver's license dataset that includes a wide range of driver photos and personal information, essential for training an accurate and reliable face recognition model.
- Development and application of an advanced pre-processing technique using the INGI method to address the challenge of varying illumination conditions, ensuring consistent and reliable face recognition performance.
- Introduction of a novel MLC-CNN architecture designed specifically for efficient face detection and recognition, handling variations in face orientations and occlusions.
- Implementation of an enhanced RDO algorithm to optimally tune the CNN parameters, significantly improving recognition accuracy and overall system performance.
- Development of an automatic system that securely retrieves driver's license information from a database upon successful face recognition, ensuring data integrity and reducing the risk of security breaches.
- Detailed evaluation of the proposed system's performance across multiple metrics, demonstrating superiority over state-of-the-art face recognition techniques under challenging real-world conditions.

The remainder of this paper is organized as follows: Section 2 reviews the findings from the literature study. Section 3 provides a detailed description of the proposed method. Section 4 presents the experimental results, demonstrating the efficiency of the proposed method and validating its superiority. Finally, Section 5 concludes the paper with a summary of the findings and suggestions for future research.

2. Literature Review

Many researchers have developed Face Recognition for Secure Driver’s License. Among them, some of the works are analysed here;

Jeong et al. [14] introduced a FER technique for real-time embedded systems, particularly those of intelligent cars, based on geometric characteristics and the hierarchical WRF. They created a

quick FER algorithm that can function in low-spec devices mounted in cars and is useful for tracking a driver's emotions. In order to increase the accuracy of the classifier, a hierarchical weighted random forest (WRF) classifier is used, which is trained depending on how similar the sample data is. Three databases—the Keimyung University Facial Expression of Drivers (KMU-FED) database, MMI, and the expanded Cohn-Kanade database (CK+)—were used to test this approach experimentally. Compared to DNN-based techniques, this method uses less memory and computational power. The limitation of the method was that it struggles with false recognition when faces are rotated or partially occluded, which affects its robustness in real-world driving scenarios.

Abuzneid et al. [15] introduced a powerful computational system for human face recognition. By employing a back-propagation neural network (BPNN) and features extraction based on the correlation between the training pictures, they suggested an improved method for better human face recognition. The creation of the T-Dataset, a new set derived from the original training data set, which is utilized to train the BPNN, is a significant addition of this work. This was accomplished by combining the various approaches as each one enhances the system overall and has an edge over the others. It relies on traditional feature extraction methods, which may not fully capture complex facial features compared to modern deep learning techniques like CNN.

Dhabe et al. [16] introduced Ensuring public safety and optimizing traffic management were of utmost importance. Furthermore, they developed the Driving License Verification System (DLVS), which revolutionizes driver license verification by utilizing cutting-edge facial recognition and optical flow techniques. The DLVS addresses the critical need for secure and accurate driver identification. The DLVS addresses the critical need for secure and accurate driver identification, offering a user-friendly interface designed for seamless and efficient verification in a single step.

A biometric-based System for driving licensing verification was presented by Barawkar et al. [17]. They created The Fingerprint identification scheme is a non-imitable biometric identification system that is used to identify non-licensees and avoid them from driving illegally. Among the various biometric features, Fingerprint is one of the best. Maintaining the record of attendance report of students has become a very difficult work because their absences may be documented or missed. The fingerprint-based licensing authentication scheme ensures secure, non-imitable identification, preventing illegal driving and simplifying license verification while maintaining future scalability.

Megha et al. [18] introduced for face recognition by taking out the face's characteristics and comparing it to the test images in order to find a match. Additionally, they created y GLCM filters and EigenFace Algorithm in PCA. A graphical user interface (GUI) has been created to facilitate user interaction with applications. The limitation of our system lies in its inability to detect multiple faces or accurately recognize faces with poses above 90 degrees, requiring more robust algorithms for improved accuracy.

Hankare et al. [19] introduced a wireless fingerprint-based ZigBee-based system for confirming driver's licenses. The license is currently verified manually under the present system, which makes it challenging to monitor. A single rectangular CMOS die has both the detecting and data conversion circuitry for the Finger Chip IC, which is used to take fingerprint images. A conventional serial port device called Max RS 232 is being utilized to solve the issue with microcontroller interface at the transmitter and receiver ports. It is capable of automatically completing tasks such wireless transmission, fingerprint matching, processing, and information

gathering from fingerprints, as well as verifying driver's licenses. The limitation of the developed system is its reliance on ZigBee, which may have limited range and data transmission speed, affecting real-time performance in larger or complex environments.

Choi et al. [20] introduced Following identification, driver-centered services were provided, and a biometrics system employing bio-signals was used. To address the deformation of the ECG morphological characteristics after normalized and the motion artifact noise of the ECG that deteriorates identification performance in the driving environment, they proposed an adaptive threshold filter-based driver identification system. The identification performance was enhanced and examined when the normalizing procedure matched the morphological aspects of the ECG.The limitation of the method is its sensitivity to varying driving environments, which may still affect ECG signal quality and identification accuracy despite the adaptive normalization method.

3. Proposed Face Recognition for Secure Driver's License

This study proposes a deep learning-based methodology for robust face detection and recognition, enabling automatic retrieval of driver's license data while handling varying lighting, orientations, and occlusions with high accuracy and security. The proposed method begins with the collection of a driver's license dataset, including driver photos and personal information. Face images are then extracted from the licenses and used to train a face recognition model for accurate identification. A pre-processing step is applied to reduce noise and account for varying illumination conditions, which can significantly affect facial appearance. To address this, the INGI method is employed to transform images into illumination-insensitive formats. The pre-processed images are then fed into a recognition process using a Multi-level Red Deer Cascade CNN, with CNN parameters optimized via the enhanced Red Deer Optimization (RDO) algorithm. Upon successful recognition, the system retrieves the driver's license details from the database. Figure 1 shows flow diagram of face recognition for secure driver's license data retrieval.

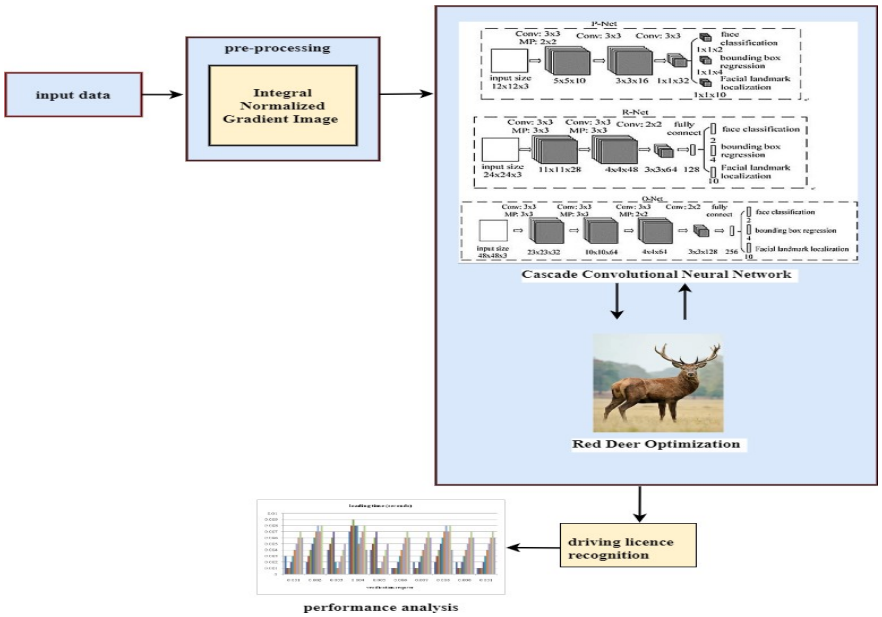

Figure 1: Flow diagram of Face Recognition for Secure Driver's License Data Retrieval

3.1. Dataset Collection

The dataset referenced on Kaggle contains 3000 images of synthetic EU driver's licenses, randomly modified with effects of blur, low light, grain, perspective, and background, along with their truth tables.

Blur: Some images are blurred to mimic motion or camera focus issues, representing situations where driver's licenses may not be clearly captured.
Low Light: The dataset contains images taken under dim lighting conditions, simulating low visibility scenarios common in security footage or poorly lit environments.
Grain: Some images have grainy textures to represent noise that might be introduced during low-resolution captures or poor-quality images.
Perspective: The dataset also includes images with perspective transformations, representing cases where the license is not aligned or is photographed at an angle.
Background Variations: The images are set against different backgrounds, helping simulate real-world environments where licenses might be photographed with clutter or non-uniform settings.

3.2. Preprocessing using INGI

Extract face images from the collected driver's licenses to train a face recognition model for accurate identification. Afterward, the extracted images undergo preprocessing using the INGI technique. Figure 2 shows INGI Preprocessing.

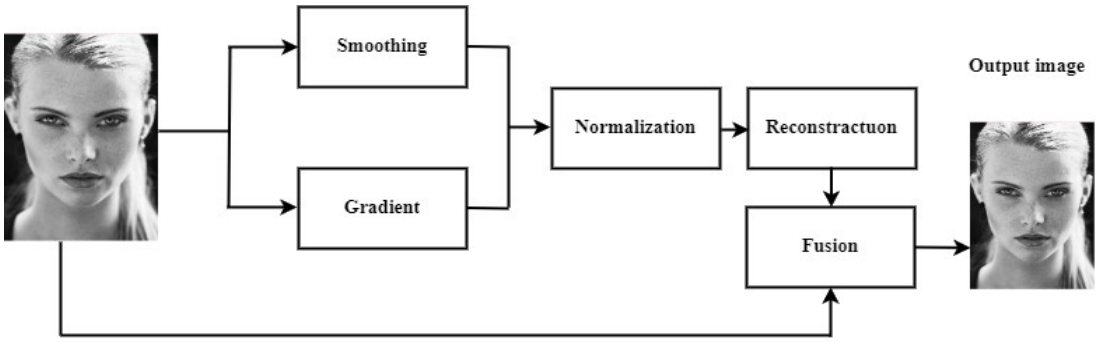

Figure 2: INGI Preprocessing

INGI method is suggested to address sudden variations in brightness, ensuring more reliable detection with minimal negative effects including halo aberrations and picture noise. This method effectively manages both intrinsic and extrinsic factors in the image. By first normalizing the gradients of a smoothed image and utilizing the anisotropic diffusion method to integrate the outcomes, the approach minimizes the impact of lighting variations. The technique operates under two key assumptions: intrinsic factors, such as facial features, are mainly found the realm of high spatial frequency, while extrinsic factors, such as lighting, reside in low spatial occurrence domain. Instead of using traditional high-pass filters, which are not robust against illumination changes and can remove essential intrinsic details, the INGI method utilizes a gradient operation to preserve critical facial features while handling illumination variations. This makes it a highly effective solution for improving the accuracy of face recognition in secure driver’s license identification systems.

$$\tilde{N}_c = \tilde{N}(r \sum_i n^H . g_i) = (\tilde{N}_r) \sum_i n^H . g_i + r\tilde{N}(\sum_i n^H . g_i) \quad (1)$$

$$= (\tilde{N}_r) \sum_i n^H . g_i \quad (2)$$

$$= (\tilde{N}_r) C \quad (3)$$

Here $\tilde{N}_c$ represents the gradient of an image feature or intensity (c) that is affected by both intrinsic (facial features) and extrinsic (illumination) factors. r represents a scalar or a weight applied to normalize the image. $\sum_i n^H . g_i$ Indicates a summation of intrinsic factors, where n^H represents the transpose of the intrinsic factor, and g_i could be a feature vector representing face details. In the Eqn. (1), the gradient of the product of r, $\sum_i$, and the dot product of $n^H g_i$ is taken. In the Eqn. (2), this gradient is expanded into two terms: one involving the gradient of the summation term and another scalar term involving r. The Eqn. (3), simplifies by focusing on the intrinsic feature sum term. Finally Eqn. (4), the equation is reduced to $(\tilde{N}_r) C$, where C represents a weighted vector of intrinsic features derived from the face image.

To overcome the illumination sensitivity, they normalized the gradient map with the following equation:

$$N = \frac{f_g}{C} = \frac{(f_r)C}{C} = \tilde{N}_r$$

Here **N** represents the normalized gradient of the image. $\tilde{N}_c$ is the gradient of the image feature (such as intensity or pixel values) that is influenced by both basic facial features and extrinsic factors (like lighting). C is a weight function that represents the essential features (details of the face) extracted through the gradient operation.

3.3. Recognition Process using Multi-level Cascade Convolutional Neural Network (MLC-CNN)

The pre-processed images are input into a recognition process using a MLC-CNN. Then face images are removed and arranged one by one. Face pictures that are correctly recognized in a series of photographs are chosen, and these filtered series of images are utilized as the inputs for MLC-CNN, which operates in the background to identify and correct the face in the image.

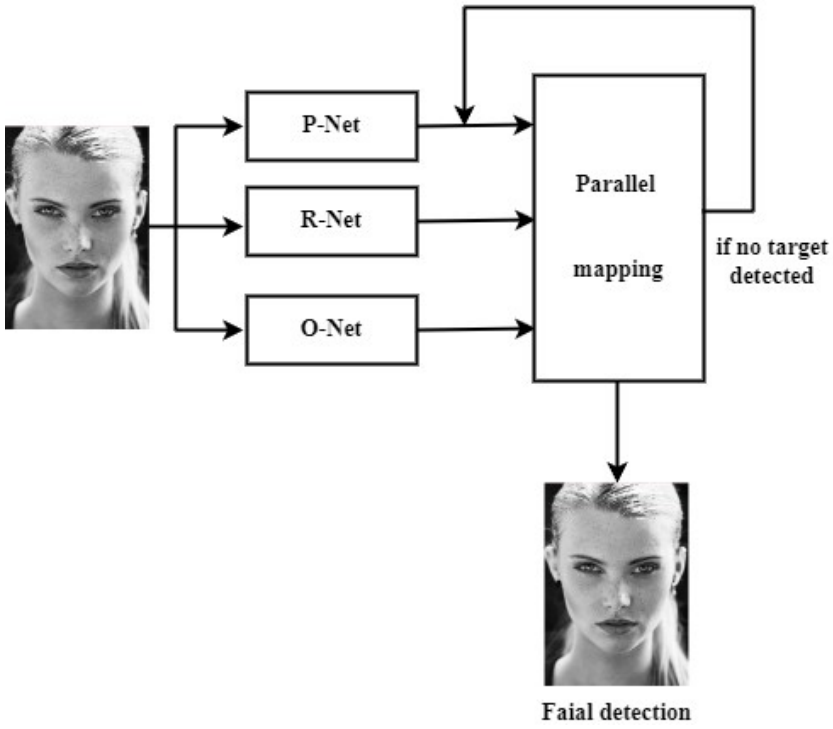

Figure 3: Flow diagram of MLC-CNN

MLC-CNN initially obtains candidate forms and boundaries regression vectors using P-Net, a complete CNN. The candidate form is calibrated in accordance with the bounding box concurrently. Next, eliminate overlapping windows using the non-maximum suppression (NMS) technique. After that, the R-Net network trains on the image that contains the candidate form identified by P-Net, and the network ultimately switches to the complete connection approach for training. After fine-tuning the candidate form using the bounding box vector, eliminate the overlapping form using the NMS. Lastly, compared to R-Net, the network topology is more convolutional. It serves the same purpose as R-Net. It eliminates the overlapping candidate window and displays only the face's key location. Figure 4 show ProposedMLC-CNN based face detection method is divided into 3 steps (a) proposal net (b) refine net (c) output net.

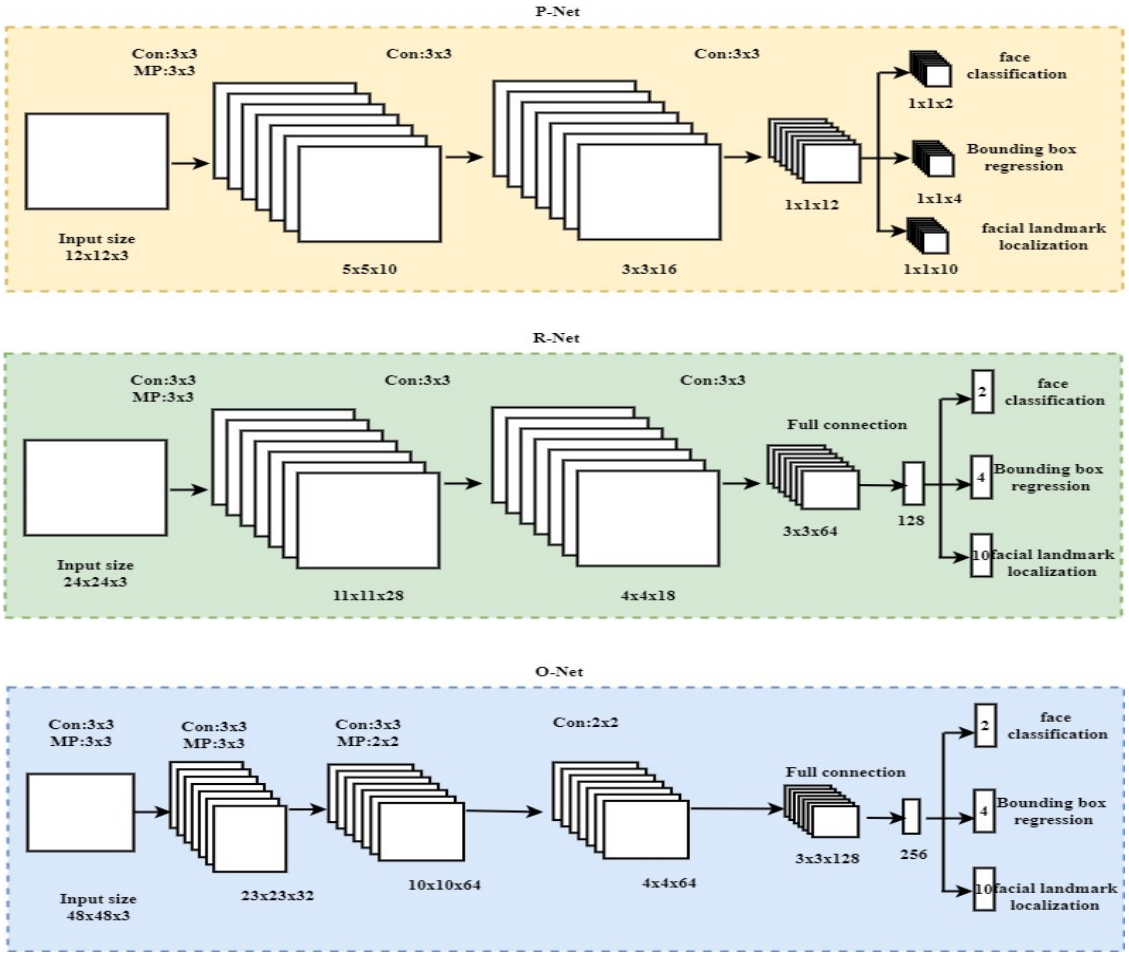

Figure 4: Proposed MLC-CNN based face detection method is divided into 3 steps (a) Proposal net (b) refine net (c) output net

Face Detection Network

Bounding box regression, landmark location, and face/non-face classifier make up the three key components of the MLC-CNN feature descriptor.

$$P_i^{\det} = -\left(a_i^{\det}\log(li)\right) + (1 - a_i^{\det})(1 - \log(li))), \ where a_i^{\det} \hat{I} \{0,1\}$$

The above formula is a cross-entropy loss function for face classification, where li is the probability of the face and $a_i^{\det}$ denotes the ground-truth label.

$$P_i^{box} = \left\| \hat{a}_i^{box} - a_i^{box} \right\|_2^2, \quad 4$$

The regression loss determined by the Euclidean distance is represented by the equation above. Of them, $\hat{a}$ is expected by the network, and a is the quadrant (long, broad, upper left d, upper left a) and the actual true backdrop coordinates. Where the regression target is $\hat{a}_i^{box}$ is obtained by the network and a_i^{box} is the ground-truth coordinate.

$$P_i^{landmark} = \left\| \hat{a}_i^{landmark} - a_i^{landmark} \right\|_2^2,$$

Regression analysis is used to define the Euclidean loss and face landmark detection problems. The left eye, right eye, nose, left mouth corner, and right mouth corner are the five facial landmarks.

$$\min \sum_{i=1}^{N} \sum_{j \in (\det, box, landmark)} a_j b_i^j P_i^j$$

Then, using the equation above, the total learning goal may be created. N denotes the quantity of training samples. a_j denotes on the task importance and $b_i^j \in \{0,1\}$ is the j-th sample's label.

Face Detection Process

Receptive Field Enhanced (RFE-C-CNN) training and testing are carried out in the P-Net, R-Net, and O-Net networks. First, randomly attach pictures from the dataset and resize them to 12 by 12 in order to train the P-Net. Next, depending on the box's to ground truth's Intersection over Union (IOU) ratio, ascertain whether the cropped picture represents a positive or negative sample Second, using a trained P-Net model to identify pictures in the dataset while training the R network. Each image will provide a huge number of candidate windows. Each candidate window's IOU with ground truth determines whether or not it represents a positive and negative sample. After that, train R-Net by resizing these windows to 14 by 14. Lastly, in a manner akin to R-Net, candidate windows are generated using the trained R-Net model; the candidate windows are classified as positive or negative samples based on their IOU with ground truth. After extracting face images from the collected licenses, the overlapped applicant windows are merged using NMS to refine the face regions. Finally, similar to O-Net, the R-Netmodel is applied to these refined face regions, detecting the final bounding boxes and outputting confidence scores, ensuring accurate face localization for further recognition tasks.

The output of classifying features from the extracted face images using MLC-CNN is the predicted class of the input data (i.e., face identity). In this research, the enhanced RDO algorithm is employed for hyperparameter optimization, as it efficiently explores the hyperparameter space to find optimal settings. This ensures that the face recognition model performs at its best, accurately identifying faces from the collected license images. Upon successful recognition, a database query is executed to retrieve the driving license details associated with the recognized driver.

3.3.1. Hyper-Parameter Optimization Using RDO

Hyper-parameter optimization is procedure of determining best mix of C-CNN hyper-parameter settings to optimize performance on data in a reasonable quantity. This process is essential to C-CNN's capacity for precise result prediction. Most of this input uses the hyperparameters' default values. The proposed model optimizes the hyperparameter using the RDO. Table 1 shows the hyperparameter initialization range.

Table 1: Range of hyperparameter initialization

Hyper-parameters	Range
Learning Rate (LR)	0.0001 – 0.01
Batch Size (BS)	16 – 128
Number of Convolutional Layers (CL)	2 – 5
Number of Filters per Layer (FL)	32 – 256
Epochs (E)	50 – 200

The RDO is used to optimize these hyperparameters. The Red Deer algorithm's step-by-step process is explained below.

Step 1: Initialization: Selecting the ideal hyperparameter is the primary goal of this approach. Initially, define the top and lower bounds of the problem, the dimensionality D of the variables, the maximum number of iterations, and the Red Deer size N. Every solution represented by the Red Deer consists of hyperparameters, including LR, BS, CL, FL, and E. Initially, a selection is selected at random. The first solution format is seen in the following equation:

$$P_N = \{S_1, S_2, \ldots\ldots, S_N\} \tag{4}$$

Here, X_N is the Nth Red Deer's position

$$S_i = \{\mathrm{LR}, BS, CL, FL, E\}_i \tag{5}$$

Step 2: Calculating fitness: Following initialization, each solution's fitness is assessed using the RDOtechnique that has been recommended. In this case, the classification accuracy is defined using fitness purpose. The solution with the highest fitness value is deemed to be the most effective. The fitness function is calculated with the assistance using,

$$Fitness = Max\left(\frac{TP + TN}{TP + TN + FP + FN}\right) \tag{6}$$

Step 3: Updating RDO: RDO utilizes 4 distinct techniques known as four different methods of propagation: Selection of Male Commanders, Fighting between Commanders and Stags, Harem Formation, and Stag Mating.

Strategy 1: Selection of Male Commanders

Commanders and Stags are the two categories into which RD is separated.

$$S_{Com} = round\{\gamma . S_{male}\}$$

where S_{Com} consists of the number of men who lead harems. The γ is the algorithm model's starting value. It can have values in the range between 0 and 1. The stag population is calculated by,

$$S_{stag} = S_{male} - S_{Com}$$

where S_{stag} is the proportion of stags to men in the population.

Strategy 2: Fighting between Commanders and Stags

At this point, the male commanders engage in intermittent combat based on their prejudices, i.e., engaging in combat only when the current goal function exceeds the previous one.

$$New1 = \frac{(Com + Stag)}{2} + z_1 \times \left((ub - lb) * z_1\right) + lb$$

$$New2 = \frac{(Com + Stag)}{2} + z_1 \times \left((ub - lb) * z_2\right) + lb$$

where, z_1, z_2 are distributed uniformly between 0 and 1. The search space's upper and lower bounds. The symbols for commanders and stags are Com and Stag, correspondingly.

Strategy 3: Harem Formation

The strength of the male leaders in harems determines how many hinds are there. The male commander's standardized value may be acquired by,

$$S.harem_n^{mate} = round\{a.S.harem_n\}$$

where $S.harem_n^{mate}$ is the quantity of nth harem hinds that mate with their leader. With respect to the solution space, they select at random $S.harem_n^{mate}$ of the $S.harem_n$. Note that a is an initial parameter value for the model of the RDO.

$$offs = \frac{(Com + Hind)}{2} + (ub - lb) \times w$$

where Com and $Hind$ represent the hinds and commanders, respectively. $offs$ is a fresh approach. Note that ub and lb are the respective upper and lower boundaries. Also w is produced at random by a function with a uniform distribution between zero and one.

Strategy 4: Stag Mating

The distance between the male RD and all hinds are calculated by,

$$x_i = \left(\sum_{i=1} \left(stag_i - hind_i^j\right)^2\right)^{1/2}$$

where x_i is the separation between a stag and the i-thhind. Consequently, the selected hind is represented by the lowest value in this matrix.

Step 4: Termination condition: Until the ideal hyper-parameter selection is achieved, the procedure is repeated. The MLC-CNN receives the chosen hyperparameter value. Table 2 provides the RDO pseudocode.

Table 2: Pseudocode for Red Deer Optimization Algorithm

Initialize the Red Deer population.
Calculate the fitness values of each individual in the population.
Sort individuals by fitness and divide them into:
Hinds (S_{hind})
Male Red Deer (S_{male})
Set S^* as the best solution found so far.

For each male RD:
Perform the roaring action
Update the position if it improves the current best position.
Sort the males and divide them into:
Stags
Commanders
For each male commander:
Fight between commanders and stags
Update the positions of both male commanders and stags.
Form harems of hinds for each commander
For each male commander:
Mate with selected hinds from his harem randomly
Randomly select another harem and mate with some of the selected hinds
For each stag:
Calculate the distance between the stag and all the hinds.
Select the nearest hind
Mate with the selected hind
Update S^* if a better solution is found.
End While
Return the best solution S^*.

4. Result and Discussion

Our proposed system achieves notable improvements in facial recognition accuracy and reliability, even under challenging conditions such as varying illumination and occlusions. The implementation of the INGI method and the optimized MLC-CNN results in superior performance compared to existing techniques, ensuring secure and efficient retrieval of driver's license data. The test is run on a machine equipped with an Intel (R) core (TM) i5 4570s CPU @ 2.90 GHz, 8GB RAM, and the computer name SSM107.smg.local running Windows 64-bit. Acer is the system manufacturer using the PYTHON tool. Our experimental configuration includes two data centers with four hosts and a total RAM of 8 GB. The host has a bandwidth of 2800 Mbps.

4.1. Dataset Description

This dataset contains 3000 images of synthetic EU driver's licenses, randomly modified with effects of blur, low light, grain, perspective, and background, along with their truth tables. Specifically:

- 40% of images contain blur
- 24% of images contain low-light and grain effects
- 16% of the images contain low-light only
- 65% of the images contain some perspective effect

4.2. Evaluation Metrics

It has chosen many metrics to gauge how well predict the face recognition for secure driver’s license. They have selected accuracy, precision, recall, and f-measure for our investigation. The confusion matrix was primarily used to determine the true positive, true negative, false positive,

and false negative for the majority of the measurements. To evaluate these findings, compute the precision, recall, accuracy, and F1-score, FPR, FNR, MCC and NPV indicators.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$recall = \frac{TP}{TP + FN} \tag{16}$$

$$precision = \frac{TP}{TP + FP} \tag{17}$$

$$F1\text{-}score = \frac{2TP}{2TP + FP + FN} \tag{18}$$

TP signifies the true positive, FP the false positive, TN the true negative, and FN the false negative.

4.3 Experimental results

The experimental results, including accuracy, precision, recall, F-measure, and accuracy vs. loss value, show the performance of the suggested method compared to FCN, GAN, and LSTM.

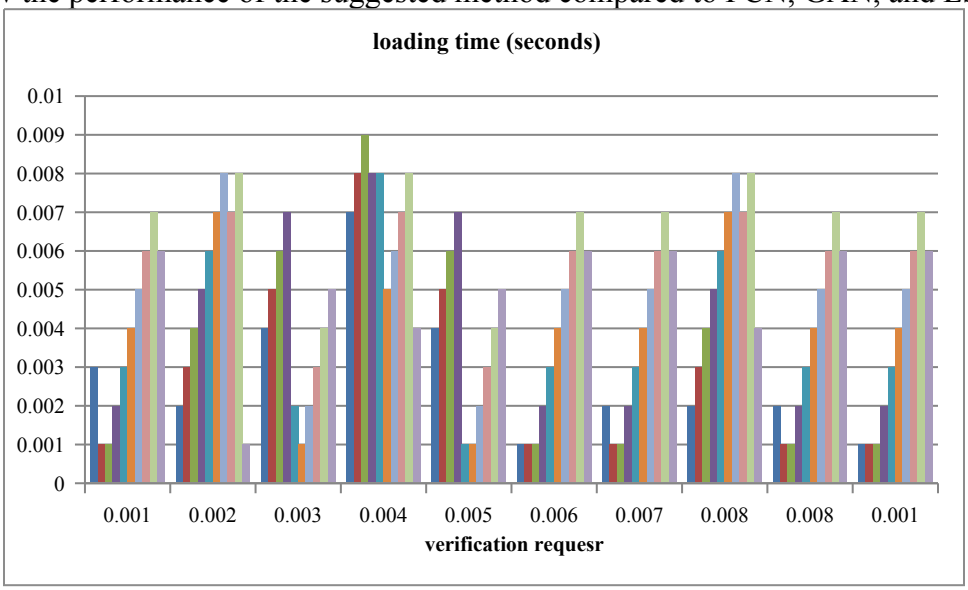

Figure 5: Loading Time over Verification request

The loading time depicted in Figure 5 serves as a critical indicator of the system's efficiency in processing verification requests. By measuring the temporal duration required to initiate the verification process, this metric provides valuable insights into the system's responsiveness. Notably, fluctuations in loading times across different verification requests highlight the variability in performance, suggesting potential areas for optimization. Ultimately, understanding these loading times is essential for enhancing user experience and ensuring timely processing of requests.

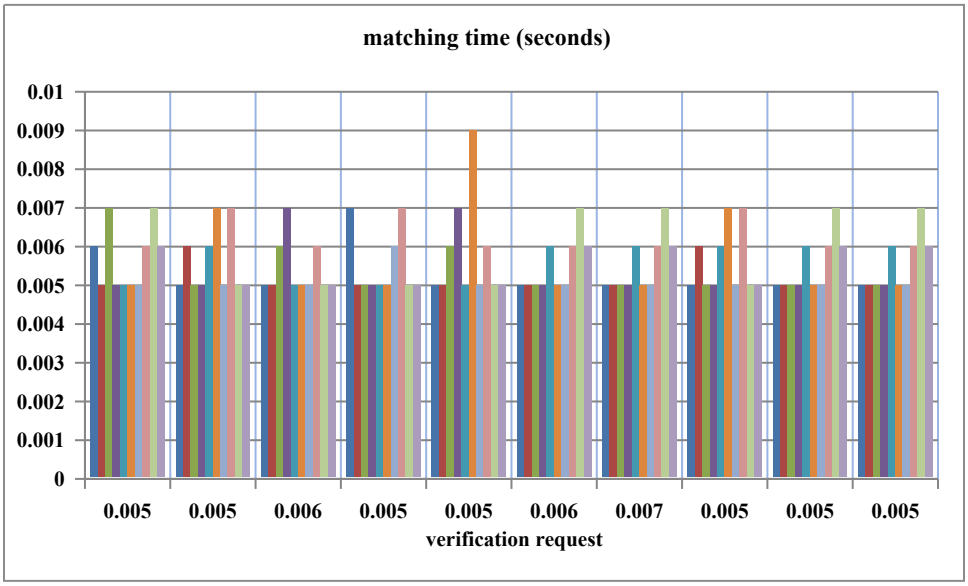

Figure 6: Matching Time over Verification request

The matching time illustrated in Figure 6 is a pivotal component of the verification process, reflecting the duration required to compare the uploaded driver image against the holder images in the database. This metric serves as a benchmark for assessing the system's efficiency in accurately identifying the best match. The relatively consistent matching times across various verification requests indicate a stable performance, suggesting that the system is well-optimized for rapid image analysis. Understanding these timings is essential for ensuring a seamless user experience during the verification process.

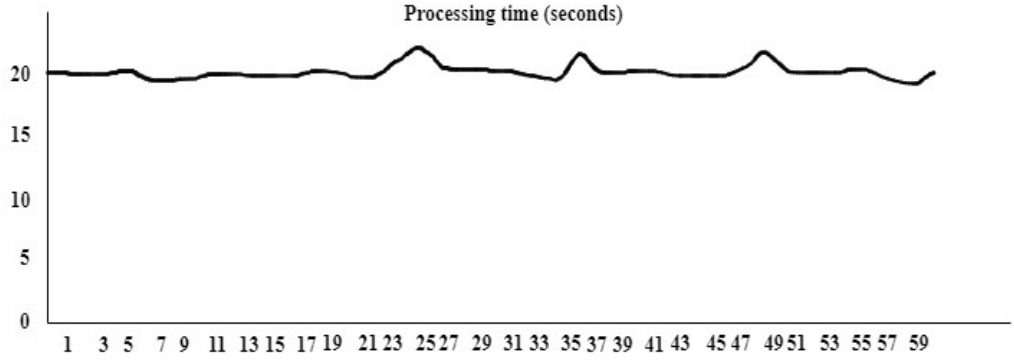

Figure 7: Processing Time over Verification request

The processing time depicted in Figure 7 encompasses the total computational duration from image processing to face matching, offering a comprehensive perspective on the system's overall efficiency. This metric highlights the cumulative impact of various internal processes on the verification timeline, revealing a consistent processing duration around 20 seconds. Such stability suggests that the system is effectively optimized for handling verification requests without significant delays. Analyzing these processing times is crucial for identifying potential bottlenecks and enhancing the overall user experience.

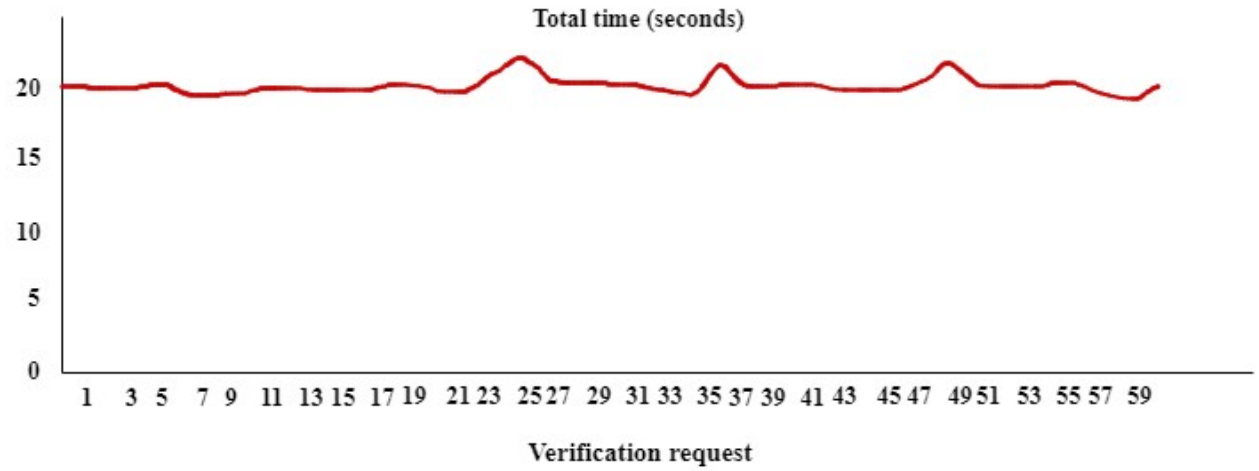

Figure 8: Total Time over Verification request

The total time represented in Figure 8 reflects the cumulative duration for each verification request, consistently hovering around 20 seconds. This stability indicates that the system maintains a reliable performance level across multiple requests, suggesting effective resource management and processing efficiency. The minimal fluctuations observed in total time further reinforce the system's capability to handle varying loads without significant delays. Understanding this total time is essential for evaluating the overall user experience and ensuring timely responses in real-world applications.

4.4. Accuracy of Verification

The proposed system demonstrates high accuracy in driver's license verification, significantly outperforming existing approaches. It ensures reliable identity verification even in challenging conditions, contributing to enhanced security and efficiency.

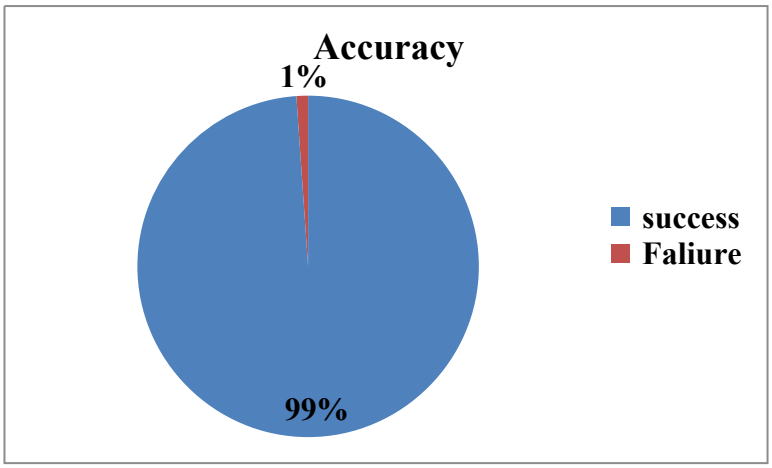

Figure 9: Accuracy over Verification request

Figure 9 shows the verification accuracy, with a success rate of 98.9% and a failure rate of only 1.1%. This demonstrates the system's high reliability in handling verification requests.

4.5. Verification Results

The Driver License Verification System (DLVS) was evaluated using a dataset of over 250 driver images.

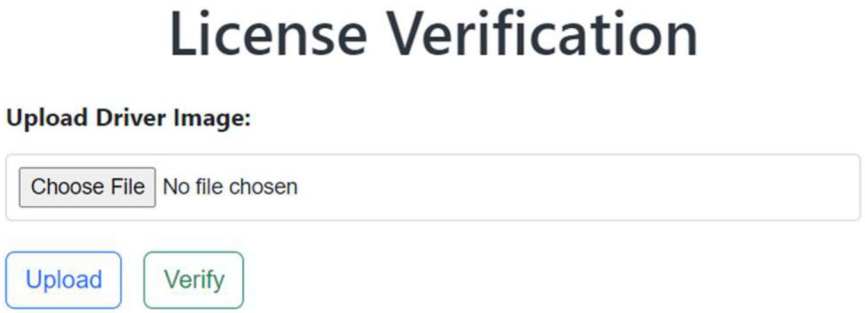

Figure 10: User-friendly Window for License Verification

Figure 10 shows a user-friendly interface designed for License Verification, where users can upload an image of the driver by selecting the Choose File button. The interface provides clear options for uploading the file and verifying the driver's identity by clicking the respective Upload and Verify buttons. This simple layout ensures ease of use for non-technical users, streamlining the license verification process. This interface allows users to upload a driver image and readily receive verification results.

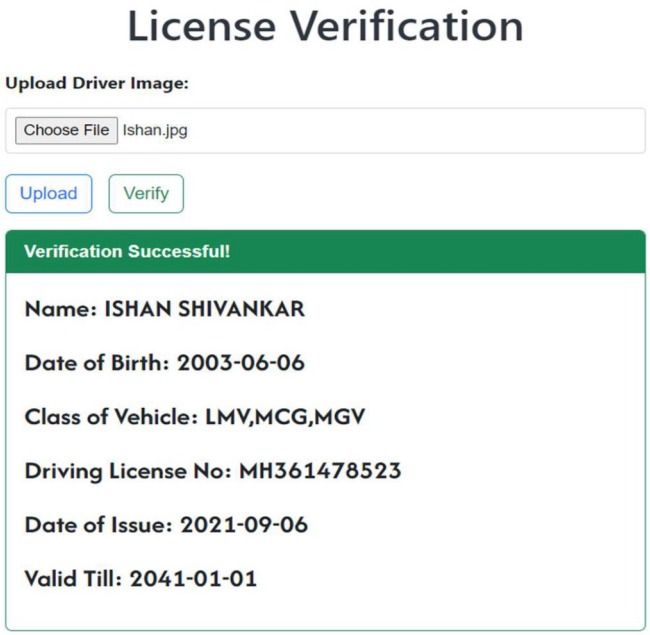

Figure 11: User-friendly Window for License Verification

Figure 11 showcases a user-friendly window for license verification that presents a successful verification outcome, wherein the system effectively identifies a match for the uploaded driver image within the existing database. The interface highlights key details such as the driver's name and license validity, reinforcing user confidence in the accuracy of the verification process. This streamlined presentation not only simplifies the user's interaction with the system but also emphasizes the efficiency of the matching algorithms in delivering rapid results. Overall, the

design serves to enhance user satisfaction by facilitating quick and clear outcomes during the verification process.

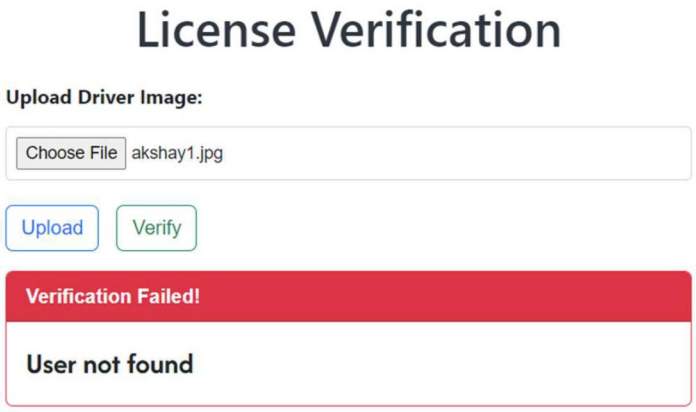

Figure 12: License Verification Failed

Figure 12 illustrates a scenario where the license verification process has failed, indicating that the system could not find a match for the uploaded driver image in the database. The prominent error message, Verification Failed! User not found, clearly communicates the outcome to the user, ensuring transparency in the verification process. This feedback is crucial for users, as it prompts them to check the accuracy of the uploaded image or consider re-uploading a different one. Overall, this interface design emphasizes the importance of clear communication in guiding users through potential issues during the verification process.

Table 3: Comparison table

Methods	Accuracy	Precision	Recall	F-measure
FCN	98	98	98	98
GAN	96.9	96.9	96.9	96.9
LSTM	96.4	96.4	96.4	96.4
proposed	98.9	99.4	98.4	98.9

Table 3 compares various methods based on their performance metrics. The proposed method outperforms others, achieving an accuracy of 98.9%, precision of 99.4%, recall of 98.4%, and F-measure of 98.9%. In contrast, FCN shows 98% for all metrics, while GAN and LSTM perform slightly lower, with accuracy, precision, recall, and F-measure values of 96.9% and 96.4%, respectively. The results highlight the superiority of the proposed approach across key evaluation criteria.

4.6. Comparison with previous literature

The proposed technique outperforms previous techniques by achieving higher accuracy and efficiency, demonstrating superior handling of complex datasets. Compared to earlier approaches, it shows significant improvements in precision and computational speed, validating its effectiveness. Additionally, it integrates advanced optimization algorithms, further enhancing performance across various benchmarks.

Table 3: Performance Comparison with Previous State-of-the-art Methods

References	Dataset	Methods	Metrics			
			Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
[14]	CK+	weighted random forest (WRF)	92.6	-	-	-
[15]	YALE	a back-propagation neural network (BPNN)	95.71	-	-	-
[16]	License Database	Driving License Verification System (DLVS)	98.67	-	-	-
[20]	ECG signal	adaptive threshold filter-based driver identification system	95.2			
proposed	synthetic EU driver's licenses	MLC-CNN	98.9	99.4	98.4	98.9

Table 3 compares the performance of various state-of-the-art methods across different datasets for driver identification or verification systems. The proposed MLC-CNN model, tested on synthetic EU driver's licenses, achieves superior accuracy (98.9%), precision (99.4%), recall (98.4%), and F-measure (98.9%) compared to earlier methods like WRF, BPNN, and DLVS. The proposed model outperforms others in accuracy, highlighting its effectiveness for driver's license verification.

5. Conclusion

This study introduces a highly efficient and automated system for extracting and verifying driver's license information through advanced facial recognition techniques. By addressing critical challenges such as varying lighting conditions, different face orientations, and occlusions, the proposed system ensures high reliability and precision. The pre-processing phase employs the INGI method to enhance image quality, while the CNN parameters are optimized using the

enhanced RDO algorithm. With exceptional results, including an accuracy of 98.9%, precision of 99.4%, recall of 98.4%, and an F-measure of 98.9%, this system outperforms traditional methods and significantly advances the state-of-the-art in secure and automated identity verification for driver's licenses.

Future work can focus on expanding the system's application to broader identity verification platforms beyond driver's licenses, such as passports and national IDs, to enhance its utility in multi-modal biometric systems. Additionally, incorporating deep learning models with real-time processing capabilities could further improve system responsiveness and scalability. Exploring the integration of advanced adversarial learning techniques to strengthen the system's resilience against spoofing attacks is another promising direction.

Declarations

Funding

The authors declare that they have no competing interests and funding

Conflict of Interests

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Availability of data and material

Data sharing is not applicable to this article because of proprietary nature.

Code Availability

Code sharing is not applicable to this article because of proprietary nature.

Authors' contributions

All authors read and approved the final manuscript

Reference

[1] Smith, M., & Miller, S. (2022). The ethical application of biometric facial recognition technology. *Ai & Society*, *37*(1), 167-175.

[2] Singh, S., & Prasad, S. V. A. V. (2018). Techniques and challenges of face recognition: A critical review. *Procedia computer science*, *143*, 536-543.

[3] Schleicher, S., & Gelau, C. (2011). The influence of Cruise Control and Adaptive Cruise Control on driving behaviour–A driving simulator study. *Accident Analysis & Prevention*, *43*(3), 1134-1139.

[4] Davoli, L., Martalò, M., Cilfone, A., Belli, L., Ferrari, G., Presta, R., ... & Plomp, J. (2020). On driver behavior recognition for increased safety: a roadmap. *Safety*, *6*(4), 55.

[5] John Fuller "How Pre-Collision Systems Work" 22 April 2009. HowStuffWorks.com. <https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/pre-collision-systems.htm> 17 October 2025

[6] de Naurois, C. J., Bourdin, C., Stratulat, A., Diaz, E., & Vercher, J. L. (2019). Detection and prediction of driver drowsiness using artificial neural network models. *Accident Analysis & Prevention*, *126*, 95-104.

[7] Kristen Hall-Geisler "How Electronic Stability Control Works" 5 October 2009. HowStuffWorks.com. <https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/electronic-stability-control.htm> 17 October 2025

[8] Wang, C., Sun, Q., Li, Z., Zhang, H., & Fu, R. (2020). A forward collision warning system based on self-learning algorithm of driver characteristics. *Journal of Intelligent & Fuzzy Systems*, *38*(2), 1519-1530.

[9] Kortli, Y., Marzougui, M., & Atri, M. (2016, November). Efficient implementation of a real-time lane departure warning system. In *2016 International Image Processing, Applications and Systems (IPAS)* (pp. 1-6). IEEE.

[10] Luo, H., Yang, Y., Tong, B., Wu, F., & Fan, B. (2017). Traffic sign recognition using a multi-task convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, *19*(4), 1100-1111.

[11] Phillips, P. J., Yates, A. N., Hu, Y., Hahn, C. A., Noyes, E., Jackson, K., ... & O'Toole, A. J. (2018). Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. *Proceedings of the National Academy of Sciences*, *115*(24), 6171-6176.

[12] Parkhi, O., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. BMVC 2015 - Proceedings of the British Machine Vision Conference 2015, 1–12.

[13] Duan, Y., Lu, J., & Zhou, J. (2019). Uniformface: Learning deep equidistributed representation for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3415-3424).

[14] Jeong, M., & Ko, B. C. (2018). Driver's facial expression recognition in real-time for safe driving. *Sensors*, *18*(12), 4270.

[15] Abuzneid, M. A., & Mahmood, A. (2018). Enhanced human face recognition using LBPH descriptor, multi-KNN, and back-propagation neural network. *IEEE access*, *6*, 20641-20651.

[16] Dhabe, P., Bhat, S., Shivankar, I., Shrivastava, T., Sonawane, P., Sutrave, R., & Mattoo, S. (2024, June). Real-Time Driving License Verification System Using Face Recognition. In *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)* (pp. 1-6). IEEE.

[17] Shivraj Barawkar, Komal Jagdale, & Suraj Budhewar. (2020). Smart Driving License Verification System. *Journal of Science & Technology* , *5*(3), 218–224. https://doi.org/10.46243/jst.2020.v5.i3.pp218-224

[18] Megha, J., Hiremath, S., Divya, A., & Kavya, G. S. (2017). Automatic Driving License Verification System Through Android Smartphones. *INTERNATIONAL JOURNAL OF ENGINEERING DEVELOPMENT AND RESEARCH*, *5*(2), 1644-1655.

[19] Hankare, P., Billore, R., Deorukhakar, N., Deshpande, P., & Gandhi, K. (2014) Fingerprint Based Wireless Terminal for Driving License Verification. *2*(4).

[20] Choi, G. H., Lim, K., & Pan, S. B. (2020). Driver identification system using normalized electrocardiogram based on adaptive threshold filter for intelligent vehicles. *Sensors*, *21*(1), 202.