Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

Driver Drowsiness Detection Based on Convolutional Neural Network Architecture Optimization Using Genetic Algorithm

Yashar Jebraeily[1], Yousef Sharafi[1], Mohammad Teshnehlab[1]

[1] Faculty of Electrical and Computer Engineering, Intelligent Systems Laboratory, K. N. Toosi University of Technology, Tehran, Iran.

Corresponding author: Yousef Sharafi (e-mail: y.sharafi@srbiau.ac.ir).

ABSTRACT In today's analysis of traffic accident reports it becomes evident that most driving accidents result from driver drowsiness, fatigue, and lack of alertness. At these moments, drivers cannot react quickly to such changes in their state. Characteristics of drowsy driving include closed eyes, semi-open eyes, and yawing. In this research, a network architecture search mechanism has been used to obtain an optimal structure for a convolutional neural network that can effectively detect driver drowsiness. The contributions in this research include the following: Firstly, to enrich the dataset related to detecting driver drowsiness and address shortcomings in existing datasets, several videos showcasing various drowsy states have been extracted, and their frames have been labeled as either drowsy or natural. Secondly, an optimal convolutional neural network structure, including the number of layers, type of objective function, etc., has been obtained using a genetic algorithm, with FER-2013 used for achieving the optimal structure. Thirdly, transfer learning is employed, where the optimized network obtained from the genetic algorithm is considered a feature extraction component, and its fully connected layer is trained for the drowsy dataset. The proposed method outperforms other approaches in accuracy and precision, achieving an accuracy rate of approximately 99.8%.

INDEX TERMS Driver Drowsiness Detection, Convolutional Neural Network (CNN), Neural Architecture Search, Genetic Algorithm.

I. INTRODUCTION

The effects of fatigue and drowsiness on human health and quality of life are essential and significant topics. This fatigue and sleepiness can result from various issues, such as staying awake at night for exams, events, or daily life problems. However, their effects are ultimately the same, leading to discomfort and risks in everyday life, work, education, and driving [1]. A United States Department of Transportation report reveals that numerous car accidents happen yearly because of drowsiness. This statistic alone cannot be considered a definitive measure, as some accidents go unreported to the police, and some accidents exact causes of drowsiness are undetermined. Despite efforts to prevent them, there have been reports of 109,000 injuries and 6,400 fatalities resulting from these accidents. Researchers say these statistics may indicate a much larger truth [2,6]. The financial damages caused by drowsy driving-related accidents in the United States have exceeded $109 billion annually [3]. The effects of driving while drowsy can be similar to driving under the influence of alcohol or drugs. Drowsy drivers may experience delayed reactions, impaired judgment, and decision-making, resulting in hazardous incidents like drifting or ignoring road signs.

Recognizing signs of fatigue while driving, such as yawning, nodding, and difficulty keeping your eyes open, is essential. To prevent driving in drowsy conditions, it is advised to pull over to a safe location and rest until feeling more alert, without delay. Taking regular and sufficient sleep, refraining from driving during hours when drowsiness is more likely, and avoiding alcohol or medications that may increase drowsiness symptoms are smart precautions. These safety measures can help us avoid driving while tired and keep ourselves and others safe. As technology progresses, road safety experts are utilizing advanced technologies to minimize accidents on the road. With artificial intelligence and advanced sensors, these technologies provide solutions to help drivers prevent the dangers of driving while drowsy. In the realm of enhancing driver safety and preventing accidents due to drowsiness, four innovative devices have emerged as crucial components of an integrated approach. These devices include Detection Systems for Driver Alertness, Wearable Devices, Head-up Displays (HUDs), and Drowsiness Alarms. The Detection Systems for Driver Alertness form the frontline defense, incorporating an array of sensors like cameras and steering sensors. These sensors work in tandem to identify subtle signs of drowsiness, such as erratic driving patterns and changes in eye movements. When these signs are detected, the system promptly sends alerts, be it through sounds or vibrations, serving as a proactive warning to the driver. Complementing this, Wearable Devices in the form of smartwatches and fitness trackers provide a continuous monitoring mechanism for the driver's vital signs. By closely tracking metrics like heart rate and biometric data, these devices can discern sudden and abnormal decreases in the driver's heart rate, triggering alerts that signify potential drowsiness. In the driver's line of sight, Head-up Displays (HUDs) project essential information onto the windshield. These displays not only offer real-time data on speed and navigation routes but also serve as a means to detect drowsiness. By observing the driver's eye movements, the system can identify signs of fatigue and respond promptly to ensure the driver remains attentive. As a last line of defense, Drowsiness Alarms are strategically positioned on the driver's head or attached to the steering wheel. Equipped with specialized sensors, these devices detect forward head movements, a classic indication of drowsiness. Upon identification of such movements, a loud alarm is activated, jolting the driver into alertness and preventing potential accidents. This interconnected approach, weaving together the strengths of each device, creates a comprehensive system that not only detects but actively addresses driver drowsiness. By leveraging a combination of technologies, this integrated solution aims to significantly enhance overall driver safety and reduce the risks associated with fatigue-related incidents on the road.

It's important to remember that while technologies can prevent drowsy driving, they are not entirely reliable. Drivers should maintain good driving habits, including getting sufficient sleep and taking breaks during long trips. In the literature, various research methods have been used to prevent accidents by implementing different technologies. Presented below is a review of some of these methods. Reference [4] offers an IoT-enabled wristband device approach for detecting driver drowsiness through machine learning. This approach analyzes blood pulse signals and facial skin conductance. The detection process involves testing multiple machine-learning algorithms on data collected from 9 participants. A low-cost system was developed to detect drowsiness using a simple camera through the application of machine vision and machine learning in a study [5]. This technique utilizes sleepiness rules based on blinking patterns in neurology. In the results presented in [5], No alerts were issued for an awake person, but an average of 16.1 alerts were issued for drowsy people with 94.44% accuracy. In [6], A deep learning-based method for detecting drowsiness has achieved over 80% accuracy and has been effectively utilized in high-accuracy Android applications. The proposed method compresses the main model into a lightweight model.

A fatigue detection system based on video analysis of drivers is presented in [7]. The system uses various visual signals to indicate the driver's readiness level. To detect fatigue, parameters such as the duration of eye closure and instances of yawning are taken into consideration. A fuzzy system classifies the driver's real-time status based on their eyes and mouth data [7]. Knapik and colleagues proposed a model for detecting driver fatigue using thermal images and detecting yawning without interfering with the driver in the daytime and nighttime conditions [8] Due to the rapid development of mobile internet, the fast growth of wireless network technologies, and the Internet of Things (IoT), fully autonomous vehicles have made significant progress and partially achieved the possibility of fully autonomous driving [9]. Two indicators can be used to identify drowsy individuals: physiological and behavioral. Physiological indicators such as heart rate variability, respiration rate, head movements, and eye-related behaviors (duration, frequency, and time of eye closure) are considered. Furthermore, driving behavior can be indicated by factors such as time to cross lanes, speed, steering angle, and lane positioning., are focused on [10,22].

Lee and his colleagues have developed an LSTM-CNN model that detects drowsiness using electroencephalography (EEG) data. The EEG dataset was obtained by exposing 19 subjects to random auditory stimuli and recording their button press responses. The LSTM-CNN model achieved an average accuracy of 85.6% and a Kappa index of 0.77 for a three-class classification problem. The proposed model was used for binary classification of natural awareness (awake) and reduced awareness (drowsiness and sleep) [11]. In a different study, researchers analyzed a driver's physiological state using a wristband device that measured skin conductance [12]. Blink counting, mouth opening, and closing are commonly used in research to detect drowsiness. When the driver's eyes are closed for an extended period, an alert sound notifies them [13,14]. The most effective method for preventing driving accidents is by detecting driver drowsiness and issuing alerts before they fall asleep. One can detect drowsiness in drivers by analyzing important facial features in an image [15]. In [16], A driver drowsiness detection system with 99.8% accuracy was developed using a convolutional neural network architecture based on behavior cues.

In [17], An LSTM autoencoder architecture was used for drowsiness detection, with ResNet-34 as the feature extractor. In another study, researchers achieved 97% accuracy in detecting drowsiness using the MobileNet model on the MRL eye dataset [18]. Recently, YOLO-based models have been utilized in various classification applications. According to reference [19], YOLO was implemented to detect the driver's face and eyes using the Dlibs library, and FFV was used as a metric to detect the driver's state. In reference [20], A model that uses a convolutional neural network to extract facial image information and classify it into drowsy and awake categories is presented. The model presented performed better than networks based on transfer learning, such as VGG-16 and ResNet-50. In [21], A method combining Inception, LSTM, VGG, and DenseNet networks extracts eyelid movements and blinking features to evaluate driver drowsiness. Detecting driver drowsiness through EEG signals is another effective method [23]. In transfer learning networks, a pre-trained deep convolutional neural network model detects driver drowsiness based on facial features [24]. In another model, Machine vision techniques, including edge detection, grayscale conversion, and scattering, are used with convolutional neural networks to detect the driver's facial state [25]. In the study [26], a small deep-learning network focused on VGG16 and VGG19 networks, and a multi-sensor system was used to track and identify driver drowsiness. In another research, researchers utilized ResNet50 for driver drowsiness detection, considering the usual eye closure time between 100 and 300 milliseconds. When the driver's eyes remain closed for more than 300 milliseconds, this condition is considered driver drowsiness [27]. Steering angle and departure frequency are features utilized for driver drowsiness detection [28]. In the study [28], three models, VGG16, InceptionV3, and Xception, were employed to analyze the driver's eye features in both awake and drowsy states. Driver drowsiness was detected by observing the eye-closure features from recorded videos. The results indicated that the Xception model achieves higher accuracy in detecting driver drowsiness [29]. In another research, the VGG-16 architecture was used for detecting the driver's eye features in drowsy and awake conditions, achieving an accuracy of 96.79% for training and testing data [30]. In [31], a method based on Multifractal Detrended Fluctuation Analysis (MF-DFA) has been introduced for the identification of fatigue resulting from long-duration driving. In this study, specific subgroups of electroencephalogram (EEG) signals from individuals (θ: 4 ~ 7 Hz, β: 14 ~ 32 Hz) have been extracted. Additionally, various features of the multifractal spectrum, including the oscillation function, mass exponent, Hurst exponent, spectral width values, and symmetry, have been precisely examined. The results of this study encompass a comparative analysis of individuals in different driving conditions. The findings reveal significant variations in the values of the Hurst exponent, spectral width, and multifractal spectrum symmetry corresponding to different driving durations. Moreover, in comparison to conventional fatigue detection methods, the MF-DFA method has demonstrated itself as a more effective approach for capturing fatigue-related features. Methods for assessing drowsiness are not solely summarized by examining facial states. In [32], the investigation initially categorizes methods for assessing drowsiness into three groups: behavioral, vehicular, and physiological parameters. In the second stage, a scrutiny of the top supervised learning techniques for detecting drowsiness has been conducted. In another research study, a comprehensive approach for driver identification and authentication using deep learning has been presented. This approach utilizes psychophysical data through a driving simulator and eye-tracking system. It employs a model with a Fully Convolutional Network (FCN) and a Squeeze-and-Excitation (SE) block. The proposed architecture analyzes three-second segmented windows of information to extract unique driving features. In the task of identifying 15 drivers in various conditions, the model achieved an accuracy of 99.60%. For driver authentication, the suggested architectural model is combined with a Siamese neural network, yielding high accuracy, retrieval, and F1 score [33]. In some cases, it is necessary to examine the driver's response to signs of drowsiness, which is one of the most critical factors in traffic accidents, especially when unexpected visual stimuli may lead to accidents. In [34], using a driving simulator, an investigation has been conducted into how drivers react to such violations with different backgrounds. Studies indicate that novice drivers may not respond promptly to their situations, leading to collisions with other road users. Furthermore, some novice drivers allocate time more to looking forward and less to assessing the surrounding environment. These findings serve as inspiration for the development of advanced driving support systems in the future. Some gadgets have been developed to alert drivers to drowsiness, and one of them is the FDWatch system [35]. FDWatch is an innovative drowsiness detection system that overcomes the limitations of existing solutions by utilizing low-cost Photoplethysmogram (PPG) sensors and integrated motion sensors in wristbands. This system employs advanced algorithms to extract various drowsiness-related indicators, including detecting nodding behavior using common PPG sensors. Another type of work carried out in the field of driver drowsiness detection involves combining facial features (blinks) and examining vital signals, presenting an innovative approach by merging frontal Electroencephalography (EEG) and machine vision techniques [36]. The quality of driver fatigue detection has been enhanced using a machine vision approach based on the percentage of eye closure (PERCLOS) and EEG signals. In [37], an Internet of Things (IoT)-based monitoring system for real-time detection of driver drowsiness has been introduced. This system comprises three levels of drowsiness detection to monitor the driver and provide alerts if necessary. The process begins with alcohol consumption detection as an initial safety measure, and if alcohol is not detected, the system proceeds to facial recognition. At Level 1, if the relative eye closure ratio is below a threshold, an auditory alert system is activated. At Level 2, if the auditory alert persists for more than two consecutive times, a human auditory alert system is activated. At the final level, a notification with GPS coordinates is sent to the driver's owner or any relevant person. Infrared light is used for drowsiness detection during nighttime.

The contributions of this research are summarized below:

- Creating a new dataset for detecting driver drowsiness during driving.
- A method for obtaining an optimal structure for convolutional neural networks in terms of the number of layers and other parameters is proposed.
- Using transfer learning to detect driver drowsiness.
- Comparing the proposed methods with other commonly used convolutional neural networks, including AlexNet, ResNet, GoogleNet, etc.

The paper is structured as follows: The second section discusses using genetic algorithms to optimize convolutional neural network parameters represented by chromosomes. In this research, the third section provides a brief overview of the transfer learning networks that were utilized. In the fourth section, we explore the neural network's training process. Here, we use a vast dataset of 35,887 images to determine the ideal network structure, including the number of layers, activation functions, and more. Afterward, we apply the transfer learning approach, utilizing the features extracted from the source network to train the final network on data related to drowsiness. The fifth section presents the proposed method's evaluation results compared to other convolutional neural networks and discusses the findings. Finally, the conclusion of the report is provided.

II. Genetic Algorithm

In this research, a genetic optimization algorithm has been utilized to optimize the architecture of convolutional neural networks. The genetic algorithm, as an evolutionary optimization method, has the capability to enhance the architecture of deep neural networks. By employing the genetic algorithm, an optimal architecture is determined for convolutional neural networks. Then, through fine-tuning the drowsy driving dataset, the performance of the deep neural network is improved.

The genetic algorithm is one of the most well-known meta-heuristic algorithms that operates based on the concepts of natural selection, mutation, and evolution. This algorithm is used to solve various optimization problems in different scientific domains. Here is a brief overview of the stages involved in the genetic algorithm.

1. Initial Random Population: The algorithm creates a random population in the search space.
2. Population Evaluation: For each solution in the population, the objective function of the problem is computed to determine the solution's fitness. In an optimization problem, the objective function is either maximized or minimized.
3. Crossover and Mutation Operators: These two operators generate new solutions in the search space.
4. Selection Operator: In this stage, using the selection operator, superior solutions are chosen for the next generation. Solutions with better objective function values have a higher probability of being selected to be part of the next generation.
5. Algorithm Termination Conditions: The genetic algorithm iterates until it reaches termination conditions (step 2). Termination conditions can include maximum generations, maximum objective

function evaluations, or achieving a specific level of accuracy in the objective function.

6. Optimal Solution: The best solution among all members of the population is considered the optimal solution.

The overall process of the genetic algorithm is depicted in Figure 1.

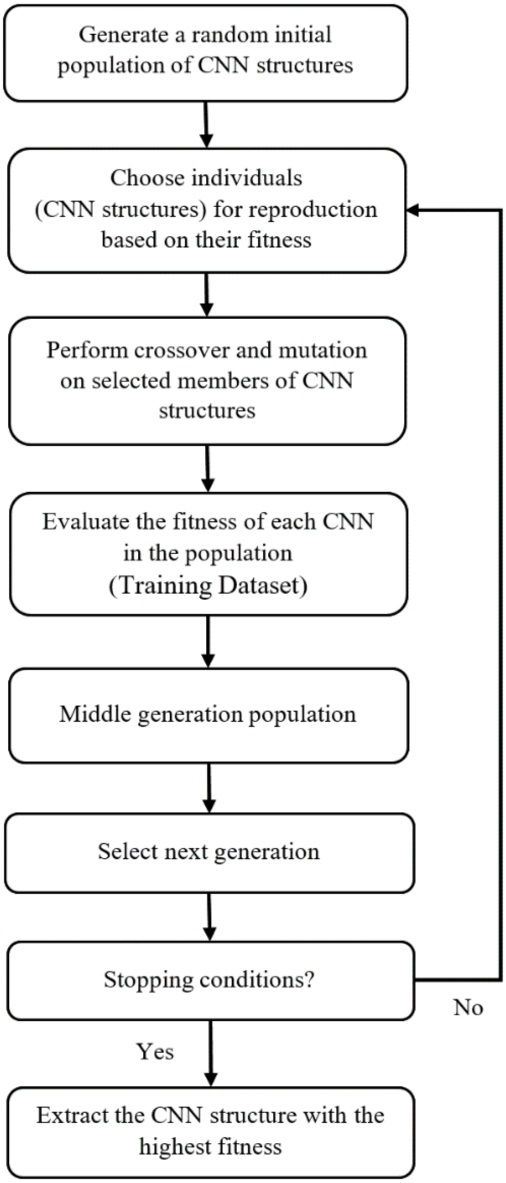


FIGURE 1. **Genetic algorithm optimization process.**

III. Deep Transfer Learning

Transfer learning networks such as ResNet, GoogleNet, AlexNet, and other architectures are considered among the most important architectural structures in deep neural networks for image classification. Convolutional networks are highly regarded for their unique features and ability to achieve excellent image processing performance, especially in image classification, due to their large convolutional layers.

Transfer learning networks are initially trained as a base model and learn general image features from large-scale datasets like ImageNet. Then, these learned features are used as input for smaller networks, which accelerates the training process. By employing transfer learning networks, the need for a large amount of training data is reduced. This means that efficient models can be trained with fewer training images. Utilizing pre-trained features from transfer learning networks typically leads to improved accuracy and performance in image processing tasks.

Given these advantages, transfer learning networks are prevalent and effective in research and practical applications in image and video processing today. In this study, the results of the proposed method are compared with several transfer learning networks, the architectures of which are briefly described below.

A. *ResNet*

ResNet-50 is one of the popular architectures in object recognition and image classification using deep neural networks [38]. This architecture is derived from the ResNet family. These networks are used to address the gradient vanishing problem in deep networks, and their development has led to significant advancements in image classification. In Figure 2, the structure of the ResNet-50 network is depicted.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381999

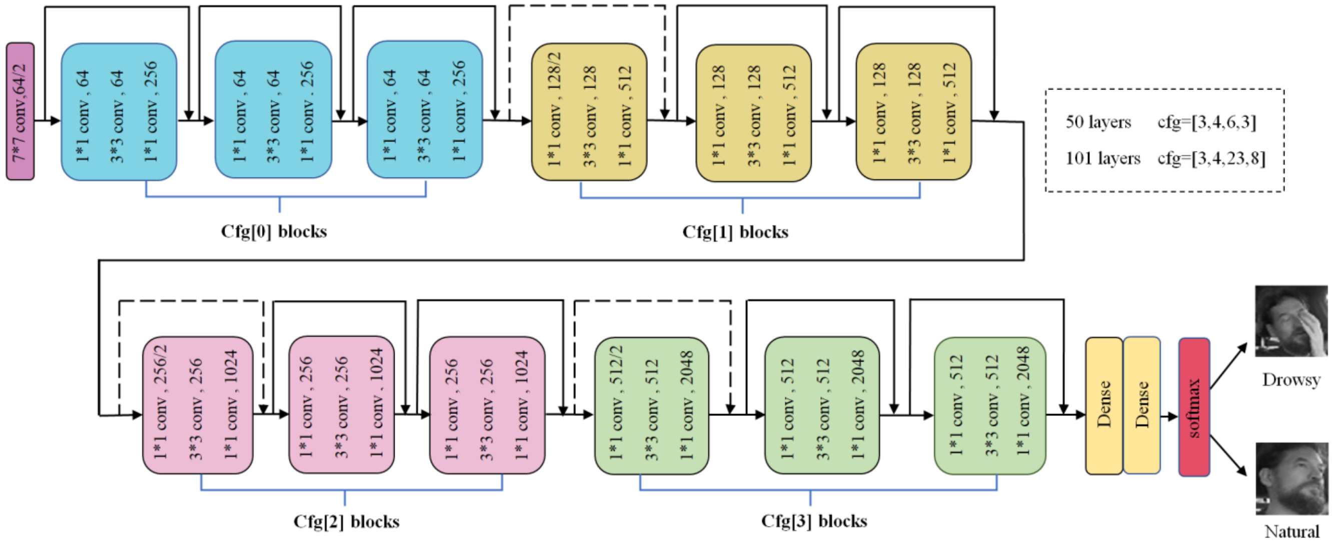


FIGURE 2. **Structure of the ResNet-50.**

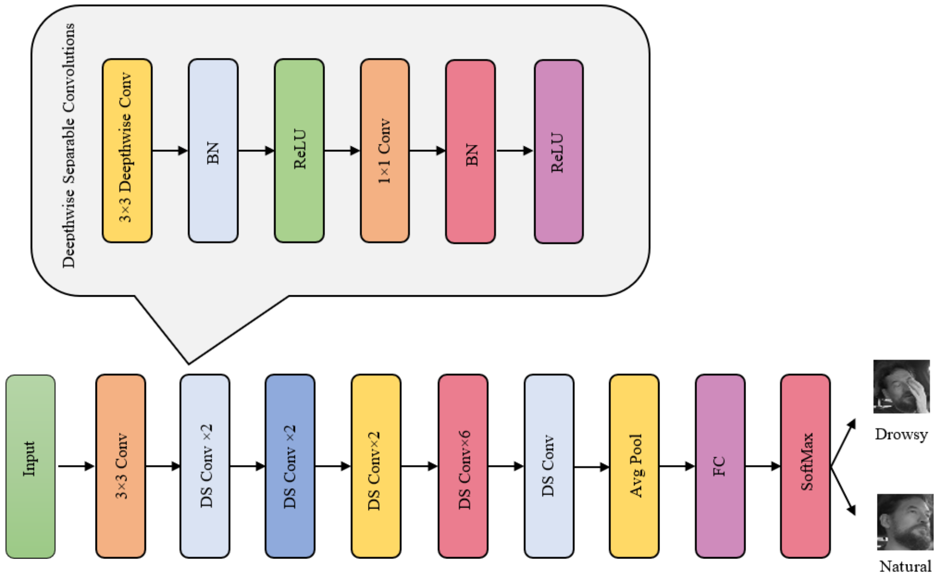


FIGURE 3. **Structure of the MobileNet.**

B. MobileNet

MobileNet is a convolutional neural network architecture with a highly appealing structure [39]. One of the significant advantages of MobileNet is its efficiency in resource-constrained environments. This network can run on devices with limited memory and processing power, using fewer parameters than other networks. MobileNet is a suitable choice for execution on smartphones and mobile devices with constrained resources. This feature highlights the increased importance of MobileNet in applications and devices that require fast and efficient image processing. Figure 3 illustrates the structure of the MobileNet network.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381999

C. VGG

VGG is another famous architecture in computer vision and image processing [40]. This architecture, with its small convolutional layers and many neural network layers, excels in detecting various components within images. It is used as a valuable tool in object recognition and image capture. The Geometry Group research team at the University of Oxford presented the VGG architecture. Figure 4 illustrates the structure of the VGG network.

D. GoogleNet

The GoogleNet architecture is one of the highly successful architectures in machine vision and image processing [41]. It was developed by Google's research team to enhance the performance of deep neural networks in object recognition within images. One prominent feature of GoogleNet is its use of complex structures known as "Inception modules," which allow information to be extracted in parallel from various image features. Figure 5 illustrates the structure of the GoogleNet network.

IV. Proposed Method

A. Designing an Optimal CNN Structure Based on a Genetic Algorithm

Designing and selecting an appropriate structure for a convolutional neural network always comes with challenges that must be carefully managed. Several influential factors in the design of CNNs can directly impact their performance, including the number of layers, filter sizes, the number of neurons in each layer, the amount of training data, and hardware settings. Inappropriate choices for these factors can lead to overfitting, reduced performance, or other issues. In recent research, various methods have been used to obtain the structure of convolutional neural networks, one of which is evolutionary optimization algorithms. In this study, as depicted in Figure 6, the genetic optimization algorithm has been utilized to obtain an appropriate structure for convolutional neural networks.

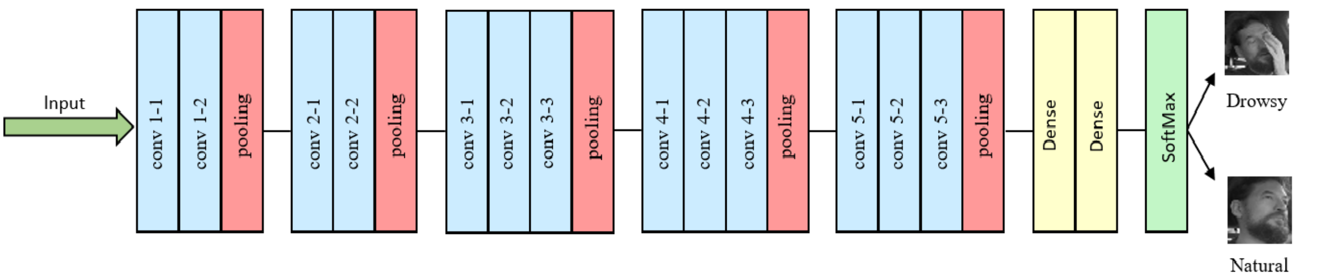


FIGURE 4. **Structure of the VGG.**

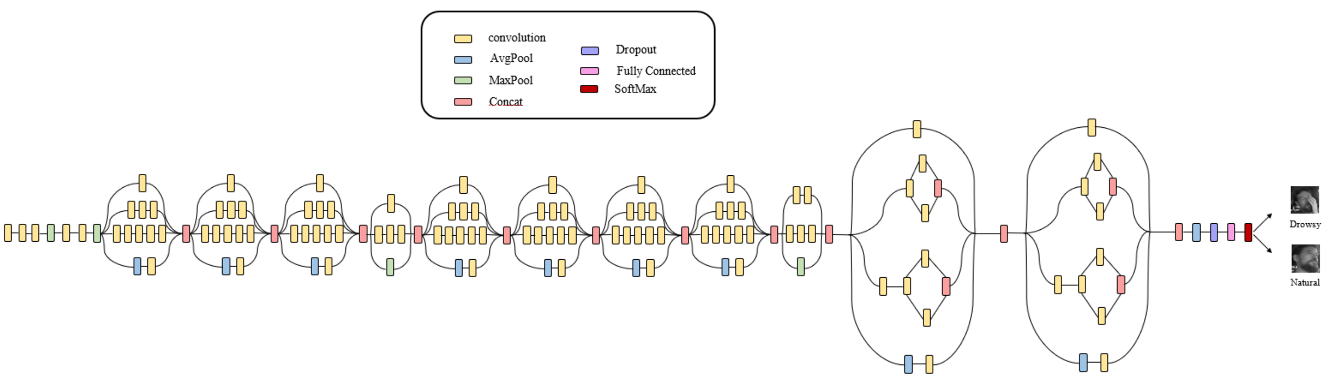


FIGURE 5. **Structure of the GoogleNet.**

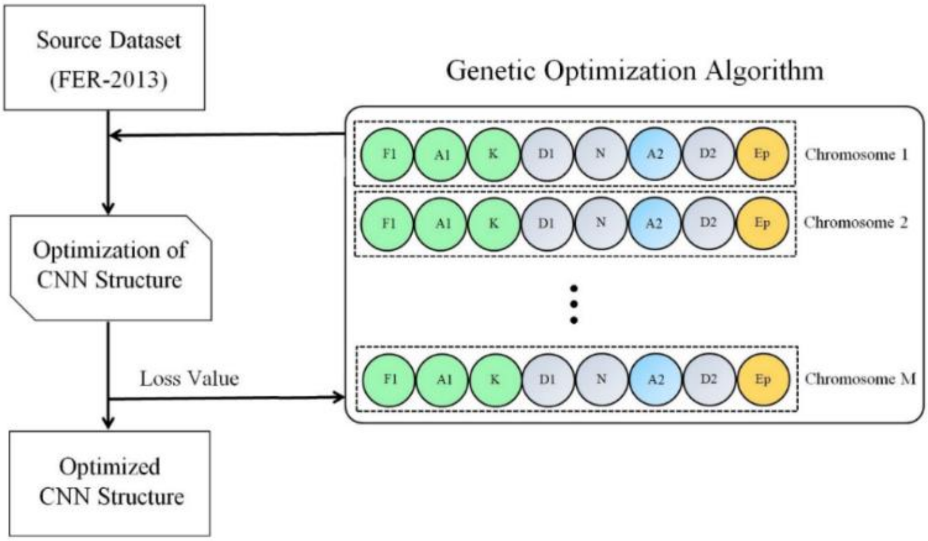


FIGURE 6. **Optimization of CNN network structure using genetic algorithm.**

To determine the appropriate number of layers and the structure of a convolutional neural network (CNN) for detecting driver drowsiness, we first obtain the optimal CNN structure using a genetic algorithm in an evolutionary approach based on the FER-2013 dataset. The FER-2013 dataset consists of 35887 images categorized into six classes. To better understand the optimization process in the genetic algorithm for obtaining the optimal CNN structure, as depicted in Figure 7, a chromosome in the genetic algorithm is defined as the parameters of a convolutional layer and a fully connected layer. As seen in Figure 7, these parameters include the number of filters in the convolutional layer (F1), kernel size (K), dropout rate (D1), the activation function of the convolutional layer (A1), the number of neurons in the dense layer (N), the activation function of the dense layer (A2), dropout rate (D2), and the number of epochs for network training (Ep). Together, these parameters form the chromosome and the genetic algorithm is used to optimize them. A CNN with one convolutional layer and one fully connected layer is initially designed based on the FER-2013 dataset using the genetic algorithm to obtain its optimal structure. Subsequently, as shown in Figure 10, while keeping the structure of the first convolutional layer fixed, a second convolutional layer is added along with another fully connected layer. Again, the optimal structure is determined using the genetic algorithm with the FER-2013 dataset. As illustrated in Figure 10, this process continues until no improvement in the performance of the CNN is observed by adding a convolutional layer.

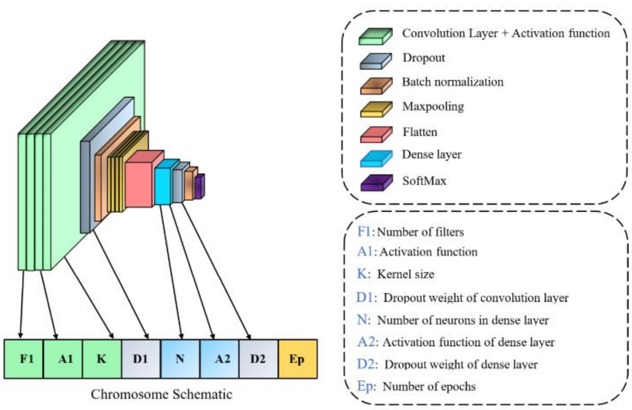


FIGURE 7. **Chromosome definition in the genetic algorithm for CNN structure optimization.**

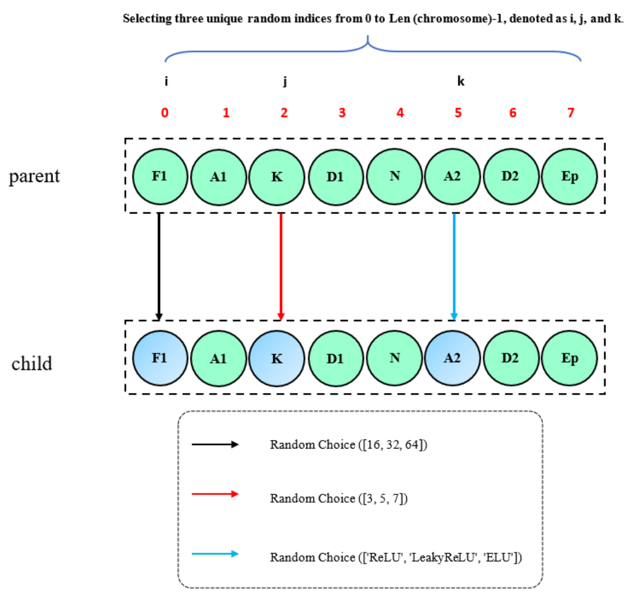


FIGURE 8. **Sample of a mutation performed for a parent**

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381999

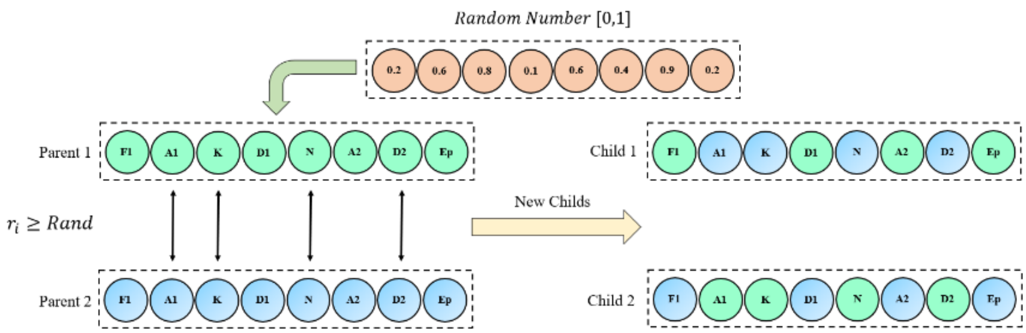


FIGURE 9. **Sample of a crossover performed for two parents**

In the genetic algorithm, two operators, mutation and crossover, play a crucial role in achieving the optimal solution. In this study, both uniform mutation and uniform crossover operators have been used. The pseudo-codes related to mutation and crossover are provided in Algorithms 1 and 2, respectively. The proposed mutation method in the genetic algorithm, as shown in Algorithm 1, is performed by randomly selecting three indices from a solution. Based on the indices chosen, a change is made to the solution, A performed sample of this algorithm is illustrated in Figure 8. Additionally, the uniform crossover method has been employed in this research to combine two solutions. In this method, each corresponding pair of genes from the parents is considered separately. Then, a random number between 0 and 1 is generated for each pair. If this random number is less than 0.5, the gene corresponding to the first parent is selected and transferred to the child. However, if the random number is equal to or greater than 0.5, the gene corresponding to the second parent is chosen and transferred to the child. Uniform crossover is a simple and effective method for generating diversity in the genetic algorithm population and greatly aids in improving the algorithm's performance. A performed sample of this algorithm is illustrated in Figure 9.

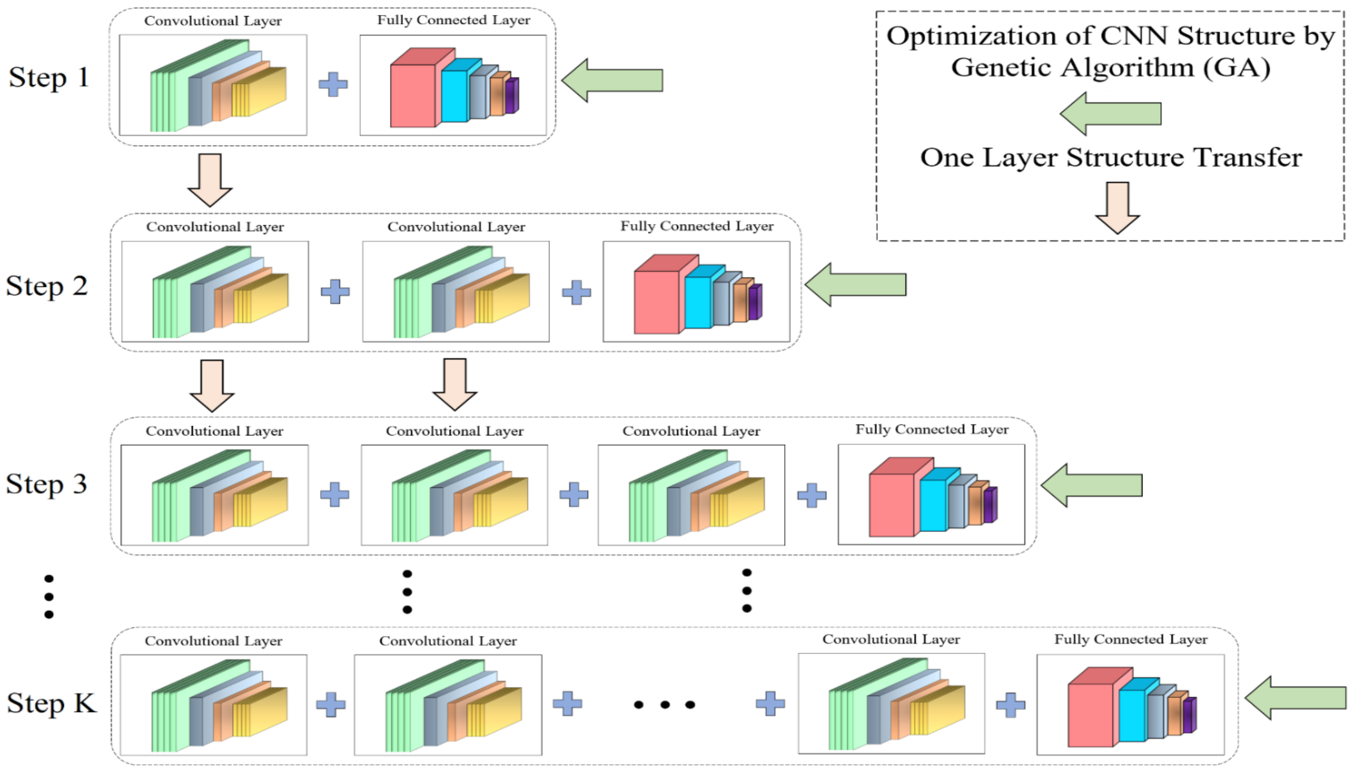


FIGURE 10. **Optimization process of CNN structure using genetic algorithm.**

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381999

Algorithm 1: Mutation Operator for CNN Architecture Optimization

```
Initialization: Mutation Rate
Input: Solution. // Current Architecture
// Create a copy of the Solution called MutatedArchitecture.
MutatedArchitecture = Solution
//Selecting three unique random indices from 0 to Len (MutatedArchitecture)-1, denoted as i, j, and k.
IndexList = [i, j, k]
For each index in IndexList
    Switch(index)
        Case 0:
            //Change of number of filters.
            MutatedArchitecture [index] = Choice ([16, 32, 64])
        Case 1:
            //Change of number of neurons.
            MutatedArchitecture [index] = Choice ([128, 256, 512])
        Case 2:
            //Change of kernel size.
            MutatedArchitecture [index] = Choice ([3, 5, 7])
        Case 3:
            //Change of activation function of convolution layer.
            MutatedArchitecture [index] = Choice (['ReLU', 'LeakyReLU', 'ELU'])
        Case 4:
            //Change of activation function of dense layer.
            MutatedArchitecture [index] = Choice (['ReLU', 'LeakyReLU', 'ELU'])
        Case 5, 6:
            //Change of dropout weight of convolution layer or dense layer.
            //Generate a random value for "b" in range [−0.2, 0.2].
            b = Uniform Random[−0.2, 0.2]
            MutatedArchitecture [index] = |MutatedArchitecture [index] + b|
            //Checking the range of the value of MutatedArchitecture [index] in [0.1, 0.5].
        Case 7:
            //Change of number of Epochs.
            //Generate a random value for "r" in range [0, 1].
            If r ≤ 0.5 then
                TempValue = RandomInteger[−5, 5]
                MutatedArchitecture [index] = MutatedArchitecture [index] + TempValue
                //Checking the range of the value of MutatedArchitecture [index] in [100, 250].
            End if
    End switch
End for
Output: MutatedArchitecture
```

Algorithm2: Crossover Operator for CNN Architecture Optimization

```
Input: parent1, parent2
  // initialize empty lists for the two CNN structures
  child1 = []
  child2 = []
  // Iterate over each gene position in the parents
  for i from 0 to length(parent1) - 1 do:
    // Randomly decide whether to inherit the gene from
       parent1 or parent2
    if random () < 0.5 then:
       // Inherit the gene from parent1 for child1
       child1[i] = parent1[i]
      // Inherit the gene from parent2 for child2
      child2[i] = parent2[i]
    else:
      // Inherit the gene from parent2 for child1
      child1[i] = parent2[i]
      // Inherit the gene from parent1 for child2
      child2[i] = parent1[i]
  // Return the resulting CNN structures
  return child1, child2
```

B. Generating a Dataset for Driver Drowsiness Detection

In this study, to have a good CNN model for detecting driver drowsiness, we require a dataset with appropriate diversity covering all possible states of a driver, including normal, fatigue, and drowsy conditions during driving. The following steps outline the dataset creation process:

1. **Collecting and Selecting Videos**: Initially, videos related to the natural, fatigue, and drowsiness states of drivers are gathered from various sources.
2. **Image Framing**: The videos are converted into frames, turning the video into multiple images captured at different moments. These images depict the natural fatigue and drowsiness states of the driver.
3. **Face Region Detection**: In this stage, facial detection algorithms are utilized to identify important facial regions such as eyes, mouth, and cheek areas. This step focuses on regions relevant to fatigue and driver drowsiness.
4. **Color Conversion and Image Resizing**: The extracted images are first converted to grayscale. This conversion reduces the dimensionality of the data and helps focus on important features. Subsequently, the dimensions of the images are resized to 48 by 48 pixels for easier and cost-effective computations.
5. **Image Labeling**: Images are accurately labeled based on the natural fatigue and drowsiness states of the driver.

The stages of creating the drowsiness dataset are shown in Figure 11.

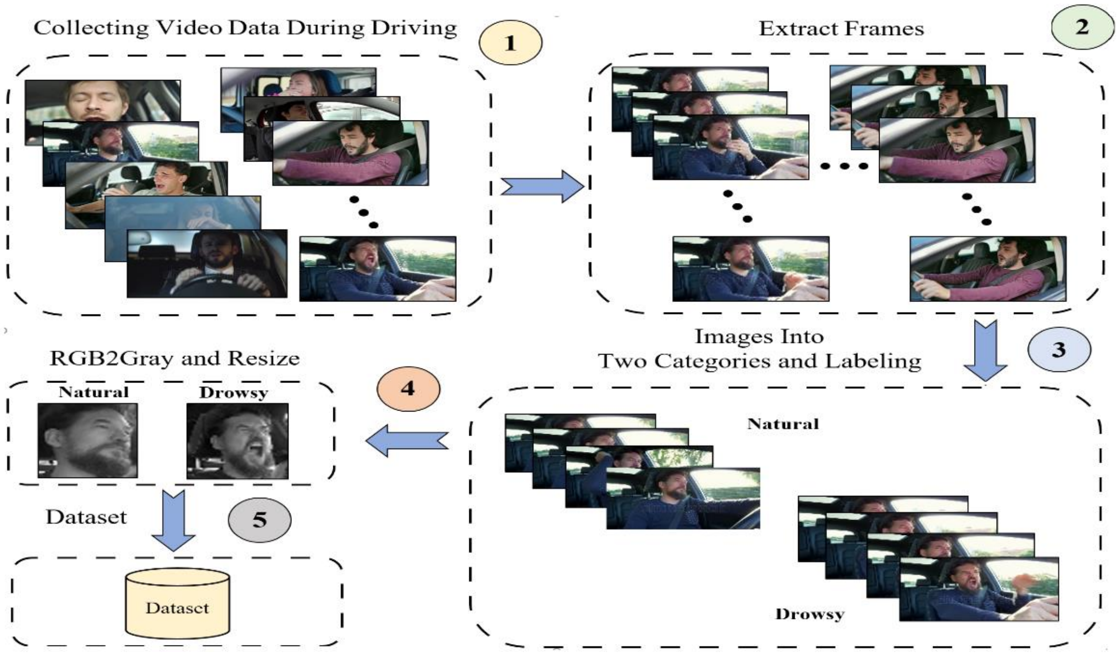


FIGURE 11. **Stages of creating the drowsiness dataset.**

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381999

C. Fine-Tuning a CNN for Driver Drowsiness Detection

As previously discussed in earlier sections, in this study, an optimal convolutional neural network (CNN) structure was initially developed using a genetic optimization algorithm based on the FER-2013 dataset. This model served as the initial foundation for emotion recognition. Subsequently, to adapt this network for driver drowsiness detection, a drowsiness dataset was utilized (the stages of creating the drowsiness dataset are explained in Section IV-B of the paper). This stage, known as "Fine-tuning," provides the capability to reconfigure the network using the knowledge acquired during training on the FER-2013 dataset. This approach leverages past experiences and utilizes the network's knowledge in emotion recognition to enhance the accuracy and performance of the CNN for driver drowsiness detection. Figure 12 illustrates the optimized neural network structure for driver drowsiness detection, which comprises six convolutional layers and one fully connected layer.

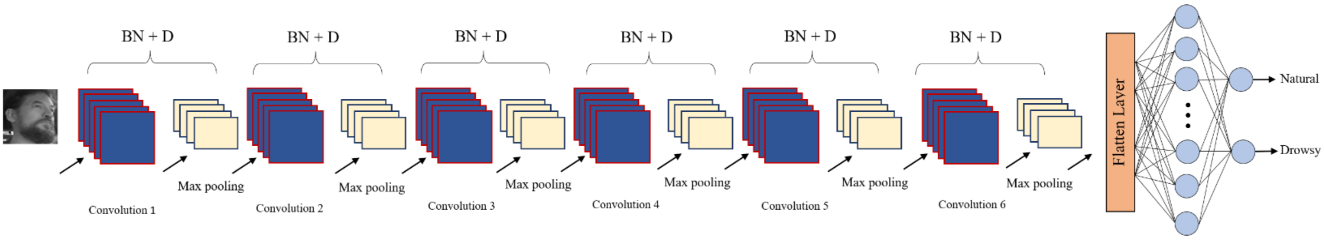


FIGURE 12. **Optimal CNN structure for driver drowsiness detection.**

According to Figure 13, in this study, a real-time driver drowsiness detection system based on Convolutional Neural Networks is presented. This system analyzes images captured from the driver to determine whether the individual is drowsy or not. The drowsiness detection system rapidly assesses driver images within a few seconds. Initially, the system employs image preprocessing to determine whether the driver is blinking. The images are sent to the operating system if a blinking state is detected. At this stage, the operating system decides whether the driver is at risk of drowsiness or not. If a drowsiness risk is identified, a warning signal is issued to the driver to help them prevent potential hazards. This drowsiness detection system, utilizing CNN and image processing, enhances safety and well-being in transportation scenarios. It assists drivers in staying attentive to their driving safety in critical situations such as drowsiness, thus improving transportation security and health.

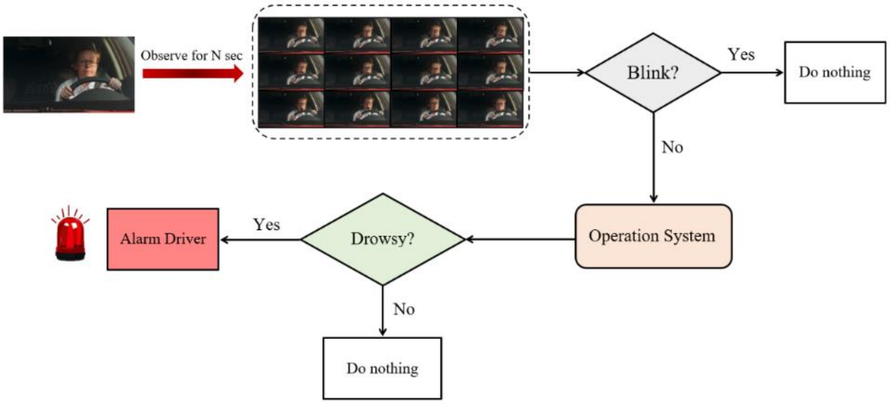


FIGURE 13. **Real-time driver drowsiness detection system based on CNN structure.**

V. Experimental Results

This section presents the results of design and implementation using the TensorFlow library on a computer with specifications of a core i7, 16G DDR5 RAM, and a 3070 graphics card. In this research, the results of the proposed method have been compared with various pre-trained networks such as VGG, ResNet, GoogleNet, and Mobilenet. Explanations about their structures are provided in Section 3. Additionally, the results have been compared with several convolutional neural networks and methods [4].

A. Parameter Setting

Table 1 provides a complete list of all the parameters required for simulating the results of this research. The values of parameters related to the genetic algorithm and other convolutional neural networks are reported in this table.

Table 1. **Parameter setting.**

Parameter setting	Value
Initial Population Size	50
Crossover Rate	0.52
Mutation Rate	0.24
Maximum Generation	250
Number of Epochs	[100, 150]
Convolution Layer Dropout Weight	[0.1, 0.5]
Number of Filters	{16, 32, 64}
Convolution Layer Activation Function	{LeakyReLU, ReLU, ELU}
Kernel Size	[(3,3), (5,5), (7,7)]
Dense Layer Dropout Weight	[0.1, 0.5]
Number of Neurons in a Dense Layer	{128, 256, 512}
Dense Layer Activation Function	{LeakyReLU, ReLU, ELU}
Epochs for Fine-tuning	1400
Learning Rate	0.001
Loss Function	Categorical Cross-entropy

B. **Evaluation Metrics**

The following metrics have been used to assess various models in analyzing the results of this research. These metrics are used to compare the best performance of each of the convolutional neural networks:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Negative\ predictive\ value (NPV) = \frac{TN}{TN + FN} \tag{5}$$

$$F1 - score = 2 * \frac{sensitivity * precision}{sensitivity + precision} \tag{6}$$

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381999

TP refers to the number of data points the model correctly predicted as belonging to the sleep disorder class. TN refers to the number of data points the model correctly predicted as belonging to the normal class. FP refers to the number of data points where their class has been incorrectly predicted as the sleep disorder class. FN refers to the number of data points where their class has been incorrectly predicted as the normal class.

Furthermore, for further analysis of the results and to eliminate the initial conditions of convolutional neural networks, including initial weights, the following metrics for the accuracy metric have been reported after ten independent runs of each method. These pieces of information show us how well the models perform under different conditions.

- **Best:** The best-obtained accuracy value from 10 independent runs of each model.
- **Worst:** The worst obtained accuracy value from 10 independent runs of each model.
- **Mean:** The average values of accuracy obtained from 10 independent runs of each of the models.
- **Variance (VAR):** This metric indicates the level of variability and deviations of results from the mean. A larger variance signifies more irregular results.
- **Standard Deviation (STD):** The value of this metric is calculated based on the square root of the variance and indicates the extent of data deviation from the mean. Smaller values represent result stability.
- **Confidence Interval (CI):** This metric shows a range around the mean value that is typically (usually 95%) certain to contain the actual value of the metric.

VI. Evaluation of the Proposed Method

In this section, the results of evaluating the performance of the proposed model in comparison to other convolutional neural networks in detecting driver drowsiness are reported. For this purpose, the proposed method has been compared to other common approaches, including GoogleNet, VGG, Mobilenet, Resnet, and several CNNs. The evaluation of the models' performance includes measuring detection accuracy, execution time, and other relevant performance metrics. The results of these comparisons provide us with valuable insights into the capabilities and efficiency of convolutional neural network models in detecting driver drowsiness. This research aims to enhance road safety and reduce drowsiness-related road accidents, contributing to improving public health and safety on the roads.

In this study, we have utilized a dataset comprising 7342 samples related to driver drowsiness, and the data generation process is described in Section 4-2. These data have been randomly divided into three sets: approximately 70% of the data (about 5125 samples) are used for the training phase, about 20% of the data (about 1483 samples) are allocated for testing the models, and finally, roughly 10% of the data (about 734 samples) are used for validation. Additionally, special attention has been given to ensuring that there is no overlap in the images of individuals between the training, validation, and testing datasets. This consideration allows for a more accurate evaluation of the models' performance and eliminates dependencies between images.

Table 2 shows the best-performing convolutional neural networks' results in detecting driver drowsiness, considering various evaluation metrics, including Accuracy, Sensitivity, Specificity, Precision, NPV, and F1-score. Based on the calculated metrics in Table 2, the comparison of different models demonstrates that there are significant differences in their performance. In terms of the accuracy metric, the results obtained from models such as GoogleNet, VGG, Mobilenet, Resnet, and other deep neural network architectures indicate that some models, such as ResNet with 2-dense layers and the proposed method, exhibit very high accuracy, with values of 99.59% and 99.79%, respectively.

Regarding the sensitivity metric (or True Positive Rate), the results of different models show how many of the positive samples (true cases of driver drowsiness) the models have correctly identified. In comparison to Accuracy, Sensitivity places greater emphasis on correctly detecting positive cases and attaches greater importance to preventing false negatives (misclassifying the absence of drowsiness). The sensitivity results indicate that some models, such as GoogleNet with 1-dense layer and the proposed method, have very high Sensitivity, with values of 99.58% and 100.00%, respectively. This information demonstrates that these models have a high capability in detecting positive cases of drowsiness, which can be highly beneficial in applications such as enhancing safety in driving.

Regarding the Specificity metric (or True Negative Rate), the evaluation results of the models show how many of the negative samples (true cases of the absence of driver drowsiness) the models have correctly identified. More precisely, specificity is a metric that reduces false positives (misclassifying the presence of drowsiness). The results indicate that some models, such as VGG with 2-dense layers, ResNet with 2-dense layers, and the Proposed Method, exhibit very high Specificity, with values of 99.31%, 99.45%, and 99.58%, respectively. These results demonstrate that these models have a high capability in detecting negative cases of the absence of drowsiness.

The Precision metric indicates how many positive cases (driver drowsiness) the different models have correctly identified. The results show that the VGG model with 2-dense layers (99.33%) and the ResNet model with 2-dense layers (99.47%) exhibit high precision in detecting drowsiness. This indicates their excellent ability to reduce false positives. Additionally, the proposed method also achieves very high precision (99.60%), demonstrating that this approach can contribute to improving the detection of driver drowsiness in driving scenarios.

The Negative Predictive Value (NPV) metric indicates how many of the negative cases (absence of driver drowsiness) the models have correctly identified. The results show that the proposed method achieves a very high NPV (100.00%), indicating its excellent ability to reduce false negative errors. Additionally, the VGG model with 2-dense layers and the ResNet model with 2-dense layers also exhibit very high NPV values (99.72% each), demonstrating that these models have effectively optimized their performance in detecting negative cases. These results are significant as, in applications such as enhancing road safety, reducing false negative errors holds high importance. Moreover, based on the F1-score metric, the proposed method ranks highest compared to other convolutional neural networks. The results in Table 2 indicate that the architecture and settings of the dense layers in the proposed model have played a significant role in detecting driver drowsiness.

Table 3 displays the values of True Negative, False Negative, True Positive, and False Positive for different methods. As observed, the proposed method exhibits the best results.

For further evaluation of the results, Table 4 reports the assessment of the accuracy metric for different architectures over ten independent runs. As seen in Table 4, the ResNet + 2-layers dense and proposed method occupy the top positions. In terms of the "worst" metric, the models ResNet + 1-layer dense, ResNet + 2-layers dense, and the proposed method also hold the highest positions.

Based on the average metric presented in Table 4, we can see that the proposed method, with an average of 98.91, exhibits the best performance and has achieved a higher average in detecting driver drowsiness. The models VGG + 2-layers dense and ResNet + 2-layers dense also rank second and third with averages of 98.56 and 98.55, respectively. In contrast, the CNN 4-layers + 1-layer dense model has a lower performance with an average of 83.82. In terms of standard deviation and variance, the MobileNet + 1-layer dense method holds the highest rank.

According to Table 4, the Confidence Interval metric provides important information about the stability of model performance in detecting driver drowsiness. This metric indicates how likely the performance values of models will vary around their mean. Models with lower standard deviation exhibit better stability in their performance. Based on the presented results, the Proposed Method and VGG + 2-layer dense models have lower standard deviations and demonstrate relatively stable performance in detecting driver drowsiness. In contrast, the CNN 4-layers + 1-layer dense model has a higher standard deviation, indicating less stability in its performance. This information helps in making decisions related to the use of different models in real-world applications and emphasizes the importance of stable models in pattern recognition and more accurate predictions.

In evaluating machine learning models' performance, the time required for training each model plays a significant role. The training time for models is associated with their performance and efficiency, and typically, more complex models with a larger number of parameters require more time for training. Based on the presented results, the VGG + 1-layer dense and VGG + 2-layer dense models require a longer training time than other models. On the other hand, the MobileNet + 1-layer dense and MobileNet + 2-layers dense models require less training time than the others. The training time for the proposed method model is 49,500 seconds. This time appears to be relatively moderate and indicates that the proposed method strikes a balance between training speed and desirable performance in detecting driver drowsiness.

Table 5 displays the optimal convolutional neural network architecture obtained by the proposed method for driver drowsiness detection.

Table 2. **Evaluation results of different architectures.**

Method	Accuracy	Sensitivity	Specificity	Precision	Negative Predictive Value	F1 score	**Total parameters**
GoogleNet +1-layer dense	97.50	99.58	95.50	95.50	99.58	97.49	48,018,722
GoogleNet +2-layers dense	95.34	97.86	92.16	91.94	98.89	94.80	74,758,434
VGG +1-layer dense	98.85	98.81	98.89	94.94	98.76	96.83	27,561,282
VGG + 2-layers dense	99.52	99.73	99.31	99.33	99.72	99.52	40,931,650
MobileNet + 1-layer dense	97.16	97.98	96.34	96.43	97.93	97.19	34,379,162
MobileNet + 2-layers dense	94.24	97.9	98.61	98.67	97.79	98.28	67,010,114
ResNet + 1-layer dense	99.39	99.47	99.31	99.33	99.44	99.39	74,969,474
ResNet + 2-layer dense	99.59	99.73	99.45	99.47	99.72	99.59	126,875,010
CNN 4-layers +1-layer dense	86.31	86.25	85.53	87.05	85.53	86.64	31,803,522
CNN 7-layers +1-layer dense	88.61	85.26	93.05	94.05	83.06	89.43	27,754,242
Proposed method	**99.80**	**100.00**	**99.58**	**99.60**	**100.00**	**99.79**	29,088,354

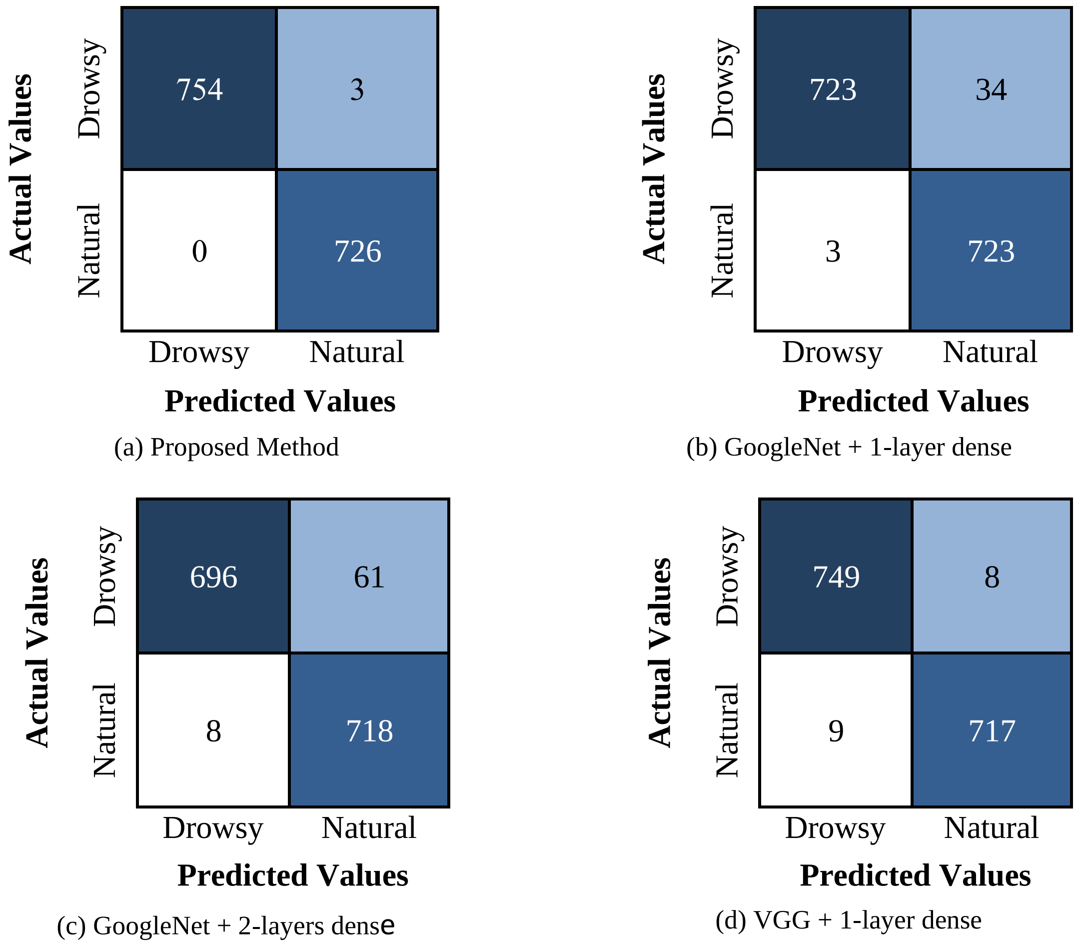
Table 3. **True Negative, False Negative, True Positive, and False Positive values for different structures.**

Metric	GoogleNet + 1-layer dense	GoogleNet + 2-layers dense	VGG + 1-layer dense	VGG + 2-layers dense	MobileNet+ 1 layer dense	MobileNet + 2-layers dense	ResNet + 1-layer dense	ResNet + 2-layers dense	CNN 4 layers +1-layer dense	CNN 7-layers +1-layer dense	Proposed method
FP (False Positive)	34	61	8	5	27	10	5	4	98	45	**3**
TN (True Negative)	723	718	717	724	711	710	722	724	621	603	**726**
FN (False Negative)	3	8	9	2	15	16	4	2	105	123	**0**
TP (True Positive)	723	696	749	752	730	747	752	753	659	712	**754**

Table 4. **Evaluation of Accuracy metric for different architectures in 10 independent runs.**

Method	Best	Worst	Mean	VAR	STD	CI	Runtime
VGG +1-layer dense	98.85	96.41	97.25	0.61	0.78	97.25 ± 0.48	66000
VGG +2-layers dense	99.52	95.67	98.56	1.00	1.00	98.56 ± 0.62	66500
MobileNet + 1-layer dense	97.16	96.42	96.83	**0.09**	**0.29**	96.83 ± 0.18	**38100**
MobileNet + 2-layers dense	94.24	89.88	92.99	3.06	1.75	92.99 ± 1.08	38756
ResNet + 1-layer dense	99.39	97.10	97.91	0.39	0.63	97.91 ± 0.39	82380
ResNet + 2-layers dense	99.59	**97.90**	98.55	0.37	0.61	98.55 ± 0.38	82500
GoogleNet + 1-layer dense	97.50	92.10	95.14	3.89	1.97	95.14 ± 1.22	55200
GoogleNet + 2-layers dense	95.34	90.20	93.92	4.38	2.09	93.92 ± 1.30	55680
CNN 4 layers +1-layer dense	86.31	79.00	83.82	5.34	2.31	83.82 ± 1.43	43500
CNN 7 layers +1-layer dense	88.61	84.25	86.08	3.54	1.88	86.08 ± 1.16	45750
Proposed method	**99.80**	96.50	**98.91**	0.90	0.95	**98.91 ± 0.59**	49500

The confusion matrix for different models for drowsiness detection based on the accuracy metric is shown in Figure 14. As seen in Figure 14, the proposed model outperforms other models in terms of accuracy. This means that the proposed model has better capabilities in correctly identifying both positive and negative samples. Additionally, the VGG + 2-dense layer model performs particularly worse in terms of False Positives (FP) compared to the proposed model. Overall, the proposed approach has demonstrated better performance than other models and can be a suitable candidate for addressing the problem of drowsiness detection in drivers during driving.



(a) Proposed Method

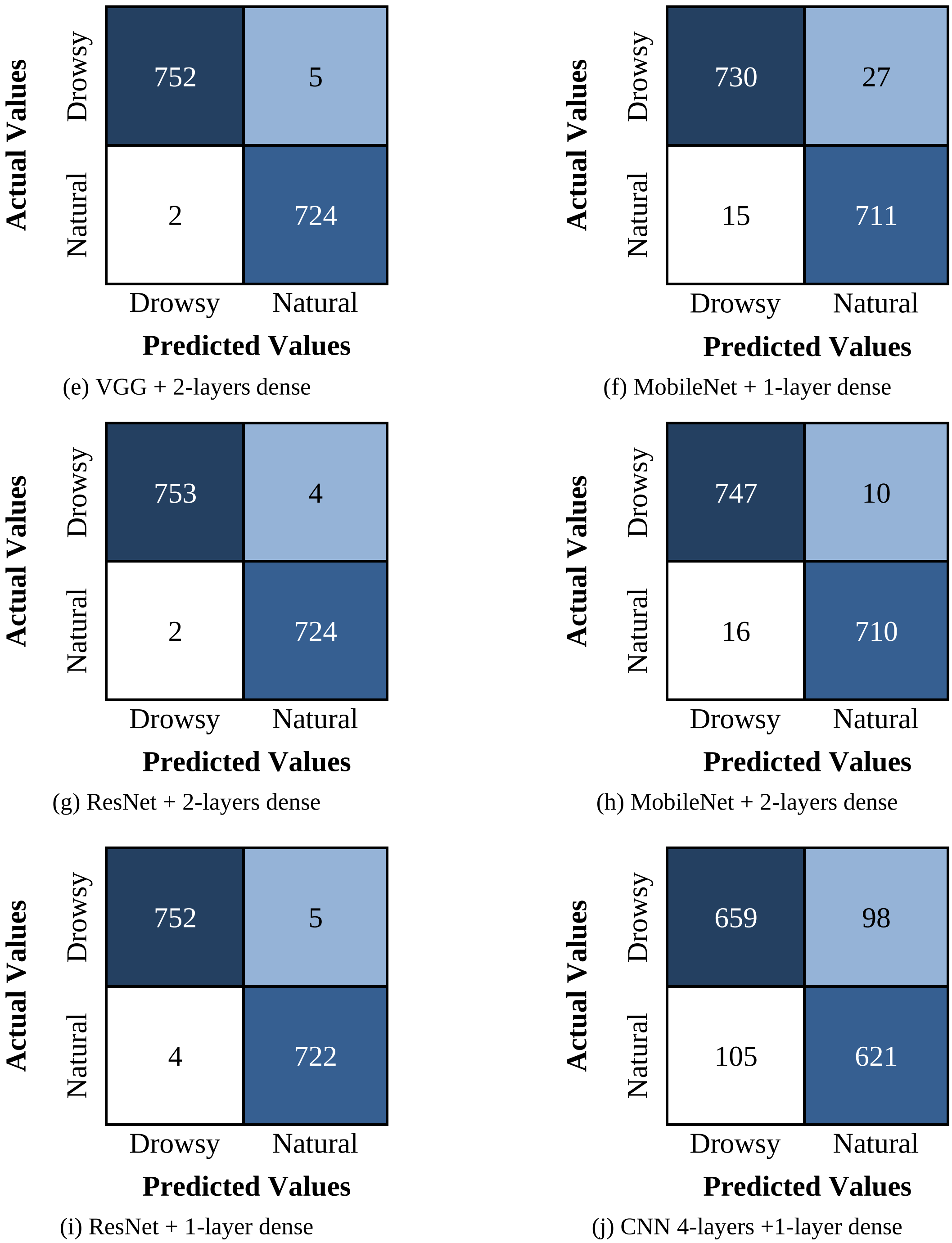
(b) GoogleNet + 1-layer dense

(c) GoogleNet + 2-layers dense

(d) VGG + 1-layer dense

Figure 14. **Evaluation of accuracy metric for different architectures in 10 independent runs.**

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381999



(e) VGG + 2-layers dense

(f) MobileNet + 1-layer dense

(g) ResNet + 2-layers dense

(h) MobileNet + 2-layers dense

(i) ResNet + 1-layer dense

(j) CNN 4-layers +1-layer dense

Figure 14. **Continue.**

Table 5 **The structure obtained from the proposed method for driver drowsiness detection.**

#	Layer Name	Type	Shape	Total Parameters	Dropout Value	Active Function	Kernel Size	Feature Map
1	conv1	Conv2D	(None, 48, 48, 32)	896		LeakyReLU	(3, 3)	32
	dropout_1	Dropout	(None, 48, 48, 32)	0	0.1			
	batch_normalization_1	Batch Normalization	(None, 48, 48, 32)	128				
	max_pooling2d_1	MaxPooling2D	(None, 47, 47, 32)	0				
2	conv2	Conv2D	(None, 47, 47, 64)	51264		LeakyReLU	(5, 5)	64
	dropout_2	Dropout	(None, 47, 47, 64)	0	0.4			
	batch_normalization_2	Batch Normalization	(None, 47, 47, 64)	256				
	max_pooling2d_2	MaxPooling2D	(None, 46, 46, 64)	0				
3	conv3	Conv2D	(None, 46, 46, 32)	18464		ReLU	(3, 3)	32
	dropout_3	Dropout	(None, 46, 46, 32)	0	0.1			
	batch_normalization_3	Batch Normalization	(None, 46, 46, 32)	128				
	max_pooling2d_3	MaxPooling2D	(None, 45, 45, 32)	0				
4	conv4	Conv2D	(None, 45, 45, 32)	9248		LeakyReLU	(3, 3)	32
	dropout_4	Dropout	(None, 45, 45, 32)	0	0.3			
	batch_normalization_4	Batch Normalization	(None, 45, 45, 32)	128				
	max_pooling2d_4	MaxPooling2D	(None, 44, 44, 32)	0				
5	conv5	Conv2D	(None, 44, 44, 64)	51264		LeakyReLU	(5, 5)	64
	dropout_5	Dropout	(None, 44, 44, 64)	0	0.3			
	batch_normalization_5	Batch Normalization	(None, 44, 44, 64)	256				
	max_pooling2d_5	MaxPooling2D	(None, 43, 43, 64)	0				
6	conv6	Conv2D	(None, 43, 43, 32)	51232		ELU	(5, 5)	32
	dropout_6	Dropout	(None, 43, 43, 32)	0	0.3			
	batch_normalization_6	Batch Normalization	(None, 43, 43, 32)	128				
	max_pooling2d_6	MaxPooling2D	(None, 42, 42, 32)	0				
7	flatten	Flatten	(None, 56448)	0				
	Fc1	Dense	(None, 512)	28901888		LeakyReLU		
	dropout_7	Dropout	(None, 512)	0	0.1			
8	batch_normalization_8	Batch Normalization	(None, 512)	2048				
	Fc2	Dense	(None, 2)	1026		SoftMax		

VII. Conclusion

This article employs advanced approaches to detect fatigue and drowsiness in drivers during driving. The main solution proposed in this research is the use of a genetic algorithm in combination with a neural network architecture search to find an optimal structure for convolutional neural network models. This approach can design and enhance a model with high accuracy and good detection capabilities. One of the strengths of this study is the utilization of a large and diverse dataset, including more than 7342 images of drivers during driving. This dataset allows training the network under more realistic conditions and improving model performance. Transfer learning has also been utilized to transfer valuable information obtained from the source dataset (FER-2013) to the proposed model's optimal architecture. The results obtained from designing and implementing various models show that the proposed method outperforms other models in terms of different metrics such as Accuracy, Sensitivity, Specificity, Precision, NPV, and F1-score. The main conclusion of this research is achieving a 99.8% accuracy in detecting driver drowsiness. This result demonstrates that the combined approach presented can be an effective tool for enhancing safety in driving environments and preventing road accidents. Furthermore, this research has faced challenges, such as the need for powerful hardware. Suggestions for future work include designing full-parameter models using genetic algorithms and using newer optimization algorithms to achieve even better accuracy. Overall, this research demonstrates that the combination of network architecture search, transfer learning, and genetic algorithms can help improve the performance of fatigue and drowsiness detection models and be valuable for enhancing safety in driving environments.

References

[1] Jang, S.W. and Ahn, B., 2020. Implementation of detection system for drowsy driving prevention using image recognition and IoT. Sustainability, 12(7), p.3037.

[2] Tefft, B.C., Arnold, L.S. and Grabowski, J.G., 2016. AAA foundation for traffic safety. Washington, DC.

[3] Council, N.S., 2020. Drivers are falling asleep behind the wheel. National Safety Council.

[4] Poli, A., Amidei, A., Benatti, S., Iadarola, G., Tramarin, F., Rovati, L., Pavan, P. and Spinsante, S., 2022, November. Exploiting Blood Volume Pulse and Skin Conductance for Driver Drowsiness Detection. In EAI International Conference on IoT Technologies for HealthCare (pp. 50-61). Cham: Springer Nature Switzerland.

[5] Maior, C.B.S., das Chagas Moura, M.J., Santana, J.M.M. and Lins, I.D., 2020. Real-time classification for autonomous drowsiness detection using eye aspect ratio. Expert Systems with Applications, 158, p.113505.

[6] Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M. and Jiang, S., 2018. Real-time driver drowsiness detection for Android application using deep neural networks techniques. Procedia computer science, 130, pp.400-407.

[7] Azim, T., Jaffar, M.A. and Mirza, A.M., 2014. Fully automated real-time fatigue detection of drivers through fuzzy expert systems. Applied Soft Computing, 18, pp.25-38.

[8] Knapik, M. and Cyganek, B., 2019. Driver's fatigue recognition based on yawn detection in thermal images. Neurocomputing, 338, pp.274-292.

[9] Bylykbashi, K., Qafzezi, E., Ikeda, M., Matsuo, K. and Barolli, L., 2020. Fuzzy-based Driver Monitoring System (FDMS): Implementation of two intelligent FDMSs and a testbed for safe driving in VANETs. Future Generation Computer Systems, 105, pp.665-674.

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

[11] Lee, C. and An, J., 2023. LSTM-CNN model of drowsiness detection from multiple consciousness states acquired by EEG. Expert Systems with Applications, 213, p.119032.

[12] Amidei, A., Poli, A., Iadarola, G., Tramarin, F., Pavan, P., Spinsante, S. and Rovati, L., 2022, July. Driver drowsiness detection based on variation of skin conductance from wearable device. In 2022 IEEE International Workshop on Metrology for Automotive (MetroAutomotive) (pp. 94-98). IEEE.

[13] Patel, P.P., Pavesha, C.L., Sabat, S.S. and More, S.S., 2022, May. Deep Learning based Driver Drowsiness Detection. In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 380-386). IEEE.

[14] Al Redhaei, A., Albadawi, Y., Mohamed, S. and Alnoman, A., 2022, February. Realtime Driver Drowsiness Detection Using Machine Learning. In 2022 Advances in Science and Engineering Technology International Conferences (ASET) (pp. 1-6). IEEE.

[15] Pandey, R., Bhasin, P., Popli, S., Sharma, M. and Sharma, N., 2022. Driver drowsiness detection and traffic sign recognition system. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 1 (pp. 25-40). Singapore: Springer Nature Singapore.

[16] William, P., Shamim, M., Yeruva, A.R., Gangodkar, D., Vashisht, S. and Choudhury, A., 2022, October. Deep Learning based Drowsiness Detection and Monitoring using Behavioural Approach. In 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS) (pp. 592-599). IEEE.

[17] Tüfekci, G., Kayabaşı, A., Akagündüz, E. and Ulusoy, İ., 2022, October. Detecting Driver Drowsiness as an Anomaly Using LSTM Autoencoders. In European Conference on Computer Vision (pp. 549-559). Cham: Springer Nature Switzerland.

[18] Dwivedi, S.A., Attry, A. and Singla, K., 2022. Leveraging Transfer Learning for Driver Drowsiness Detection. In Advances in Data and Information Sciences: Proceedings of ICDIS 2021 (pp. 603-611). Singapore: Springer Singapore.

[19] Yogesh, R., Ritheesh, V., Reddy, S. and Rajan, R.G., 2022, July. Driver Drowsiness Detection and Alert System using YOLO. In 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-6). IEEE.

[20] Jumana, R. and Jacob, C., 2022, December. Deep CNN Based Approach for Driver Drowsiness Detection. In 2022 IEEE International Power and Renewable Energy Conference (IPRECON) (pp. 1-6). IEEE.

[21] Phan, A.C., Trieu, T.N. and Phan, T.C., 2023. Driver drowsiness detection and smart alerting using deep learning and IoT. Internet of Things, 22, p.100705.

[22] Shaik, M.E., 2023. A systematic review on detection and prediction of driver drowsiness. Transportation Research Interdisciplinary Perspectives, 21, p.100864.

[23] Fouad, I.A., 2023. A robust and efficient EEG-based drowsiness detection system using different machine learning algorithms. Ain Shams Engineering Journal, 14(3), p.101895.

[24] Bekhouche, S.E., Ruichek, Y. and Dornaika, F., 2022. Driver drowsiness detection in video sequences using hybrid selection of deep features. Knowledge-Based Systems, 252, p.109436.

[25] Walizad, M.E., Hurroo, M. and Sethia, D., 2022, April. Driver drowsiness detection system using convolutional neural network. In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1073-1080). IEEE.

[26] Jahan, I., Uddin, K.A., Murad, S.A., Miah, M.S.U., Khan, T.Z., Masud, M., Aljahdali, S. and Bairagi, A.K., 2023. 4D: a real-time driver drowsiness detector using deep learning. Electronics, 12(1), p.235.

[27] Florez, R., Palomino-Quispe, F., Coaquira-Castillo, R.J., Herrera-Levano, J.C., Paixão, T. and Alvarez, A.B., 2023. A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification. Applied Sciences, 13(13), p.7849.

[28] Albadawi, Y., AlRedhaei, A. and Takruri, M., 2023. Real-time machine learning-based driver drowsiness detection using visual features. Journal of imaging, 9(5), p.91.

[29] Suresh, A., Naik, A.S., Pramod, A., Kumar, N.A. and Mayadevi, N., 2023, May. Analysis and Implementation of Deep Convolutional Neural Network Models for Intelligent Driver Drowsiness Detection System. In 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 553-559). IEEE.

[30] Saini, P., Kumar, K., Kashid, S., Negi, A. and Saini, A., 2023. Driver Drowsiness Detection for Road Safety Using Deep Learning. In Robotics, Control and Computer Vision: Select Proceedings of ICRCCV 2022 (pp. 197-205). Singapore: Springer Nature Singapore.

[31] Wang, et Al (2022). A Driving Fatigue Feature Detection Method Based on Multifractal Theory. IEEE Sensors Journal, 22(19), 19046-19059. doi: 10.1109/JSEN.2022.3201015

[32] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas and A. Mahmood, "A Survey on State-of-the-Art Drowsiness Detection Techniques," in *IEEE Access*, vol. 7, pp. 61904-61919, 2019, doi: 10.1109/ACCESS.2019.2914373.

[33] Jiawei Xu, Sicheng Pan, Poly Z.H. Sun, Seop Hyeong Park, and Kun Guo, “Human-Factors-in-Driving-Loop: Driver Identification and Verification via a Deep Learning Approach using Psychological Behavioral Data,” IEEE Transactions on Intelligent Transportation Systems(IEEE-TITS),2022.DOI: 10.1109/TITS.2022.3225782

[34] Xu, J., et al (2022). Driving Performance Under Violations of Traffic Rules: Novice Vs. Experienced Drivers. IEEE Transactions on Intelligent Vehicles. doi: 10.1109/TIV.2022.3200592

[35] Yetong Cao, Fan Li, Xiaochen Liu, Song Yang, and Yu Wang. 2023. Towards Reliable Driver Drowsiness Detection Leveraging Wearables. ACM Trans. Sen. Netw. 19, 2, Article 39 (May 2023), 23 pages. https://doi.org/10.1145/3560821

[36] Min, J., Cai, M., Gou, C. et al. Fusion of forehead EEG with machine vision for real-time fatigue detection in an automatic processing pipeline. Neural Comput & Applic 35, 8859–8872 (2023). https://doi.org/10.1007/s00521-022-07466-0

[37] Chandana, R., Sangeetha, J. (2023). Drowsiness Detection for Automotive Drivers in Real-Time. In: Reddy, K.A., Devi, B.R., George, B., Raju, K.S., Sellathurai, M. (eds) Proceedings of Fourth International Conference on Computer and Communication Technologies. Lecture Notes in Networks and Systems, vol 606. Springer, Singapore. https://doi.org/10.1007/978-981-19-8563-8_17

[38] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[39] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[40] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[41] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

YASHAR JEBRAEILY was born in Tehran, Iran, in 1996. He received the M.S. degree in electrical engineering from K. N. Toosi University of Technology, in 2023, He is also a member of the Intelligent Systems Laboratory. His research interests include artificial intelligence, machine learning, deep learning, intelligent control, and Evolutionary Optimization Algorithms.

MOHAMMAD TESHNEHLAB was born in Borujerd, Iran, in 1957. He received the B.Sc. degree in electrical engineering from Stony Brook University, Stony Brook, NY, USA, in 1981, the M.Sc. degree in electrical engineering from Oita University, Japan, in 1991, and the Ph.D. degree from Saga University, Japan, in 1993. He is currently a Faculty Member with the Department of Electrical Engineering, K. N. Toosi University of Technology. He is also a member of the Industrial Control Center of Excellence and the Founder of the Intelligent Systems Laboratory (ISLab). His main research interests include intelligent systems and control, including artificial rough and deep neural networks, fuzzy systems, neural nets, optimization, and applications in the identification, prediction, classification, and control. He is the Head and the Co-Founder of the Intelligent Systems Scientific Society of Iran (ISSSI), and a member of the International Journal of Information and Communication Technology Research (IJICTR) Editorial Board.

YOUSEF SHARAFI received the M.S. and Ph.D. degree in artificial intelligence from Science and Research Branch, Islamic Azad University, Tehran, Iran. His primary research interests encompass a broad spectrum of Artificial Intelligence, including Machine Learning Algorithms, Deep Learning Models, Fuzzy Systems, and Evolutionary Optimization Algorithms. With a wealth of experience in machine learning algorithms, he has made significant contributions to interdisciplinary projects spanning fields such as medicine, mechanical engineering, electrical engineering, psychology, agriculture, and environmental engineering.