REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

FATIGUE DETECTION OF DRIVERS USING IMAGE PROCESSING AND ARTIFICIAL INTELLIGENCE TECHNIQUES

Omer Nasih Ismael ALHURMUZI

Master`s Thesis

Supervisor

Asst. Prof. Dr. Timur İNAN

Istanbul, 2023

FATIGUE DETECTION OF DRIVERS USING IMAGE PROCESSING AND ARTIFICIAL INTELLIGENCE TECHNIQUES

Omer Nasih Ismael ALHURMUZI

Information Technologies

Master`s Thesis

ALTINBAS UNIVERSITY

2023

The thesis titled FATIGUE DETECTION OF DRIVERS USING IMAGE PROCESSING AND ARTIFICIAL INTELLIGENCE TECHNIQUES prepared by OMER NASIH ISMAEL ALHURMUZI and submitted on 09/08/2023 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr. Timur İNAN

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Timur İNAN	Department of Software Engineering, Altınbaş University	____________________
Asst. Prof. Dr. Mesut ÇEVİK	Department of Electrical and Electronics Engineering, Altınbaş University	____________________
Asst. Prof. Dr. Vedat ESEN	Department of Electrical and Electronics Engineering, İstanbul Topkapi University	____________________

I hereby declare that this thesis meets all format and submission requirements of a Master`s Thesis.

Submission date of the thesis to the Institute of Graduate Studies: ____/____/____

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Omer Nasih Ismael ALHURMUZI

Signature

DEDICATION

To my beloved father, whose constant support and advice shaped the man I am today. Your courage, intelligence, and selflessness have been an ongoing source of inspiration in my life. Thank you for fostering tenacity and compassion in me, and for always believing in my dreams.

To my beloved mother, whose love and caring have been a beacon of light for me on my journey. Your unending love, soothing words, and unfailing faith in me have given me the confidence to confront any difficulty. Your efforts and unending encouragement have been the foundation of my success, and I will be forever thankful.

To my wonderful wife, the foundation of my existence. My greatest blessings have been your unshakable faith in me and your unconditional affection. Your constant support, patience, and understanding have seen me through everything that life has to offer. You are my constant source of strength, my companion in all endeavours, and the one who brightens my day.

To my lovely brother and sisters, who have been my constant companions throughout my life. Your laughter, friendship, and shared memories have brought joy and warmth into my life. We've overcome obstacles, celebrated victories, and formed an unshakable friendship. Your kindness and support have given me a sense of purpose and belonging.

This dedication is a salute to my father, mother, wife, brother, and sisters for their unfailing love, support, and inspiration. Each of you has played an important role in moulding my character and giving me the strength to tackle challenges. Your presence in my life has filled it with boundless love, joy, and meaning. I will be eternally grateful for the family I am privileged to name my own.

ACKNOWLEDGEMENTS

I'd like to show my deepest thanks and admiration to Dr. Timur Inan, whose excellent advice and skills were critical to the successful completion of this project. Dr. Inan's consistent support, mentorship, and commitment to quality have been tremendously motivating throughout this journey.

Dr. Inan's extensive knowledge in the subject, combined with his constant drive to pushing boundaries and discovering new horizons, has greatly influenced my thinking and attitude. His enthusiasm for study and ability to communicate knowledge in a clear and straightforward manner have been significant advantages in my learning and development.

I am eternally grateful for Dr. Inan's patience, as he kindly spent his time and resources to providing me with advice and comments. His astute observations, constructive criticism, and encouragement moved me ahead and inspired me to strive for excellence.

Finally, this project would not have been possible without the support and advice of Dr. Timur Inan, whose experience and mentorship have been priceless. I am deeply appreciative for the knowledge and skills that Dr. Inan has imparted to me during this journey, and I look forward to continuing to learn and contribute to the field under his direction.

ABSTRACT

FATIGUE DETECTION OF DRIVERS USING IMAGE PROCESSING AND ARTIFICIAL INTELLIGENCE TECHNIQUES

ALHURMUZI, Omer Nasih Ismael

M.Sc., Information Technologies, Altınbaş University,

Supervisor Asst. Prof. Dr. Timur İNAN

Date: August / 2023

Pages: 44

Road traffic accidents pose a serious threat to public safety and are a major source of fatalities around the world, particularly those brought on by driver drowsiness. With the development of artificial intelligence (AI), efforts have been made to create innovative projects and systems targeted at identifying driver weariness and issuing early warnings. In this study, we describe a real-time driver detection and warning system that makes use of deep artificial intelligence, including convolutional neural networks (CNNs), open CV face recognition, and image processing methods. CNNs can recognize small changes in a driver's behaviour or appearance that suggest fatigue because they perfectly replicate the human visual system.

These networks are highly accurate in identifying patterns and signs of fatigue because they have been trained on huge datasets of driving behaviour. The CNNs are integrated into a system that continuously analyses the behaviour of the driver using a variety of sensors, including cameras and biometric sensors, to identify indicators of fatigue, such as drooping eyelids, and notify the driver through tactile, visual, or audio messages. By creating a cutting-edge AI system that accurately identifies driver fatigue and swiftly alerts other drivers, the research potentially prevents accidents and saves lives on the road.

Keywords: Road Traffic Accidents, Public Safety, Driver Drowsiness, Artificial Intelligence (Ai), Driver Detection, Convolutional Neural Networks (CNN’s), Saving Lives.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF CHARTS

<u>Pages</u>

ABBREVIATIONS

ML	:	Machine Learning
DL	:	Deep Learning
CNN	:	Convolution Neural Network
DCNN	:	Deep Convolution Neural Network
OACE	:	Open and Close Eyes
MAE	:	Mean Absolute Error
RMSE	:	Root Mean Squared Error

1. INTRODUCTION

1.1 OVERVIEW

In 2016, 1.35 million individuals died from traffic-related injuries annually [1].Road traffic accidents are the primary cause of death for people between the ages of 15 and 29 around the world. Road traffic accidents are now thought to be the eighth biggest cause of mortality worldwide across all age groups, and by 2030 they are expected to overtake natural causes as the seventh leading cause of death [2]. Among these car accidents, the percentage of accidents that occur due to fatigue is estimated at 2.2%[3] compared to other causes. Which may lead to serious losses, including death, injury, or financial losses.

Figure 1.1: Car Accident.

To reduce these incidents and because of the development of artificial intelligence and its entry into various aspects of our lives to facilitate and relieve human beings of such accidents and disasters, the researchers worked on developing a set of programs and projects [4]–[12]to warn drivers in the event of fatigue, including the system of identifying fatigue and early warning. They used a set of variables that they adopted. It varies from one system to another.

Figure 1.2: Risks of Sleeping or Fatigue While Driving.

In this research, we have chosen a driver early warning system that relies on a deep artificial intelligence system for detecting driver fatigue and alerting drivers in real-time. Specifically, Convolution Neural Network (CNNs), an open CV face recognition system, and an image processing system to improve and adjust the accuracy of the images captured for the driver from the live camera in real time.

1.2 INSPIRATION

On our highways, driver fatigue is a severe issue that can cause accidents and injuries. Yet with the advancement of deep artificial intelligence, we can now construct smart systems that can recognize signs of exhaustion and warn drivers in real-time.

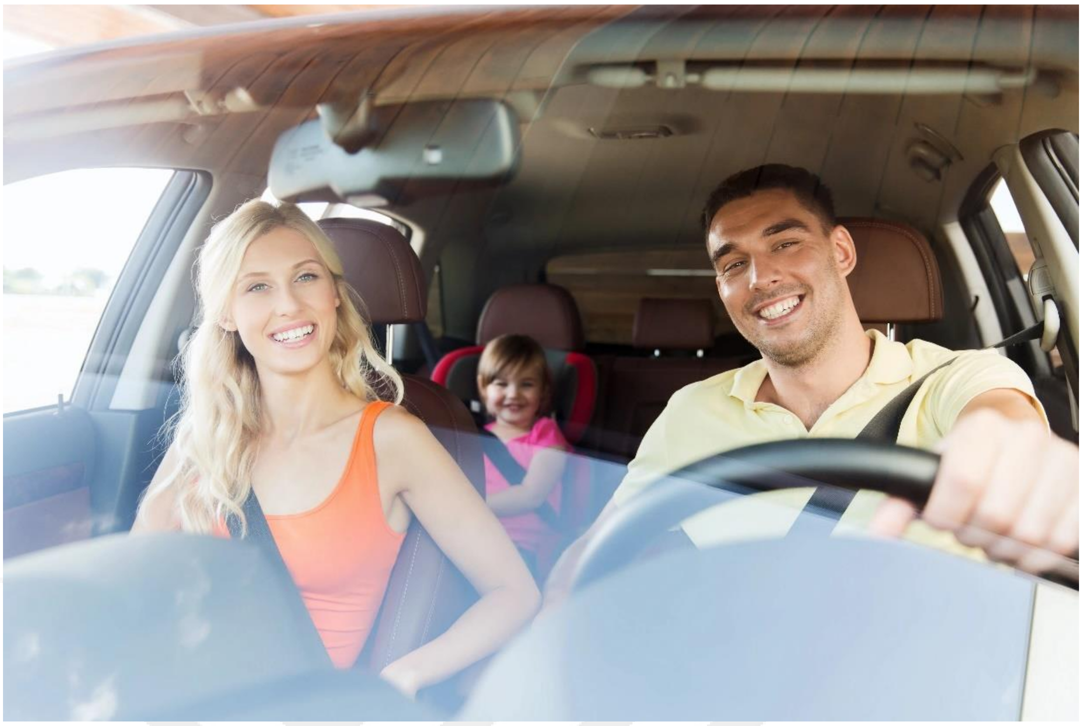


Figure 1.3: Driving Safely with The Family.

One of the most promising methods for detecting driver weariness with artificial intelligence is Deep Convolutional Neural Networks (DCNNs). These neural networks are created to scan photos and video data in a way that replicates the human visual system, enabling them to identify minute adjustments in a driver's behaviour or appearance that could signify fatigue.

One of the main benefits of utilizing CNNs for this purpose is that they can be trained on substantial datasets of driving behaviour, which enables them to develop very accurate pattern recognition and tiredness detection abilities. Many methods, including supervised learning, unsupervised learning, or a combination of the two, can be used to carry out this training.

Once trained, the CNNs can be included into a system that continuously observes a driver's behaviour. Many sensors, including cameras, microphones, and even biometric sensors, can be used by this system to gather information on the driver's actions and mental state.

If the CNNs notices indicators of exhaustion, such as drooping eyelids, yawning, or other actions that could suggest drowsiness, it can then analyse this data and notify the driver. These signals may be physical, such as vibrations or changes in the steering or braking of the vehicle, or they may be visual, audible, or both.

We can develop intelligent systems that are precise and unobtrusive to detect indicators of exhaustion using CNNs, allowing drivers to remain alert and focused on the road without being interrupted by pointless notifications.

In conclusion, the application of CNNs and artificial intelligence constitutes a significant advancement around driver safety. We can develop intelligent systems that can recognize indicators of weariness and notify drivers in real-time, potentially saving lives on our roads, by leveraging the power of cutting-edge algorithms and machine learning approaches.

1.3 RESEARCH CONTRIBUTION

My contribution to the research focuses on the creation of a novel deep artificial intelligence system for identifying driver weariness and immediately warning other drivers. Convolutional Neural Networks (CNNs) are specifically used in my research to evaluate video footage and identify indicators of exhaustion, such as drooping eyelids, yawning, or other actions that may signify drowsiness.

Using a mix of supervised and unsupervised learning strategies, I initially trained the CNNs on a sizable dataset of driving behaviour to create this system. This made it possible for the CNNs to develop very accurate pattern recognition and fatigue detection abilities.

When the CNNs had been trained, I incorporated it into a system that used cameras to continuously observe a driver's behaviour. With the use of vibrations, changes in the steering or braking of the vehicle, or visual, audio, or physical signals, the system can identify indicators of driver weariness.

I ran a few experiments with a driving simulator and a dataset of driver behaviour to gauge the system's efficacy. The outcomes of these tests showed that the system had a high degree of accuracy in identifying indicators of weariness and that it could warn drivers in real-time, possibly averting collisions and saving lives on the roads.

1.4 RESEARCH AIM

Ultimately, by creating a new artificial intelligence system that can recognize indicators of weariness and warn drivers in real-time, my research significantly advances the subject of driver safety. This method has the potential to greatly reduce the number of accidents brought on by fatigued driving and increase overall driver safety.

1.5 THESIS OUTLINES

The paragraph that follows is how the thesis is structured:

Chapter One: It presents an overview, inspiration, and the research aim.

Chapter Two: It presents a literature review of some similar work and explains some important branches like Artificial Intelligence (AI), and DL based on the Convolutional Neural Networks (CNN) method are used in the proposed machine learning (ML) system.

Chapter Three: Highlights the Methodology, the algorithms and the dataset adopted for model training.

Chapter Four: This chapter explains the results gained from implementing the proposed system.

Chapter five: This chapter presents the conclusions that emerged from the application of the system, as well as recommendations for future work of the proposed system.

2. LITERATURE REVIEW

Driver fatigue detection and alarm systems have been the subject of more research recently in the fields of computer vision and artificial intelligence.
The creation of these technologies is essential to minimizing accidents brought on by drive r drowsiness, one of the main factors contributing to traffic accidents globally.
A review of the existing research is necessary to create effective and efficient driver fatigu e detection systems.

This review of the literature concentrates on four research that investigate the use of real-time driver warning systems, fuzzy logic, and convolutional neural networks (CNNs) to detect driver fatigue. The main goals of this review are to compare and synthesize the findings of these studies, to uncover overlaps and gaps in the literature, and to emphasize the most significant contributions made by each study.

This review of the literature focuses on four studies[5], [6], [9], [10]that look at the application of convolutional neural networks (CNNs), fuzzy logic, and real-time driver warning systems to identify driver weariness. The main objectives of this review are to compare and integrate the results of various investigations, to identify gaps and overlapping areas in the literature, and to highlight the most important contributions made by each study. Since the development of these systems is key to reducing the number of accidents caused by driver fatigue, reviewing the literature is important in the context of driver fatigue detection. A review of this literature will help determine the most effective techniques and procedures for developing such systems. The development of driver fatigue detection systems has been considerably facilitated by the volume of current literature on the issue.

It is crucial to carry out this literature review to assess the current state of research in driver tiredness detection and alert systems and to provide guidance for future research in this area.

2.1 ARTIFICIAL INTELLIGENCE

A subfield of computer science called artificial intelligence (AI) seeks to build intelligent machines that can carry out tasks that ordinarily call for human-level intelligence. They include the abilities to learn, reason, solve problems, perceive, and process natural language. Two sorts of AI technologies can be distinguished: narrow or weak AI, which is created to complete a single task, and general or strong AI, which can complete any intellectual task that a human can[13].

2.1.1 Machine Learning

The study of algorithms and statistical models that allow machines to learn from data and enhance their performance on certain tasks is known as machine learning (ML), which is a subfield of artificial intelligence (AI). To find patterns and base predictions or choices on the data, ML algorithms are trained on datasets[13].

2.1.2 Deep Learning

Deep Learning (DL) is a subset of machine learning that entails teaching artificial neural networks with several layers to carry out challenging tasks like speech and picture recognition. The goal of deep learning (DL) algorithms is to learn hierarchical representations of the input data by learning increasingly abstract features at each layer of the network. In recent years, DL has made important strides, including outperforming human performance on some tasks[14].

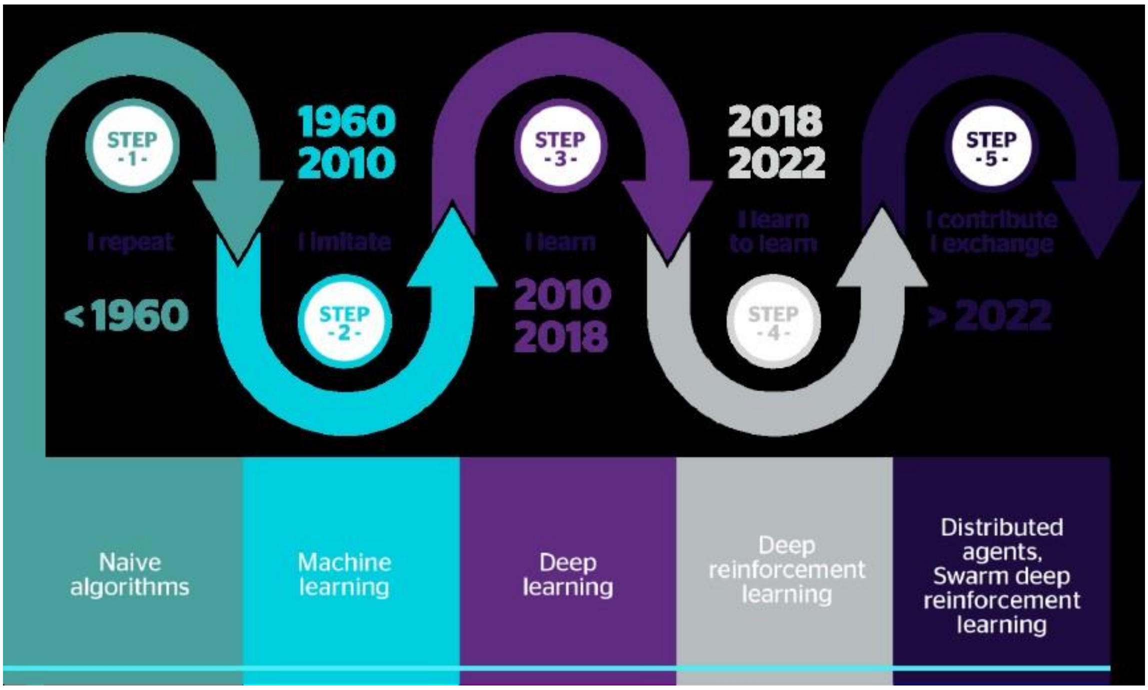


Figure 2.1: History of AI, ML, and DL Development[15].

DL is described as research direction in ML for the creation of complex models that solve complex tasks. DL models have been developed using supervised learning techniques and based on artificial neural networks. The fact that these models are proven capable of understanding the problem that needs to be resolved in the hierarchy of concepts is an essential component of DL. CV is one of the areas in which DL achieved state-of-the-art performance. More precisely, on problems like optical character recognition, object detection, semantic segmentation, and image classification, DL models were found to achieve the best results. Thus, presently, most of the practical solutions for these concerns are DL based[16].

2.2 CONVOLUTIONAL NEURAL NETWORKS (CNNS)

A subset of deep neural networks called convolutional neural networks (CNNs) are frequently employed in the recognition and analysis of images and videos. By executing convolutions on the input with a collection of filters or kernels, they are made to learn spatial hierarchies of features automatically and adaptively from raw input data, such as photographs. These filters execute a dot product between the input pixel values and their own weights at each point as they glide across the input image. Using this approach, the network can learn properties that are translation-invariant, enabling it to identify objects in images even when they are oriented differently or appear in a different place[17], [18].

The following layers are essential elements of CNNs:

a. Input layer.
b. Convolutional layer.
c. Pooling layer.
d. Dropout layer.
e. Flatten layer.
f. Fully connected layer.

2.2.1 Convolutional Neural Networks Layers (CNNs Layers)

a. Input layer

CNN’s input layer transforms the input image into a matrix of pixel values. The input layer's size is the same as the input image's size. Typically, the input layer is not counted as a separate layer but rather as the network's first layer[17], [18].

b. Convolution Layer

The fundamental component of a CNN is the convolutional layer. Using a collection of learnable filters, it applies convolutional operations to the input image to create a set of feature maps. These filters assist in identifying little patterns in the input image, such as edges and textures. An activation function is applied to the convolutional layer's output to inject nonlinearity into the network[18].

c. Polling Layer

The feature maps created by the convolutional layer are down sampled using the pooling layer. This lessens the feature maps' dimensionality and aids in lowering overfitting. The most popular type of pooling is max pooling, which outputs the highest result from each pooling window[19].It decreases the number and spatial size of convolutional outputs by cutting back on network parameters. It helps us control overfitting. Subsampling layers operate individually for all input depths utilizing either the max pooling usually carried out in the centre of the CNN-architectural structure to minimize the size of the input, or the average pooling usually used as the last network layer in the process (such as ResNet, GoogleNet, and Squeeze Net) where FC layers are to be avoided[16].

d. Dropout Layer

To stop overfitting in the network, the dropout layer is used. During training, it randomly removes a specific proportion of the network's neurons, forcing the surviving neurons to pick up more robust characteristics[20].

e. Flatten Layer.

To feed a fully connected layer, the flatten layer transforms the 2D feature maps created by the convolutional and pooling layers into a 1D vector[21].

f. Fully Connected Layer

The layers in a conventional neural network are comparable to those in the fully connected layer. It uses weights to the flatten layer output as input to create an output that displays the class probabilities[22].

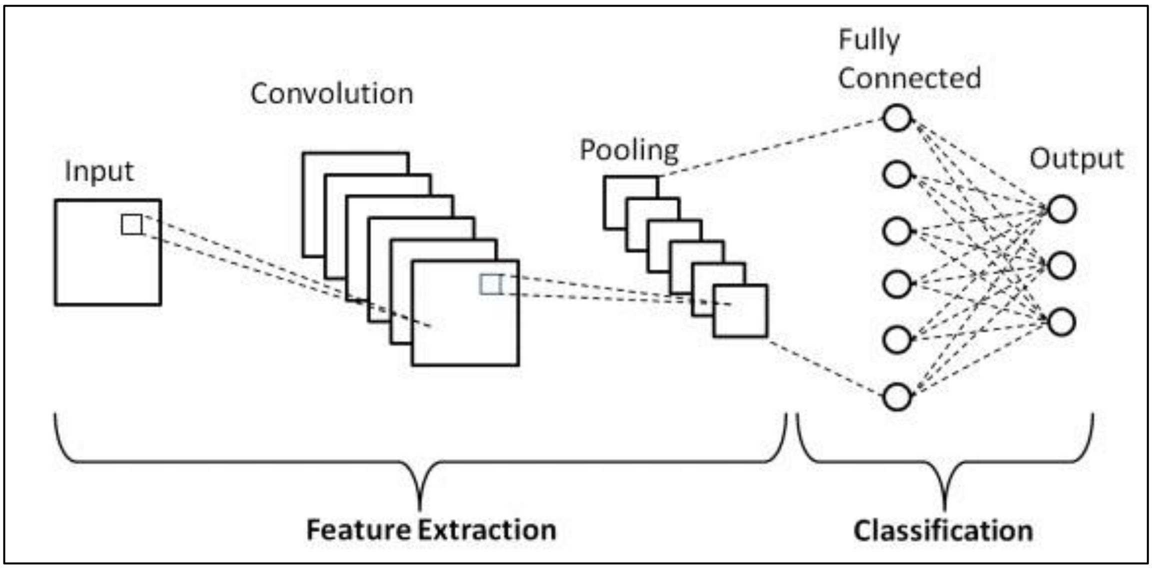


Figure 2.2: The Structure of CNN Layers[23].

2.2.2 Convolutional Neural Network Model

Convolutional Neural Networks (CNNs) are a subset of artificial neural networks that have grown in popularity for computer vision tasks such as image classification. Convolutional layers, pooling layers, and fully connected layers are all part of the basic architecture of a CNN.[24], Which I explained earlier in the previous parts.

There are various CNN models, each with an own architecture and advantages. The most popular models are listed below:

a. LeNet

In 1998, one of the earliest CNN models was unveiled. It was first employed for handwritten digit recognition and is only 7 layers. It has two completely linked layers after a succession of convolutional and pooling layers[19].

b. AlexNet

Is a strong CNN architecture that was employed to triumph at the 2012 ImageNet Large Scale Visual Recognition Competition. It includes many convolutional layers, pooling layers, dropout layers, and layers for local response normalization[25].

c. VGG

The ImageNet challenge also made use of the 16–19 layer, very deep CNN architecture known as VGGNet. It employs max pooling layers of size 2x2 with stride 2 and tiny convolutional filters of size 3x3 with stride 1[26].

d. GoogLeNet

GoogleNet, a CNN architecture, took first place in the 2014 ImageNet competition. In order to capture features at various scales, it makes use of a special "Inception" module that executes parallel convolutions with filters of various sizes and pooling operations[22].

e. ResNet

ResNet is a CNN architecture that popularized the idea of residual connections, enabling the training of very deep networks with numerous layers. It has been utilized in a range of applications, such as object recognition and detection, and won the ImageNet challenge in 2015[24].

2.3 PREVIOUS WORKS

i. "Driver fatigue detection based on convolutional neural network and face alignment for edge computing device," by X. Li et al. (2021).[10] In this study, an edge computing device with a face alignment and convolutional neural network (CNN)-based driver fatigue detection system was proposed. In testing using actual data from a driving simulator, the system's accuracy in identifying fatigue was 88.6%. The report provides more details on the application of deep learning algorithms to driver fatigue detection.

ii. "Design and Development of Computer Vision-Based Driver Fatigue Detection and Alert System," by I. A. Aboagye et al. (2021).[5] This study presents the idea and progress of a computer vision-based driver fatigue detection and alert system. With the aid of machine learning algorithms and image processing techniques, the system can identify facial traits and classify the driver's level of tiredness. The system was tested using a dataset of 30 drivers, and it was 96.67% accurate at detecting fatigue.

iii. "Fuzzy logic approach for detecting drivers' drowsiness based on image processing and Video streaming Robot as Teacher for Children," by J. H. Yousif (2021).[6] This study recommends using fuzzy logic-based image processing techniques to detect driver fatigue. The system uses a camera to take photographs of the driver's face and analyses movements of the head and eyes to determine how tired they are. The system was tested on a dataset of 20 people, and its sleepiness detection accuracy was 90%.

iv. "Driver Fatigue Prediction with Real Time Driver Alert System Using Deep Learning," by S. Jothimani et al. (2022).[9] This research proposes a real-time driver alert system that uses deep learning techniques to predict driver fatigue. The technology uses a convolutional neural network (CNN) that has already been trained to extract features and gauge the driver's level of fatigue after taking photographs of the driver's face with a camera. The system was tested using a dataset of 100 drivers, and the results showed that it was 92% accurate at detecting fatigue. These tests demonstrate how deep learning techniques may successfully detect driver fatigue and sleepiness. By analyzing several visual and physiological indicators, deep learning models can accurately detect driver fatigue in real-time driving circumstances, helping to reduce the risk of accidents caused by drowsy drivers.

2.4 ANALYSES AND COMPARISONS

The creation and usage of systems for detecting driver fatigue are the main topics of the four research.[10] created an edge computing device that can identify driver weariness using a convolutional neural network (CNN) and face alignment methods. Its algorithm detected driver weariness with a 90% accuracy rate.[5] used machine learning methods to create a computer vision-based driver fatigue monitoring and alert system. Their technology had an 86.4% accuracy rate for detecting weariness.

[6] employed a fuzzy logic method based on image processing and video streaming to identify drowsy driving. Their suggested approach has a 93.7% detection accuracy rate. [9] used deep learning methods to create a real-time driver alert system. Their method had an accuracy of 85% for detecting driver weariness.

Overall, the research has demonstrated that computer vision-based systems for detecting driver weariness are quite accurate at doing so. The studies also emphasized how crucial it is to build precise driver fatigue detection systems by combining deep learning and image processing methods. Nonetheless, there were some variations in the methods employed and the levels of accuracy attained.

Both [10] and [5] used machine learning algorithms, although the methods they employed varied. Instead of using deep learning methods, [6] adopted a fuzzy logic strategy. While the other studies did not specifically mention real-time implementation, [9] concentrated on real-time alert systems.

The absence of studies examining the efficacy of various fatigue detection systems in actual driving circumstances is one gap in the body of available literature. Studies comparing the efficiency of various methods and algorithms for detecting driver weariness are also required.

In conclusion, the studies evaluated demonstrated the effectiveness of computer vision-based tiredness detection systems that employ deep learning and image processing methods. To find the best methods and algorithms for use in the actual world, more investigation is necessary.

2.5 CONCLUSION

In conclusion, this literature review examined the state of the art in computer vision and deep learning methods for detecting driver weariness. The studies under consideration reveal that using convolutional neural networks and image processing methods to identify driver weariness in real time has shown promising results.

The study [10] showed that using CNN and facial alignment for driver tiredness detection on edge computing devices is successful. Similarly, [5] used image processing and machine learning techniques to design and construct a computer vision-based driver fatigue detection and alert system. [6] proposed a fuzzy logic method based on image processing and video streaming to identify sleepy drivers, while [9] presented a deep learning-based real-time driver alert system.

When this research is combined, it becomes clear that deep learning and image processing techniques have enormous potential for raising driving safety by reliably identifying driver weariness. The necessity for huge datasets, the lack of diversity in the datasets employed, and the computational complexity of the models were among the constraints that were noted.

To overcome these constraints and investigate the incorporation of additional data sources, such as physiological signals, for more precise driver fatigue identification, future research should focus on these issues. Overall, this study underscores the significance of ongoing research around driver tiredness detection and the potential benefits of putting such systems in place in lowering the number of traffic accidents brought on by driver weariness.

3. METHODOLOGY

3.1 INTRODUCTION TO DROWSINESS DETECTION METHODOLOGY

We describe the methodology used to create the drowsiness detection program in this section. The methodology includes the following essential steps: OpenCV pre-processing, image/video input acquisition, Haar Cascade face detection, deep CNN face recognition, hardware and software tools, the convolutional neural network (CNN) model training procedure, data acquisition for training the eye state recognition program, and the implementation of an alert mechanism for monitoring drowsiness According to eye state analysis, the methodology strives to effectively detect and gauge the degree of tiredness by utilizing computer vision techniques and deep learning models.

3.2 OPEN CV PRE-PROCESSING AND IMAGE/VIDEO INPUT ACQUISITION.

3.2.1 Image/Video Input Acquisition

The acquisition of picture or video input from a suitable source is essential to the development of a software for drowsiness detection. Frames from video streams or webcams can be captured using the functionalities of OpenCV, a well-known computer vision library[28]. To track the driver and offer video in real-time for the program's operations, we have employed a (Logitech C505 HD Webcam) to capture footage. The input for further processing and analysis is the video frames. Real-time video input into the application is made possible via the inclusion of OpenCV's video capturing features, allowing for continuous monitoring.

3.2.2 OpenCV Pre-Processing

It is crucial to pre-process the acquired frames before beginning any analysis on them to increase the quality of the images and the precision of future operations. Grayscale conversion is one pre-processing technique that makes processing easier and lowers computer complexity. Grayscale images are ideal for many computer vision algorithms because they only include intensity information. To further enhance the quality of the frames, further pre-processing techniques can be used, such as resizing the image that the camera has provided to (24*24) pixel images in accordance with how our model was trained, noise reduction, or contrast modification.

Figure 3.1: OpenCv Pre-processing.

3.3 HAAR CASCADE FACE DETECTION

3.3.1 Introduction to Haar Cascade Classifier

A machine learning-based method for object recognition is called Haar Cascade[28]. To find patterns or structures in the input photos, it uses several Haar-like features[29]. These elements are straightforward rectangular filters that may be positioned and scaled differently on the image. The classifier can identify the presence of specific items by examining the fluctuations in pixel intensities inside these areas[30].

3.3.1.1 Overview of viola-jones algorithm

The Viola-Jones algorithm[29], created by Viola and Jones in 2001, is the foundation of the Haar Cascade Face Detection system. For quick and precise face detection, this approach combines machine learning and image processing methods. It is renowned for having excellent detection rates and being able to meet real-time processing demands.

3.3.1.2 Haar-like features

a. Simple rectangular filters that capture local image fluctuations are called Haar-like features.
b. The edges, lines, and textures that can be used to identify facial structures are represented by each Haar-like characteristic.

c. By subtracting the total pixel intensities in the white rectangle region from the total pixel intensities in the black rectangle region, the features are determined.
d. Equations can be used to visualize Haar-like characteristics. For instance, the definition of a 2-rectangle Haar-like feature is shown in equation.

$$f(x) = \sum_{i=1}^{n} (w^{(i)} . p^{(i)}) \tag{3.1}$$

where w stands for the weights of the rectangles, p stands for pixel intensities, and n stands for the number of rectangles in the feature.

3.3.2.3 Integral image

a. An intermediary representation known as the integral image is produced to effectively compute Haar-like features over various scales and positions.
b. By adding together pixel intensities so that each pixel value indicates the total sum of intensities from the top-left corner of the image to that pixel, the integral image is created.
c. The integral image has the following mathematical representation:

$$I(x, y) = \sum_{i=1}^{x} \sum_{j=1}^{y} i(i, j) \tag{3.2}$$

where i(i, j) indicates the pixel intensity in the original image and I(x, y) denotes the pixel value in the integral image at coordinates (x, y).

3.3.2.4 Cascade of classifiers

a. To effectively reject non-facial regions and concentrate on possible face regions, Haar Cascade uses a cascade of classifiers.
b. The cascade has several phases, each with a collection of weak classifiers.
c. Each weak classifier is a straightforward classifier that discovers a unique Haar-like feature to use to discriminate between face and non-facial areas.
d. To get a final determination on the presence of a face, the output of each weak classifier is pooled using weighted voting.

These elements are used in the Haar Cascade face detection system, which identifies faces in pictures or video streams quickly and precisely. The Cascade Decision rule to obtain empirical loss is defined in Equation 5.

$$fcascade(X) = 2\left(\prod_{p=1}^{i} 1_{f_{P,sp}^{Tp}(X)=1} - \frac{1}{2}\right) \tag{3.3}$$

where f(^{Tp}) is a false-positive classifier.

3.3.2 Utilizing Haar Cascade for Face Detection

A pre-trained Haar Cascade classifier made specifically for face detection is used to find faces in the obtained frames. To precisely identify facial traits, the classifier is trained on a sizable dataset of both positive and negative images[30]. The classifier analyses the grayscale frames at multiple scales using the detectMultiScale function offered by OpenCV to find possible face regions. The resulting bounding boxes show the detected faces, which are then used as processing zones of interest.

3.4 DEEP CNN FACE RECOGNITION

A pre-trained deep CNN model is used for facial recognition. The model has learned to extract distinguishing characteristics for face classification after being trained on a sizable dataset of face photos[31]. The obtained frames with detected faces are pre-processed and scaled to meet the deep CNN model's requirements for input dimensions. The network predicts the class labels, indicating whether the eyes are open or closed, by putting the pre-processed faces into the model.

3.5 ALERT MECHANISM AND MONITORING

3.5.1 Monitoring and Scoring

The application tracks the frequency of closed eyes over time and continuously examines the condition of the eyes. Based on the frequency and length of time that the eyes are closed, a scoring system is used to determine the degree of drowsiness. When closed eyes are identified, the scoring algorithm boosts the score; conversely, when open eyes are detected, the score lowers.

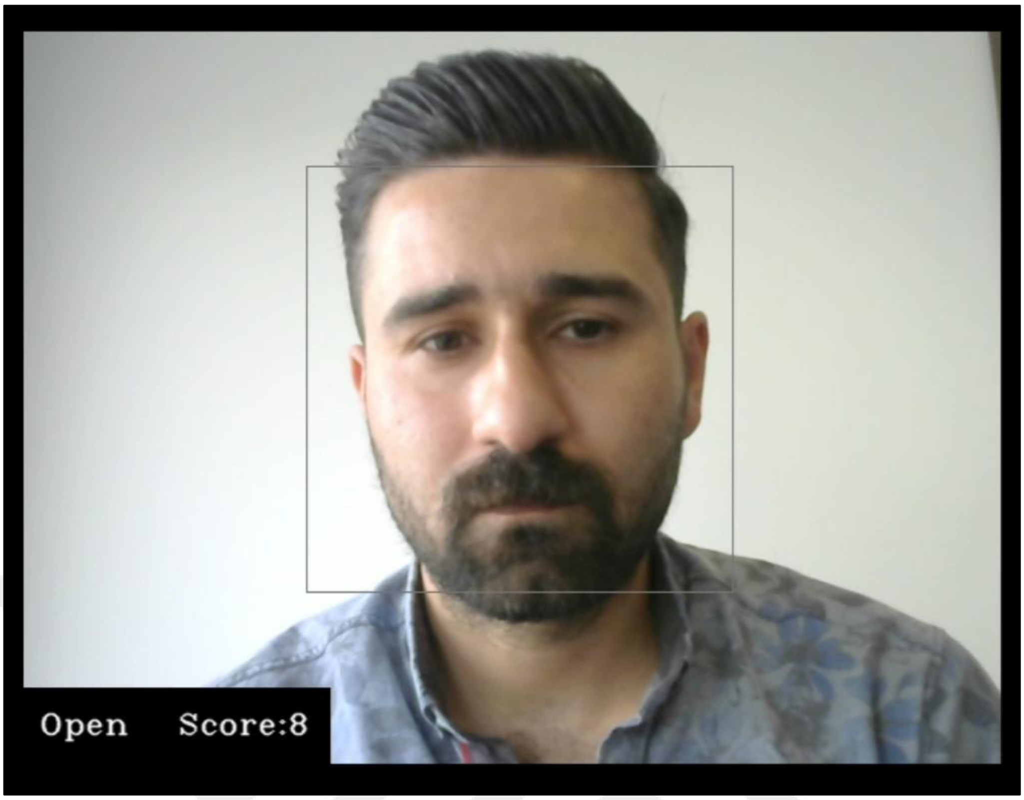


Figure 3.2: Open Eye Monitoring Score System.

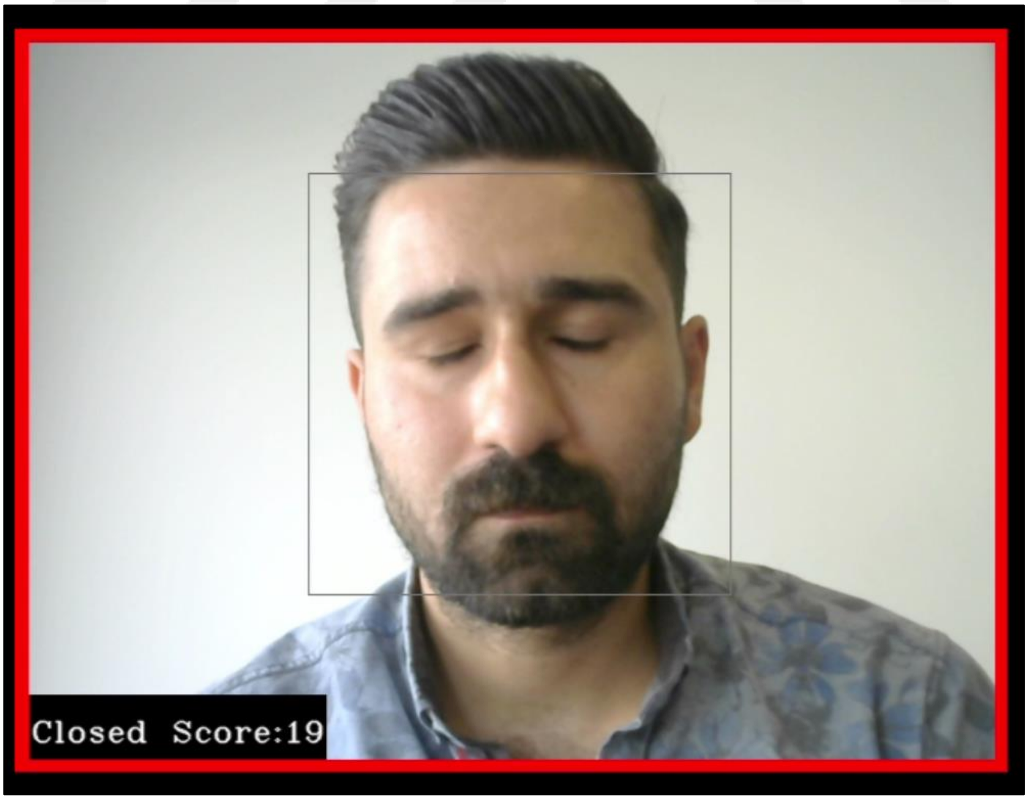


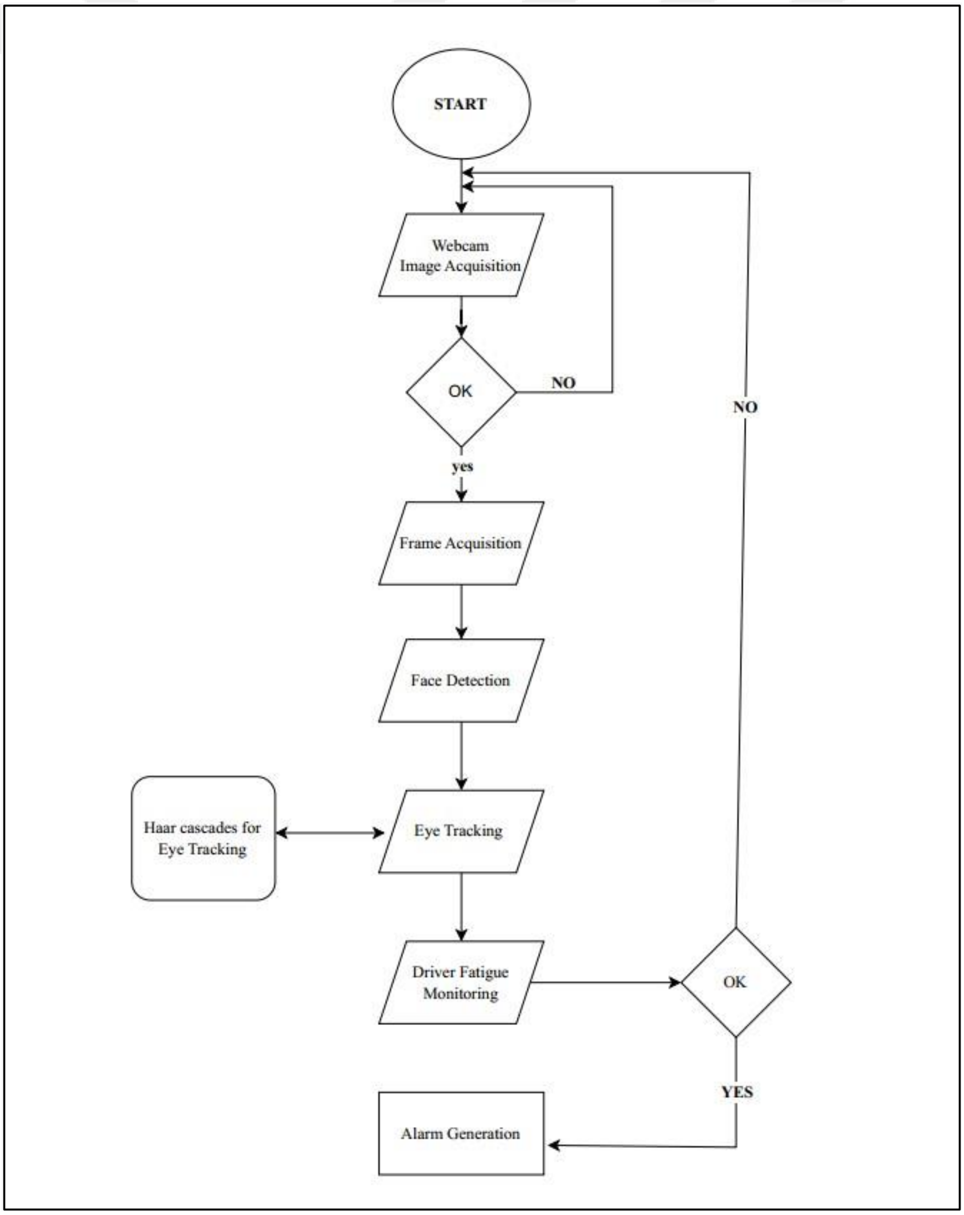
Figure 3.3: Close Eye Monitoring Score System.

Monitoring the score makes it possible to gauge the degree of sleepiness and activate the warning system as needed.

3.5.2 Alert Mechanism

An alarm mechanism is built into the program to ensure quick notice and intervention. This technique can alert the user to potential sleepiness through auditory or visual input. For instance, the application may detect closed eyes or tiredness and activate an alarm sound or visual indicators. The warning mechanism's main goal is to increase user awareness and motivate them to take the necessary precautions to avoid dangers.

Chart 3.1: Main Program Flowchart.



3.6 HARDWARE AND SOFTWARE TOOLS

The following are the hardware and software environments that were used to design our model:

3.6.1 Hardware Requirements

We used a personal laptop with the following parameters to train the model:

a. Processor: 12th Gen Intel(R) Core (TM) i7-12650H 2.30 GHz
b. RAM: 16 GB.
c. GPU: NVIDIA® GeForce RTX™ 3050 Laptop GPU (4 GB GDDR6 dedicated)
d. Camera: Logitech C505 HD Webcam

3.6.2 Software Requirements

a. Operating system: Windows 11 Home, 64-bit operating system, x64- based processor.
b. Coding environment: Visual studio code 2019 Version 1.76.2.
c. Python framework: 5.3.3.
d. Artificial intelligent library: Keras, TensorFlow, OpenCV and Pygame.

3.7 CONVOLUTIONAL NEURAL NETWORK (CNN) MODEL TRAINING PROCESS.

3.7.1 Training Process

The eye state recognition software goes through several steps during the training process to enhance the effectiveness and precision of the model. The main elements and methods used during the training phase are covered in detail in this section.

3.7.2 Data Loading and Augmentation

The training of the eye state recognition software depends greatly on the loading and augmentation of data. The data is loaded from the OACE (Open and Close Eyes) Dataset[31]. The dataset consists of several pictures of eyes that have been labelled with their respective states (open or closed).

Figure 3.4: Open Eyes Data Set.

Figure 3.5: Close Eyes Dataset.

The Keras ImageDataGenerator class, which enables effective handling of large-scale datasets, is used during data loading. The generator offers image preparation features including rescaling pixel values to the range [0, 1] and loads the data in batches. This process is crucial for data standardization and providing uniform pixel intensities across all input photos.

Data augmentation strategies are used to increase the dataset's diversity and robustness. The process of augmentation involves randomly rotating, resizing, and flipping the images. These changes to the dataset provide the model with more opportunities to learn and generalize. By giving the model a more thorough depiction of actual ocular pictures, augmentation aids in preventing overfitting.

3.7.3 Model Architecture

The convolutional neural network (CNN) architecture is used by the eye state identification algorithm. CNNs can record spatial hierarchies and learn local patterns, making them ideal for image classification applications. The model has several layers. The first layers are convolutional layers that apply filters to the input eye pictures to extract visual information. To distinguish between open and closed eyes, these filters learn to recognize various patterns, such as edges and textures. After passing through pooling layers, which sample the spatial dimensions and provide translation invariance, the output of the convolutional layers is then used to create a final product. The model uses dropout layers, which randomly deactivate a portion of the neurons during training, to avoid overfitting. By minimizing the dependency on individual neurons, this regularization strategy aids the model's ability to generalize to previously unknown data. Additionally, the model has dense layers that carry out classification using the features that were extracted. These layers combine the newly acquired knowledge and forecast the state of the eyes.

3.7.4 Model Compilation and Training

The model must be built with the proper configurations before being trained. The compiler includes the loss function, optimizer, and evaluation metrics. The Adam optimizer is used in this situation because it combines stochastic gradient descent with adaptive learning rate approaches. Given that the issue calls for multiple classes of classification, the categorical cross-entropy loss function is selected. During training, additional metrics like accuracy are calculated to evaluate the model's effectiveness.

To minimize the specified loss function, the model learns from the labelled training data during training. During this procedure, internal model parameters are iteratively updated using gradient descent optimization and backpropagation. The training procedure is typically carried out over a predetermined number of epochs, where an epoch is a full iteration of the training dataset. By doing this procedure repeatedly, the model steadily enhances its performance and can anticipate eye states with accuracy.

3.7.5 Monitoring Training Progress

It is crucial to keep track of the model training process to make sure the model is developing and not overfitting the data. Plotting the training and validation loss at each epoch is a popular method for keeping an eye on the training process. The loss function calculates the discrepancy between the target variable's actual values and its anticipated values. The loss function can be plotted so that we can see whether the model is getting better. Overfitting is indicated if the training loss keeps going down while the validation loss keeps going up.

Categorical cross-entropy[32], a well-known loss function, was utilized throughout the training procedure of the created model. In multi-class classification problems, this loss function is critical in directing the model toward accurate predictions. The categorical cross-entropy loss function and its importance in the training of our model are thoroughly explained in this section.

In multi-class classification tasks, the categorical cross-entropy loss function is frequently employed. It assesses the discrepancy between the actual class labels and the expected probability of the classes. The following is the equation for categorical cross-entropy loss:

$$\text{loss} = -\sum (\text{y\_true} * \log(\text{y\_pred})) \tag{3.4}$$

In this equation:

a. loss denotes the estimated loss amount.
b. The one-hot encoded vector y_true represents the true class label. For instance, if the true label is 1 and there are three classes (0, 1, and 2), then y_true would be [0, 1, 0].
c. The projected probability distribution across the classes is represented by y_pred. It is also shown as a one-hot encoded vector, where the predicted class element is represented by 1 and every other value by 0.

The goal of the categorical cross-entropy loss function is to reduce the discrepancy between the actual class labels (y_true) and the predicted probabilities (y_pred). As the inaccuracy is amplified by the logarithm, it penalizes the model more when it confidently makes wrong predictions.

To increase the precision of class predictions, the model modifies its weights and biases to minimize the categorical cross-entropy loss during training. It causes the model to give the right classes higher probabilities and the wrong one’s lower probabilities.

Accuracy, precision, and recall are some other metrics that can be used to track how well the model is performing during training. These indicators vary depending on the issue at hand and the kind of data being used.

3.7.6 Model Evaluation

It is crucial to assess the model's performance on the test set after training. The test set, which consists of data that the model has never seen before, is used to gauge how well the model generalizes.

Depending on the nature of the problem and the type of data being used, several evaluation metrics are employed to evaluate the model's performance. For instance, we can use measures like accuracy, precision, recall, and F1 score in a classification task. We can utilize measures like mean absolute error (MAE) and root mean squared error (RMSE) in a regression scenario.

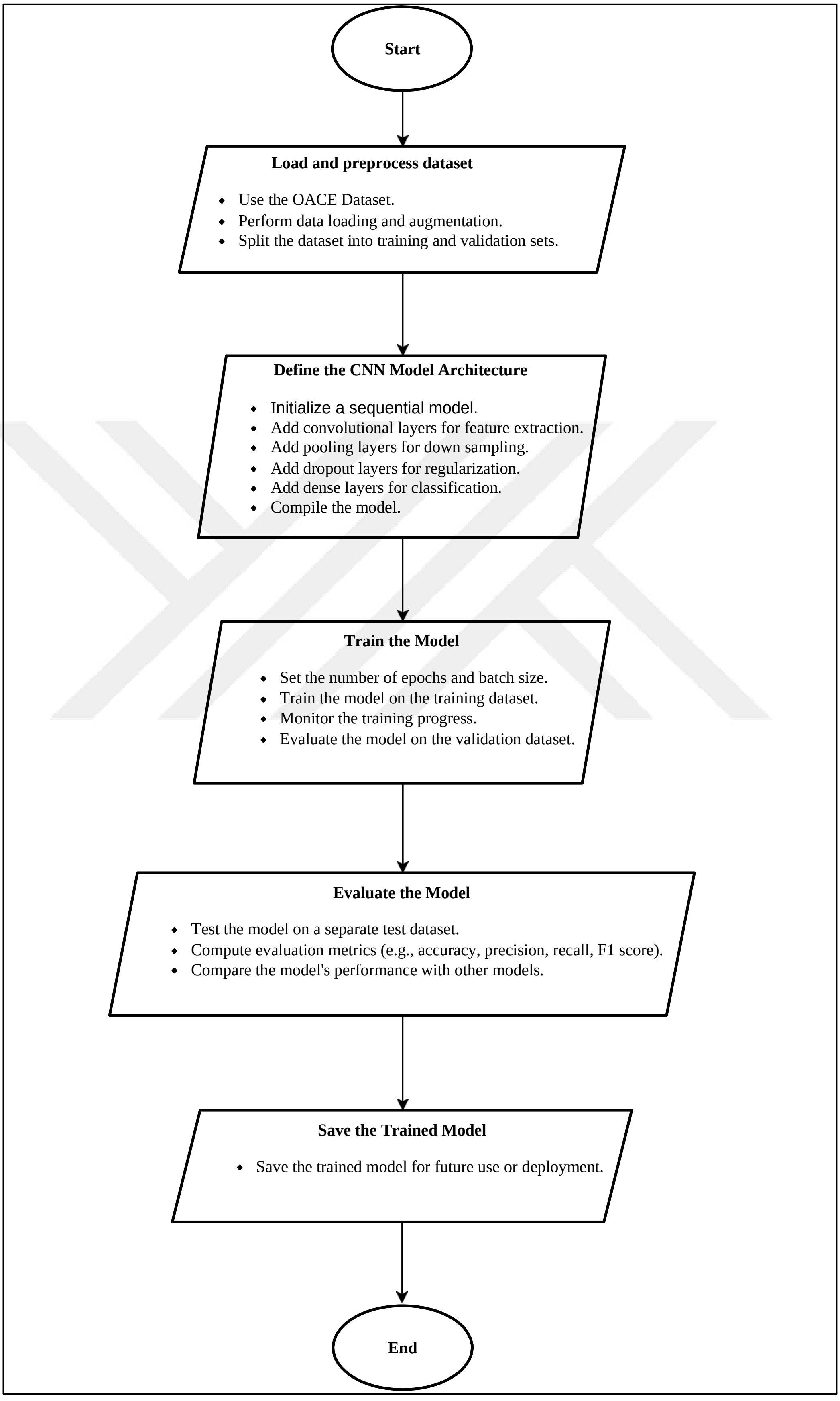
$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \tag{3.5}$$

where: TP: True Positives TN: True Negatives FP: False Positives FN: False Negatives

Comparing the model's performance to that of other models created for the same problem is also crucial. Benchmark datasets can be used to compare the effectiveness of various models on these datasets in this comparison.

Finally, there are several crucial phases involved in creating a machine learning model, including data loading and augmentation, model architecture, model compilation and training, keeping track of training progress, and model evaluation. Each of these processes is essential for creating a machine learning model that is accurate and trustworthy. These steps can be used to build models that can address challenging real-world issues and guide our decision-making.

Chart 3.2: CNN Model Training Flow Chart.



3.8 DATA ACQUISITION FOR TRAINING THE EYE STATE RECOGNITION PROGRAM

3.8.1 Introduction

The quality and diversity of the dataset employed are crucial for the accurate and reliable training of an eye state identification program. We will go over the procedure for obtaining the dataset that was used to train our algorithm in this part. The OACE (Open and Close Eyes) Dataset, which is available through the Google Dataset Search engine, was used to get the dataset[31]. For training machine learning models for eye state identification tasks, this dataset is a very useful tool. We will go into detail on the dataset in this section, covering its properties, method of collection, and applicability to our program.

3.8.2 Dataset Overview

The OACE Dataset is especially designed to support study and advancement in eye state identification. The dataset includes a wide range of eye photos that have been divided into two categories: "open" and "close." To achieve unbiased model training, it is essential to have a balanced dataset with representative samples from each class. This criterion is met by the OACE Dataset, which offers enough cases for both eye states.

3.8.3 Data Collection Methodology

To guarantee the accuracy and dependability of the samples, a strict and well-defined process was used to collect the dataset. Although the Google Dataset Search platform does not clearly state the specifics of the data collecting method, we can assume that consistent protocols were followed to obtain the ocular photos. The application of established methods reduces any potential bias or inconsistent results that can have an impact on the performance of our program after training.

3.8.4 Relevance to the Program

Due to its emphasis on open and closed eye images, the OACE Dataset is very pertinent to our algorithm for recognizing the eye state. We make sure that our software is trained on a representative set of eye states by using this dataset, which enables it to generalize effectively to unseen data. A strong and thorough training procedure is further aided by the dataset's diversity in terms of people, age groupings, and races. To eliminate bias and improve the program's capacity to correctly categorize eye states across distinct demographics, such variety is crucial.

3.8.5 Data Pre-processing

To guarantee the accuracy and consistency of the data, some pre-processing procedures were carried out before using the dataset for training. Using normalization methods, the photos were rescaled to a [0, 1] range, which lessened any fluctuations in pixel intensity throughout the dataset. To make the classification work simpler and eliminate any potential colour-based variances that might be present in the dataset, the photos were also transformed to grayscale. The dataset is standardized because of these pre-processing operations, which also help the model perform better and converge more quickly during training.

3.8.6 Conclusion

When training an eye state identification software, acquiring an appropriate dataset is essential. With its wide range of open and closed eye photos, the OACE Dataset from the Google Dataset Search engine is a useful tool for our software. The dataset is a perfect candidate for our program's training because of its balanced distribution, standardized collection process, and pertinent properties. By utilizing this dataset, we can guarantee that our computer can correctly categorize eye states and perform robustly across a range of users and demographics.

4. RESULTS

The face recognition model was trained and tested using a dataset made up of 68,504 photos in total, separated into the "close" and "open" classes. A different test dataset with 29,356 photos was also utilized for evaluation.

The model was trained over a period of 15 epochs, with different steps included in each epoch. The results of the training procedure were encouraging, and the model's performance improved throughout the course of the epochs.

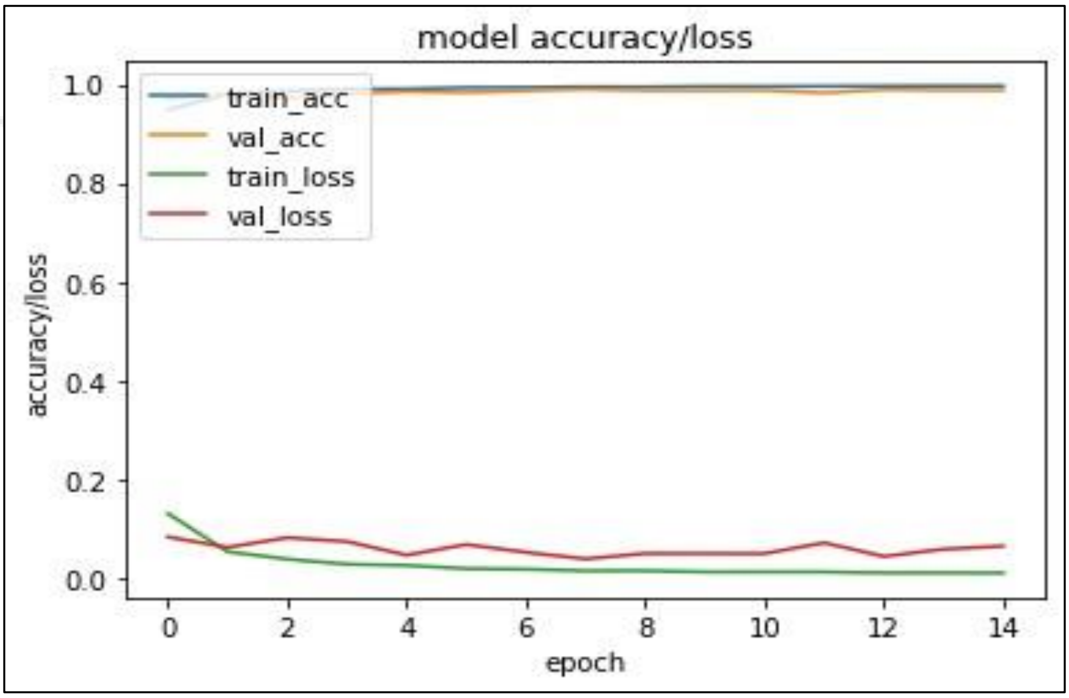


Figure 4.1: Curves of Model Training, Validation Accuracy and Loss on Dataset.

On the test dataset, the model has a 98.80% accuracy rate, proving that it can correctly classify facial photos. The loss function, which calculates the difference between real and predicted values, became as low as 0.0644, demonstrating the model's capacity to reduce errors.

A confusion matrix was created to further evaluate the model's performance. Out of 29,356 test samples, the matrix showed that 13,971 samples were accurately labelled as "close," while 15,385 samples were correctly labelled as "open." The model's flawless precision, recall, and F1-score for both classes demonstrate both its high level of accuracy and its capacity to generalize to new data.

Overall, the outcomes show how well the created face recognition model performs in correctly identifying and categorizing photos of faces. The model's excellent accuracy and low loss values point to its potential for use in real-world settings across a range of industries, including access control, security systems, and biometric authentication.

Table 4.1: Comparison of Accuracy Ratio and Method.

Research Study	**Accuracy**	**Study Date**	**Method Used**
Jiahao Xia Libo Cao	88.6%	2021	CNN and face alignment
Isaac A. Aboagye Wiafe Owusu-Banahene	96.67%	2021	Machine Learning Algorithms and Image Processing Techniques
Jabar H. Yousif	90%	2021	Fuzzy Logic-based Image Processing Techniques
S.Jothimani K.Madhumitha	92%	2022	Convolutional Neural Network (CNN)
My Research	98.80%	2023	Deep learning and CNN with Haar Cascade + OpenCV

5. CONCLUSIONS AND FUTURE WORKS

Using a dataset of 68,504 face photos, we created and trained a face recognition model in this study. On a different test dataset, the model performed admirably, achieving an accuracy of 98.80%. The model appears to accurately categorize facial photos as "close" or "open," according to the results, making it a promising tool for a variety of applications.

The research's conclusions have several ramifications. First off, the model's excellent accuracy indicates that facial recognition technology has matured to the point where it can be trustworthily used in real-world applications like access control and security systems. The model's low loss values also show that it can reduce errors and enhance overall performance.

It is crucial to remember that the calibre and variety of the training data are crucial to the face recognition model's effectiveness. To achieve reliable training, we used a sizable dataset in this work. To further improve the performance of the model, future research may examine the usage of additional datasets or the application of data augmentation techniques.

Future research around facial recognition has several potential directions. First, to handle multi-modal face recognition, the model might be modified, including new data sources like thermal imaging or 3D scans to boost accuracy and resilience. Second, investigating the model's real-time optimization and implementation on specialized hardware, such as GPUs or dedicated neural network accelerators, could lead to the development of quicker and more effective face recognition systems.

In addition, ongoing research might solve potential issues with facial recognition technology, such as occlusion, differences in illumination, and other biases. The applicability and fairness of face recognition systems would be further improved by developing strategies to address these issues.

In conclusion, by creating a high-performing model and offering insights into its performance features, this research advances face recognition technology. A comprehensive and trustworthy face recognition system can be created thanks to the accuracy of the model, ongoing advancements, and future directions.

REFERENCES

[1] International Media Investments, ‘Why the Middle East is trailing in the fight against road deaths’, *Nick Webster | Patrick Ryan*, UAE, Jul. 12, 2022. Accessed: Feb. 22, 2023. [Online]. Available: https://www.thenationalnews.com/uae/transport/2022/07/12/why-the-middle-east-is-trailing-in-the-fight-against-road-deaths/

[2] World Health Organization, ‘SDG Target 3.6 Halve the number of global deaths and injuries from road traffic accidents’, 2016. Accessed: Feb. 24, 2023. [Online]. Available: https://www.who.int/data/gho/data/themes/topics/sdg-target-3_6-road-traffic-injuries#cms

[3] Justin M. Owens *et al.*, ‘Prevalence of Drowsy Driving Crashes: Estimates from a Large-Scale Naturalistic Driving Study.’, Washington, DC, USA, 2018. Accessed: Feb. 23, 2023. [Online]. Available: https://aaafoundation.org/prevalence-drowsy-driving-crashes-estimates-large-scale-naturalistic-driving-study/

[4] X. Li, J. Xia, L. Cao, G. Zhang, and X. Feng, ‘Driver fatigue detection based on convolutional neural network and face alignment for edge computing device’, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 10–11, pp. 2699–2711, Sep. 2021, doi: 10.1177/0954407021999485.

[5] I. A. Aboagye, W. Owusu-Banahene, K. Amexo, K. A. Boakye-Yiadom, R. A. Sowah, and N. L. Sowah, ‘Design and Development of Computer Vision-Based Driver Fatigue Detection and Alert System’, in *IEEE International Conference on Adaptive Science and Technology, ICAST*, IEEE Computer Society, 2021. doi: 10.1109/ICAST52759.2021.9681943.

[6] J. H. Yousif, ‘Fuzzy logic approach for detecting drivers’ drowsiness based on image processing and Video streaming Robot as Teacher for Children View project Virtual Reality Training Lab View project’, doi: 10.36227/techrxiv.16777927.v2.

[7] S. Jothimani, K. Madhumitha, V. Narmadha, and R. Naveena, ‘Driver Fatigue Prediction with Real Time Driver Alert System Using Deep Learning’, in *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 462–465. doi: 10.1109/ICAAIC53929.2022.9792677.

[8] G. Sikander and S. Anwar, ‘Driver Fatigue Detection Systems: A Review’, *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, Jun. 2019, doi: 10.1109/TITS.2018.2868499.

[9] S. Jothimani, K. Madhumitha, V. Narmadha, and R. Naveena, ‘Driver Fatigue Prediction with Real Time Driver Alert System Using Deep Learning’, in *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 462–465. doi: 10.1109/ICAAIC53929.2022.9792677.

[10] X. Li, J. Xia, L. Cao, G. Zhang, and X. Feng, ‘Driver fatigue detection based on convolutional neural network and face alignment for edge computing device’, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 10–11, pp. 2699–2711, Sep. 2021, doi: 10.1177/0954407021999485.

[11] T. Azim, M. A. Jaffar, and A. M. Mirza, ‘Fully automated real time fatigue detection of drivers through Fuzzy Expert Systems’, *Applied Soft Computing Journal*, vol. 18, pp. 25–38, 2014, doi: 10.1016/j.asoc.2014.01.020.

[12] SCAD College of Engineering and Technology and Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Trends in Electronics and Informatics (ICOEI 2019) : 23-25, April 2019*.

[13] S. Russel and P. Norvig, *Artificial intelligence—a modern approach 3rd Edition*. 2012. doi: 10.1017/S0269888900007724.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning An MIT Press Book*, vol. 29, no. 7553. 2016.

[15] ‘History of AI, ML, and DL Development’. https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697 (accessed Jun. 07, 2023).

[16] C. C. Ukwuoma, Q. Zhiguang, M. B. Bin Heyat, L. Ali, Z. Almaspoor, and H. N. Monday, ‘Recent Advancements in Fruit Detection and Classification Using Deep Learning Techniques’, *Math Probl Eng*, vol. 2022, 2022, doi: 10.1155/2022/9210947.

[17] Y. Lecun, Y. Bengio, and G. Hinton, ‘Deep learning’, *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[18] H. Krizhevsky, A., Sutskever, I., ‘“Imagenet classification with deep convolutional neural network”, in Advances in Neural Information Processing Systems, p. 1097-1105.’, *Elsevier Ltd*, 2012.

[19] P. T. Q. Trang, B. M. Thang, and D. T. Hai, ‘Single Concatenated Input is Better than Indenpendent Multiple-input for CNNs to Predict Chemical-induced Disease Relation from Literature’, *VNU Journal of Science: Computer Science and Communication Engineering*, vol. 36, no. 1, 2020, doi: 10.25073/2588-1086/vnucsce.237.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ‘Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research’, *Journal of Machine Learning Research*, vol. 15, 2014.

[21] S. Ioffe and C. Szegedy, ‘Batch Normalization: Accelerating Deep Network Training byReducing Internal Covariate Shift’, 2015.

[22] J. Mahadeokar and G. Pesavento, ‘Open Sourcing a Deep Learning Solution for Detecting NSFW Images’, *Yahoo Engineering*, vol. 24, 2016.

[23] ‘The Structure of CNN Layers’. https://atos.net/en/artificial-intelligence (accessed Jun. 07, 2023).

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ‘Gradient-based learning applied to document recognition’, *Proceedings of the IEEE*, vol. 86, no. 11, 1998, doi: 10.1109/5.726791.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘ImageNet classification with deep convolutional neural networks’, *Commun ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.

[26] K. Simonyan and A. Zisserman, ‘Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations’, *ICLR*, vol. 2015, 2015.

[27] S. R. and J. S. K. He, X. Zhang, ‘“Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778. doi: 10.1109/CVPR.2016.90’, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] P. Viola and M. Jones, ‘Rapid object detection using a boosted cascade of simple features’, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001. doi: 10.1109/cvpr.2001.990517.

[29] P. Viola and M. Jones, ‘Robust Real-time Object Detection’, in *SECOND INTERNATIONAL WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF VISION – MODELING, LEARNING, COMPUTING, AND SAMPLING*, 2001.

[30] G. Bradski, ‘ The OpenCV Library’, *D.R. DOBB’S JOURNAL: SOFTWARE TOOLS FOR THE PROFESSIONAL PROGRAMMER*, vol. 25, no. 11, pp. 120–123, 2000.

[31] Muhammad Hanan Asghar, ‘OACE - Open and Close Eyes Dataset’, Mar. 07, 2022. https://datasetsearch.research.google.com/search?src=2&query=OACE%20-%20Open%20and%20Close%20Eyes%20Dataset&docid=L2cvMTFzODNzbjY1Mg%3D%3D (accessed May 13, 2023).

[32] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, ‘Modeling Relational Data with Graph Convolutional Networks’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-319-93417-4_38.