Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Detection of Driver Drowsiness Employing Deep Learning Primarily Based on Transfer Learning

Author

Md.Tanvir Sarwar

Roll No. 1603072

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Supervised by

Prof. Dr. Md. Al Mamun

Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

To begin with, I would want to thank Allah Almighty for providing me with the opportunity and inspiration to complete our thesis.

We would like to convey our sincere gratitude, appreciation, and respect to my supervisor, Dr. Md. Al Mamun, Professor and Head of the Department of Computer Science and Engineering at Rajshahi University of Engineering and Technology in Rajshahi. In addition to supplying us with the necessary technical guidelines, counseling, and documentation for me to complete the task, he has consistently shown me support, offered me advice, assisted me, and collaborated with me sympathetically whenever he deemed it appropriate over the course of the year. He has also consistently encouraged me. His consistent support was the factor that contributed the most to my success in achieving my objective. If we ever found ourselves in a difficult situation or conundrum, we could call on him at any hour of the day for assistance. This piece of art would not exist in its current iteration if he had not shown his sincere concern for it.

I would also like to take this opportunity to extend my appreciation to all of the professors who work in the Computer Science and Engineering departments at Rajshahi University of Engineering and Technology. They have taken the time and made the effort to provide me with insightful guidance and motivation.

In conclusion, I would want to convey my appreciation to my parents, friends, and well-wishers for the continuous motivation they have provided as well as the various contributions of support they have made to this endeavor.

October 13, 2022 Md.Tanvir Sarwar

RUET, Rajshahi

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

This is to certify that this thesis report entitled ***"Detection of Driver Drowsiness Employing Deep Learning Primarily Based on Transfer Learning"*** *submitted by* ***Md.Tanvir Sarwar, Roll:1603072*** *in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidate own work carried out by him under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

Prof. Dr. Md. Al Mamun
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

External Examiner's Name
External Examiner's Designation
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

ABSTRACT

Due to the fact that each human driver is an individual with their own set of driving qualities, experiences, and feelings, human drivers exhibit their own distinct behaviors and habits behind the wheel. The problem of identifying abnormal human driving behavior has been studied by a number of researchers who have taken the approach of capturing and analyzing the face of the driver and the dynamics of the vehicle through the use of image and video processing. However, the traditional methods are incapable of capturing the complex temporal features of driving behaviors, in addition to being very time consuming and expensive, and they are not user friendly. However, with the advent of deep learning algorithms, a large amount of study has also been carried out to predict and evaluate the behavior of drivers or information connected to their actions using neural network algorithms. This research has been carried out. In this body of work, we make a contribution to the first classification and discussion of Human Driver Inattentive Driving Behavior (HIDB), which we divide into two main categories: driver distraction (DD) and driver fatigue (DF) or drowsiness (DFD). Then, we move on to talk about the factors that lead to another risky driving conduct exhibited by humans, which is known as aggressive driving behavior (ADB). Aggressive driving behavior, often known as ADB, refers to a wide range of risky and belligerent driving habits that frequently result in serious collisions. Abnormal driving behaviors in humans, such as DD, DFD, and ADB, can be caused by a variety of reasons, such as the driver's experience or lack of expertise behind the wheel, age, gender, or even disease. The study of the consequences of these elements that may lead to a degradation in the driving abilities and performance of a human driver is beyond the scope of this work. However, further research on this topic is definitely warranted. After providing some basic information on deep learning and the algorithms that it uses, we will now give an in-depth analysis of the most recent deep learning-based systems, algorithms, and methodologies for the identification of aggressiveness, distraction, and fatigue in human drivers. Through the presentation of a complete and in-depth comparative examination of all of the most recent detection methods, our goal is to acquire an all-encompassing comprehension of the HIADB detection process. In addition, we emphasize the things that are absolutely necessary. In conclusion, we will show and debate a number of key and essential open research challenges that will be addressed in the future.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

This chapter begins with a general introduction to the impact of driver's drowsiness on road accident. Next, we discussed the problems we were trying to solve and why we were doing it. The benefits of our study for future academic research were then underlined. By summarizing the other chapters at the end of this chapter, we discussed how the thesis was organized overall.

1.2 Motivation

It has been demonstrated that sleepiness is closely connected to fatalities that occur as a result of car accidents. The more we educate people about how dangerous it is to drive while sleepy and make it easier for them to take preventative action, the fewer people we can expect to lose their lives in the years to come.

Drowsiness that lasts for long periods of time might put drivers and passengers at risk of being involved in serious car accidents. Because of this, it should be a requirement that drivers be alert at all times during crucial times like sunrise and dusk. Additionally, research has demonstrated that prevention should begin inside an individual's own household (drowsy driving often starts between work hours). Creating an environment in which we are able to govern our desires without infringing on the rights of others can be facilitated by cultivating a culture in which potentially hazardous areas are denoted by signs or in which elevated sidewalks are installed in close proximity to sidewalks that lead to areas with high pedestrian traffic.

Therefore, in this thesis work, we used Deep Learning approaches to prevent this kind of significant loss of life because they are less time consuming, cheaper, easier, and faster to execute than other strategies.

1.3 Problem Statements

According to this difficulty, the detection device has problems such that a costly, bulky, and inconvenient system is required to be employed, and on top of that, it is not user-friendly.

Employers can measure the levels of their employees' performance on the job in a number of different ways. One of these ways is through testing employees' reaction times and short-term memories. While very accurate, approaches like this are obtrusive and not useful in settings like the vehicle or the office.Some have suggested using hats as a substitute for measuring instruments, however this solution is impractical for usage over extended time periods, too.

1.4 Thesis Objectives

The primary purpose of this research is to determine if a driver is in a sleepy or non-drowsy condition in order to reduce the number of car accidents that can be attributed to the underlying cause.Nonetheless, the existing strategies have several drawbacks, which prompted us to propose a new methodology for this purpose. The objectives of our research can therefore be presented in the following order:

- To identify an effective computational approach for determining whether a motorist is experiencing drowsiness.
- In order to assess how well CNN performs according to this methodology.
- To make the evaluation matrix more effective.
- Detecting drowsiness is essential not only for the protection of humans, but also for minimizing losses in industrial production.

1.5 Research Contribution

The following is a list of the primary contributions that this research has made:

- We have made an effort, through more precise tuning, to enhance the performance of the model that is now in use.
- We have tested the proposed model using a number of different datasets.
- The methodology that was employed in this instance is very recent, and it is suitable for the purposes of academic study as a point of reference.
- The model can be altered to produce even better outcomes in the long run, and there are potential for other improvements to be made to it as well.

1.6 Thesis Organization

In the second chapter, driver fatigue is a major cause of road and rail crashes. Tired drivers are more likely to become distracted, resulting in lack of focus. This can lead to unsafe moves like neglecting to secure cargo and other driving blunders. Similarly, autopilot-equipped cars experienced higher rates of driver disengagement due to fatigue. This chapter includes facts about drowsy driving accidents for our research and describes earlier studies from various perspectives.

The next chapter will address feedforward convolutional neural networks, which take convolutional inputs. On connected kernel layers, we use self-similarity convolution. Convolution is employed recursively on deeper layers to provide spatial invariance and capture localized properties from higher resolutions.
This chapter covers convolutional neural network architecture, including its layers, and some popular models. It differs between standard and convolutional neural networks.

Here's the fourth chapter where we explain the background study related to the research domain. Drowsy driving causes many crashes. Studies should be done to help drivers keep their concentration on the road and avoid accidents. The European Society of Biomechanics in Vehicles (EMBC) recently released three research on driver fatigue, sleepiness, and human-fault

events like illness and irregular driving.
This chapter explains earlier research from many angles and evaluates key research terms for our investigation.

Next chapter discusses data processing concepts including p-value, upregulation, downregulation, fold-change, and log transformation. It describes related items and datasets.
This sixth chapter portion includes the thesis's findings, results, analysis, training, validation, and testing procedures. It compares to earlier study.
Last chapter summarizes thesis, evaluates its limitations, and outlines future study.

1.7 Conclusion

In this chapter, we have made an effort to present the significance of the study about tiredness and its relationship to vehicular accidents. In addition to this, we have discussed the significance of our investigation as well as the reasons for our decision to focus our research on the detection of drowsiness in drivers. In addition, in order to bring this chapter to a close, we went over the research objectives, the contribution, and the organization of the thesis.

Chapter 2

Causes and Consequences of Driver Fatigue

2.1 Introduction

The public's understanding of how to reduce the risk of vehicular collisions has undergone a sea change in recent years, resulting in tremendous progress. In this chapter, we covered the repercussions and reasons associated to driver drowsiness, as well as the impact that it has on a large number of lives being lost. In addition, we talked about the several other aspects that are being changed as a result of this one particular cause of losses.

2.2 Factors Contributing to Driver Sleepiness

According to statistics, driver weariness and carelessness are to blame for the majority of fatal collisions. According to the American Automobile Association, fatigued drivers were at blame for 21% of road fatalities and 7% of all accidents [1].The National Sleep Foundation's statistics show that 32% of drivers suffer at least one drowsy driving incident per month [2].A lack of sleep is frequently to blame, although sleep disorders, drugs, alcohol, and night shift employment can all play a role as well[3]. The following is a list of some of the primary causes of sleepiness:

- Those who drive while sleep deprived.
- Drivers who utilize their vehicles for work, such as those who operate tow trucks, tractor

trailers, and buses. [3]

- People who work shifts include those who work overnight or for extended periods of time.
- Drivers who have sleep disorders that are not being treated, such as sleep apnea, which causes them to continuously stop breathing and start it again.
- Individuals who operate motor vehicles while under the influence of drugs that induce slumber.
- Drowsy driving accidents are most common between midnight and six in the morning or in the middle of the afternoon, when sleepiness is at its highest [4].
- Those who have significant difficulties falling asleep, staying asleep, or both, whether they suffer from insomnia or another sleep disorder.
- High-risk teen drivers include those with little expertise behind the wheel and poor sleep hygiene.
- Patients with Parkinson's disease, dementia, epilepsy, chronic heart failure, and other severe medical conditions have all reported experiencing sleepiness. Drugs, whether taken on a regular basis or for the first time, can cause insomnia. This includes benzodiazepines, antidepressants, antihistamines, antipsychotics, and many others. Sleepiness in those who don't have a sleep problem may have avoidable reasons, such as poor sleep habits that contribute to sleep deprivation [5].

In addition to these factors, drowsiness can be attributed to a number of normal, biologically-based human behaviors. These factors play a role in the development of the condition. The following is an explanation of some of them:

2.2.1 Body Clock

Our body has its own internal clock that alerts it to wake up and sleep at the appropriate times. This clock also notifies our body when it is time to sleep. This clock can be found in the top portion of the brain, directly above the location from which nerves depart towards the eyes. The SCN refers to this particular region. The so-called ”circadian rhythms” that occur in our body are controlled by our clock.

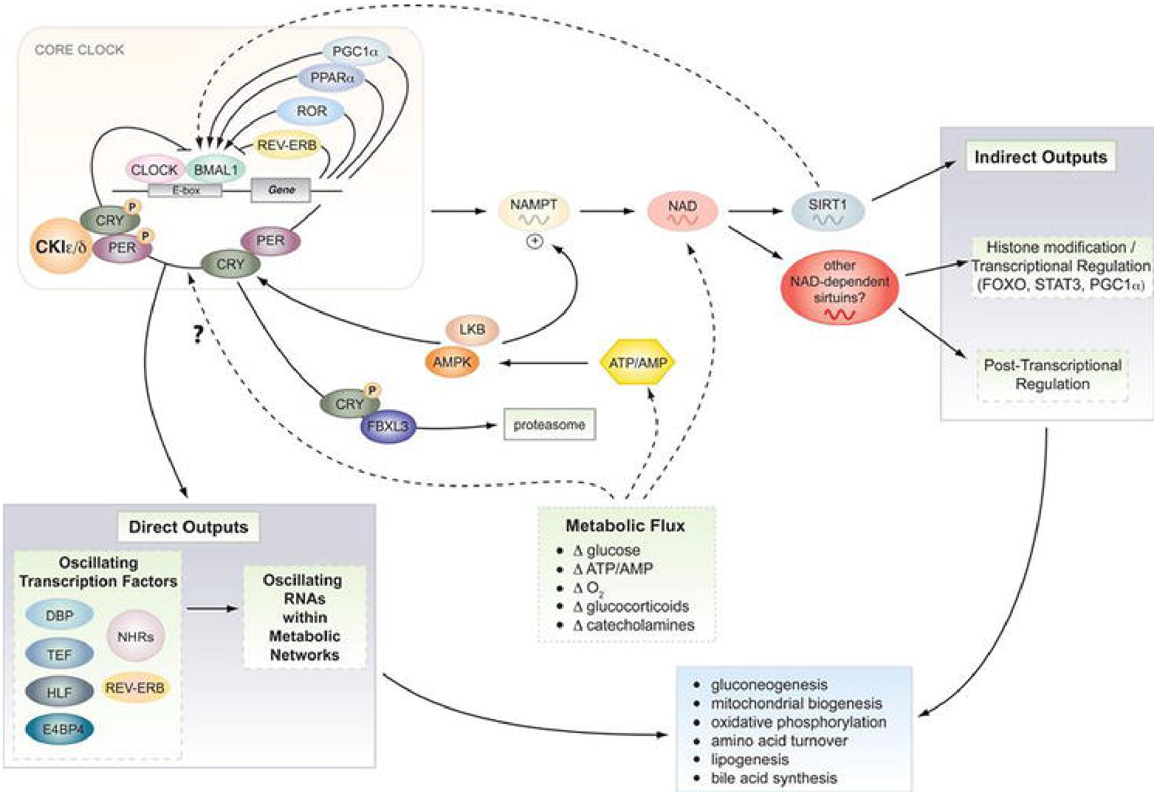


Figure 2.1: Various outputs from the clock's central mechanism[6]

Temperature, alertness, and the 24-hour cycles of a wide variety of hormones are all examples of these rhythms. The term ”circadian” refers to something that happens in cycles that last around 24 hours[7]. Our body's circadian rhythms determine when we are most likely to feel drowsy or awake throughout the day. In the evening, just before bed, we start to feel drowsy. The middle of the day is also when we start to feel sleepy again. This is the time of day when some people like to have their ”siesta”[8].

2.2.2 Chronic Illness

Biological modifications (inflammation, autonomic nervous system activation[9], and dysregulation of the hypothalamic-pituitary-adrenal axis) [10] appear to be linked to weariness in people with chronic illnesses.

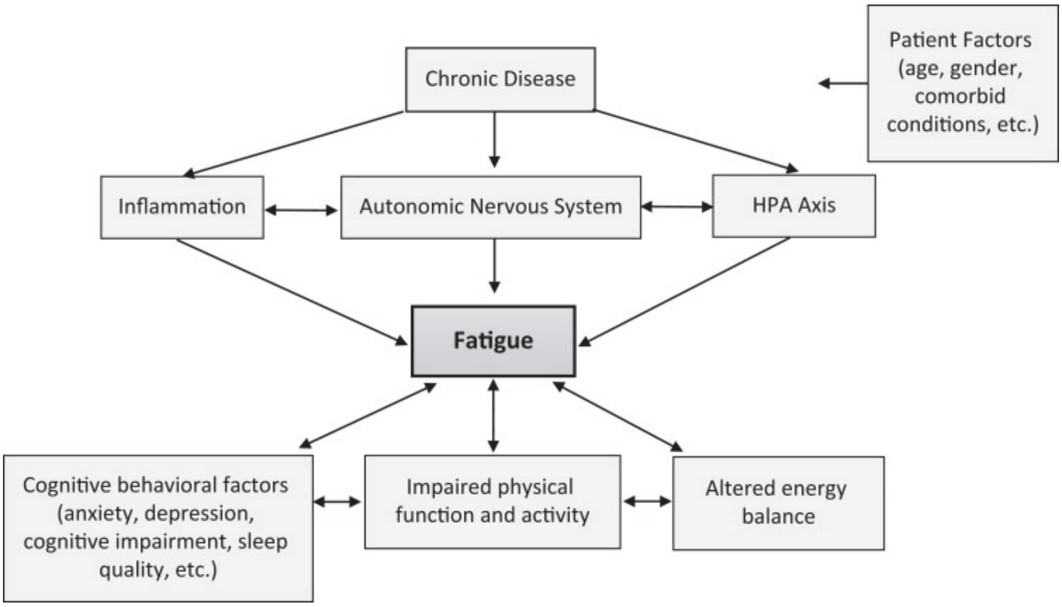


Figure 2.2: The Role of Long-Term Illness in Excessive Sleepiness [11]

The patient's background and other changeable factors can have an impact on these biological shifts. The chain reaction of biochemical reactions depicted in this diagram has the potential to alter cognitive behavioral characteristics, as well as physical function, activity, and energy balance, all of which have an effect on weariness [11].

2.3 Symptoms and Indicators of Drowsiness

The following are examples of common sleepiness patterns and symptoms:

- Having numerous bouts of yawning or blinking.
- Difficulty recalling the distance driven in the most recent few kilometers [12].
- Users failed to take the exit.
- Losing one's bearings.
- Crashing onto a rumble strip by the roadside.

Additionally, a person should seek emergency care if they begin to feel drowsy after they have:

- Start a fresh course of treatment.
- Take an excessively high dose of your medication.

- A brain injury was sustained by the victim.

- Put theirselves out in the freezing weather.

- Use of Sedatives and Other Drugs to Calm the Patient.

2.4 The Repercussions of Being Sleepy Behind the Wheel

Fatigued driving has significantly contributed to a rise in the number of auto accidents over the past few years, and the lack of active safety features in vehicles has become a significant problem as a direct result of this trend. In recent years, there has been a tremendous rise in the number of vehicles owned per capita all over the world. It is common known that methods and techniques for identifying drowsy driving need to be developed. Despite this consensus, little progress has been made. The automobile acts as the backbone of the transportation system. It is of the utmost importance to diagnose and prevent fatigue-related driving, as well as safeguard drivers and reduce the total accident rate of the transportation system. In addition, because driving when fatigued is both a risky and time-consuming activity, it is difficult to monitor and assess its prevalence in the aftermath of a collision. As active safety technology for the identification of driver fatigue continues to be developed on a global scale, there will be an increased need for a fatigue detection algorithm that possesses real-time monitoring, a high level of accuracy, and non-sensitive characteristics [13].

According to statistics, driver weariness and carelessness are to blame for the majority of fatal collisions. According to the American Automobile Association, fatigued drivers were at blame for 21% of road fatalities and 7% of all accidents[1].The National Sleep Foundation's statistics show that 32% of drivers suffer at least one drowsy driving incident per month[2]. According to the US National Highway Traffic Safety Administration (NHTSA), between 2005 and 2009, driver fatigue was a factor in only 2.2% to 2.6% of all fatal incidents in the USA. The outcomes in this study do not include accidents that solely resulted in material damage. According to estimates from 2009, driver drowsiness was a factor in around 30,000 injury accidents (2.0% of all injuries that year)[14].

Figure 2.3: Road Accident [15]

According to a 2017 research by the Foundation for Traffic Safety, 42.4% of drivers in a typical week were operating their vehicles with less than six hours of sleep, which is significant for the majority of drivers (87.9) and is viewed as inappropriate behavior by 95.2% of drivers. However, approximately 3 out of 10 drivers (30.8%) indicate that in the preceding months they drove even when they were too sleepy to keep their eyes open[16].The act of dozing off while driving happens gradually. Driver drowsiness can develop due to monotonous driving conditions and other environmental factors. As a result, the first important problem with the fatigue detection system to be recognized is how to reliably and promptly identify sleepiness[17].

Research on driver behavior and traffic safety is carried out by the National Highway Traffic Safety Administration (NHTSA) of the United States of America. People believe that drowsy driving is responsible for at least 100,000 car accidents each and every year. On average, these collisions cause 40,000 persons to sustain injuries and result in approximately 1,500 fatalities [18].

2.5 Safer Driving Through the Elimination of Tiredness

Paying close attention to one's own actions is the best way to spot signs of sleepy driving. If you find yourself yawning or blinking more often than usual, if you find yourself veering out of your lane, or if you hit the rumble strip on the side of the road, it's time to pull over and rest. There are other techniques to maintain focus and concentration than just drinking coffee and throwing open the window [19].

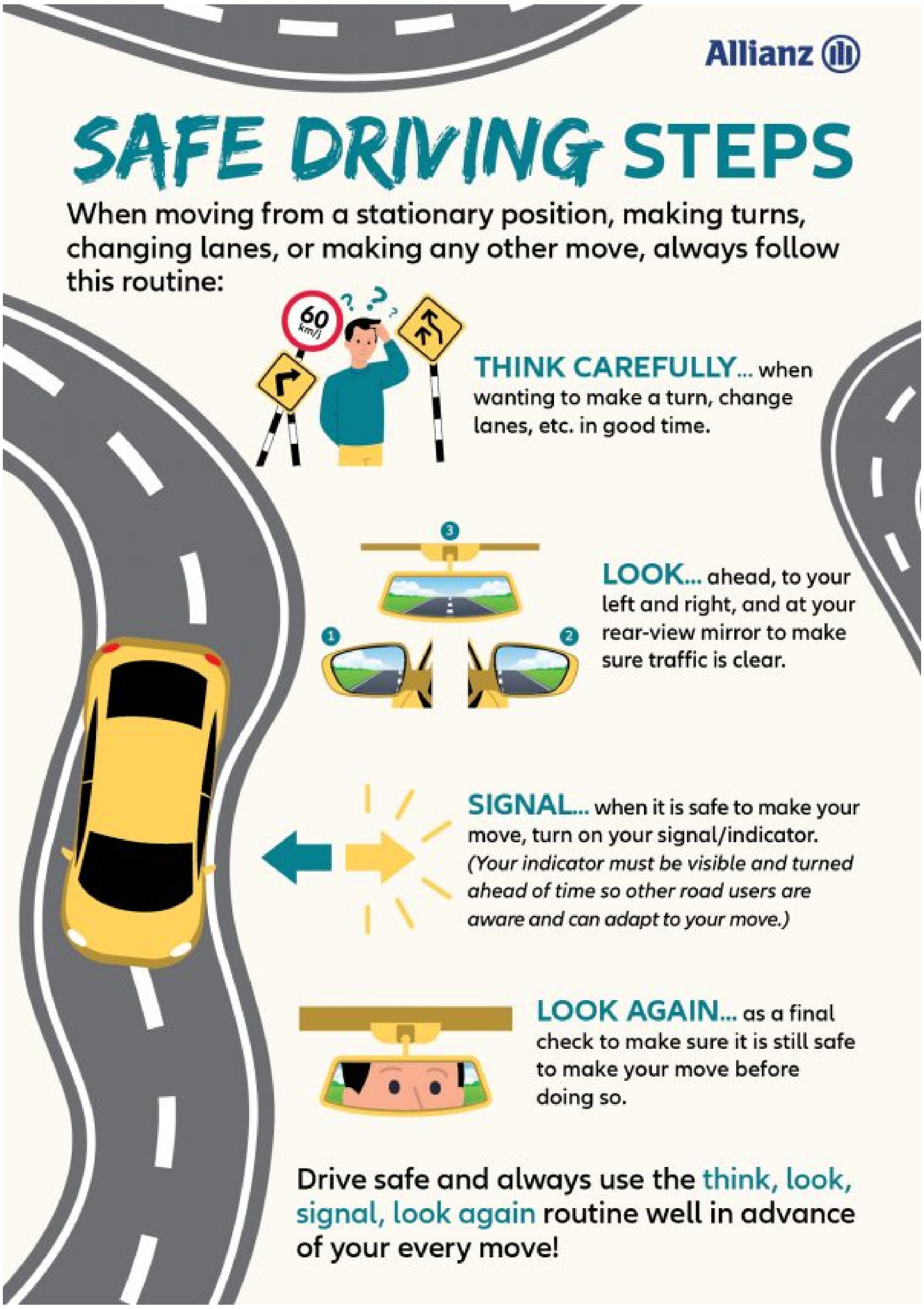


Figure 2.4: Safe Driving Tips [20]

When fatigue is the result of external elements in one's life, such working longer hours, or an internal one, like stress, it may be possible to manage it at home. Drowsiness can often be avoided by maintaining a consistent sleep schedule each night. In addition to this, every driver is expected to abide by the rules that are outlined in the following paragraphs:

- Achieve a healthy sleep routine! Most individuals require 7 or more hours of sleep every night, with teenagers requiring 8 or more [3].
- Create healthy patterns of resting, such observing a regular bedtime.
- Talk to the primary care physician about the many treatment options available to patients if individuals suffer from a sleep issue or exhibit symptoms of a sleep disorder, such as snoring or feeling sleepy during the day.
- Stay away from sleep-inducing substances and activities, such as consuming alcohol or taking certain drugs. When taking any medication, it is important to read the label carefully and to ask the pharmacist any questions they may have.

Chapter 3

Convolutional Neural Networks (CNN)

3.1 Introduction

In this chapter, a convolutional neural network was broken down into its component parts and explained in further detail. In this section, we discussed the many sorts of layers that are used in the architecture of a CNN. In addition to this, we talked about some well-known architectural designs that are frequently implemented for transfer learning in a wide range of different kinds of programs.

3.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks, also known as CNNs, are extremely comparable to the standard Neural Network that was covered earlier. CNN is likewise built up of neurons that contain weights and biases that may be taught to it through training. In order to process the information it receives, each neuron first executes a dot product and then, optionally, a non-linearity. From raw picture pixels at one end to class scores at the other end, the entire network reflects a single differentiable score function. In addition to this, a conventional neural network-style loss function, such as SVM [21] or Softmax [22], is applied to the network's final layer of fully-connected nodes. With ConvNet [23] designs, the fundamental distinction is that the model may embed specific attributes into the architecture because it makes the explicit assumption that the inputs are images. Because of this, the implementation of the forward function becomes much more effective, and the number of parameters utilized by the network is drastically reduced.

Convolutional neural networks (CNNs) are a deep learning architecture that can be taught. Animal vision systems [24], which learn features immediately from the incoming data rather than starting with a predetermined template, have been a major inspiration here. The convolutional neural network (CNN) is made up of numerous nonlinear transformations that take place over the course of several steps. At each step, CNN strives to reduce the amount of error so that it can more properly identify data. In conclusion, a convolutional neural network (CNN) consists of several trainable layers stacked on top of one another like a data stack, with a supervised classifier [25] sitting at the top. Feature vectors arrays, which contain both input and output vectors, are used at each level [26]. In order to train the network, CNN requires a significant amount of labeled data as well as computational capacity. The amount of time needed to train a network is getting much shorter compared to what it used to be as a result of the expansion of the amount of digital data that is currently available and the powerful computational resources, such as graphics processing units (GPUs) [27]. With this convenience, we can train more complex CNN architectures in our setting, increasing the likelihood of a successful end result.

3.3 The Structural Nucleus of a Conventional Neural Network

Let's start with the big picture and describe how a CNN network converges before diving into the specifics.

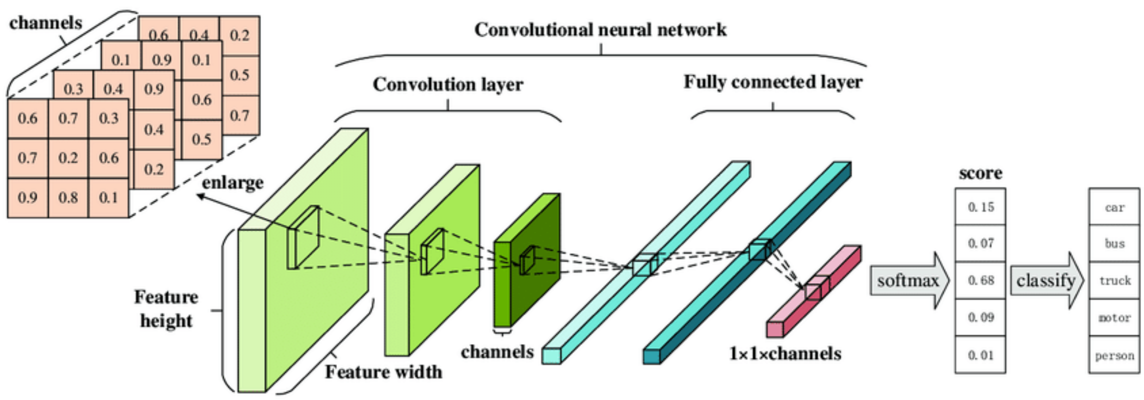


Figure 3.1: Architecture of a Convolutional Neural Network Model [28]

The initial few of layers of this structured neural network [29] are sequentially integrated in order to process visual information. The network needs only minimum preprocessing because of its structured nature. Because the output of one layer becomes the input of the following layer, CNN can be thought of as having a feed-forwarding structure [30]. Learnable parameters, such as weights and biases, are associated with each neuron in the CNN. The forward pass kicks off the process of the network training itself. The volume is transformed from one form into another by a series of interconnected layers. Class scores are represented as probabilities, and these probabilities are used to make the prediction at the output layer. After making a prediction, it is possible to calculate the error by comparing the result to the actual one. The gradient in backpropagation [31] originates from the error that was calculated and moves in the opposite direction of the original error flow. At each stage of the process, the parameters are adjusted in such a way that it makes an effort to cut down on the error that was generated in the stage before [32] [33] [34]. Model convergence is achieved through many iterations of this approach.

Since the input is a collection of images, Convolutional Neural Networks are able to use that fact to their advantage by placing sensible limitations on the network's construction [35]. For one thing, a ConvNet's [36] neurons are not uniformly spaced across its layers like they are in a traditional Neural Network; instead, they are spread out in width, height, and depth. (It is important to distinguish between the depth of a full Neural Network, which can relate to the entire number of network layers, and the depth of an activation volume, which simply refers to the third dimension of an activation volume.) Instead of being connected to all of the neurons in the layer below it in a comprehensive manner, the neurons in a layer will only be connected to a tiny portion of the layer below it. Layers are the component parts that make up a ConvNet [36]. Every Layer has a straightforward application programming interface (API) [37], which consists of the following: taking as input a three-dimensional volume and producing, as output, another three-dimensional volume using a differentiated function that may or may not have parameters.

3.3.1 Comparative analysis of the Deep Neural Network and the Conventional Neural Network

The word "deep" is typically used in marketing to make something sound more professional than it would otherwise. There are a great many different varieties of deep neural networks in addition to CNN, which is a form of these networks. Image identification is one of the many fields in which CNNs have proven to be particularly useful, which has contributed to their widespread use.

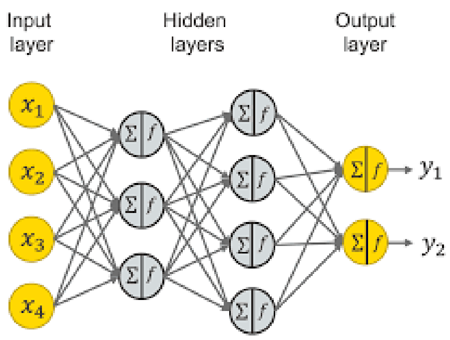


Figure 3.2: Architecture of a Deep Neural Network Model [38]

A CNN model trained from scratch, a classical classification model with hand-crafted features, and three transfer learning models constructed from pretrained CNN models called ResNet50 [39], InceptionV3 [40], and Xception [41] are the five models that have been suggested. The CNN model was trained from scratch. A deep feature-combining model was created at the end of the process using an ANN model and deep features taken from the three transfer learning models described earlier.

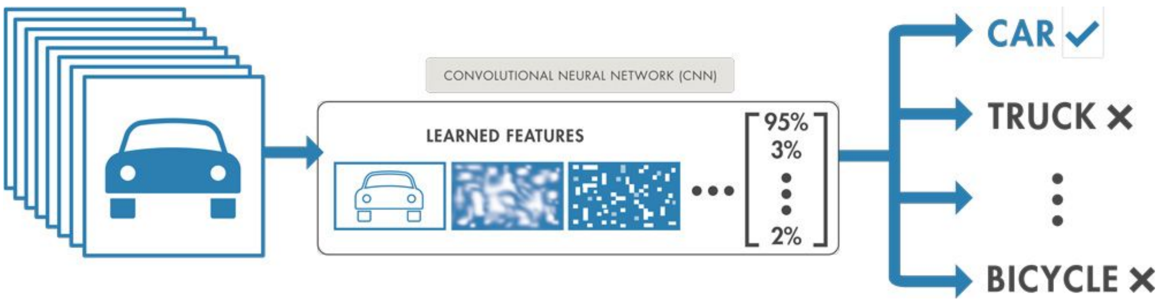


Figure 3.3: The workflow for deep learning. The CNN is fed images, and it uses those images to train itself to recognize features and categorize objects automatically. [42]

A deep neural network model for the classification of breast masses on a small-scale ultrasonic image dataset may be effectively built by transferring an InceptionV3 [40] model that had been pretrained on a large-scale natural image dataset. In addition, merging transferred information from several different CNNs is a potential method that could further increase classification accuracy [43].

3.4 Various layers in CNN

Every layer of a ConvNet [36] is responsible for transforming one volume of activations into another by the application of a differentiable function. A straightforward ConvNet [36] is composed of a series of layers. Convolutional Layers [44], Pooling Layers [45], and Fully-Connected Layers [46] are the three primary types of layers that are essential for the construction of ConvNet architectures (exactly as seen in regular Neural Networks). Construct a whole ConvNet architecture by piling these layers on top of one another.

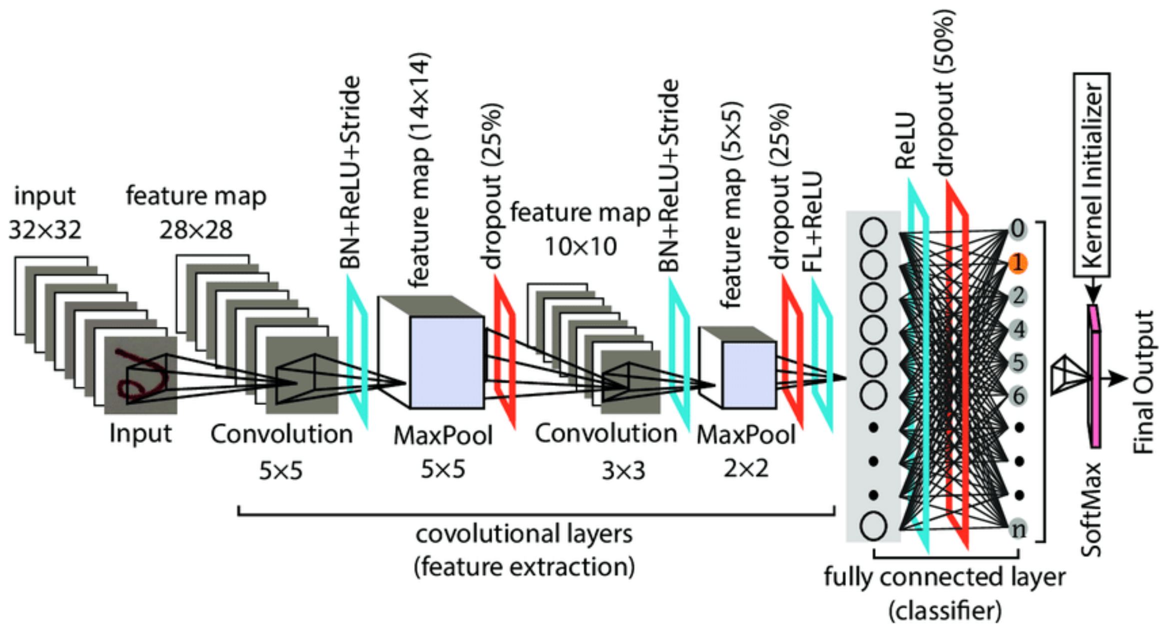


Figure 3.4: Simple Model of a Convolutional Neural Network (also known as CNN). [47]

The graphic provides a description of the activations that constitute one example of a ConvNet design. The initial volume is responsible for storing the unprocessed picture pixels (on the left), and the final volume is responsible for storing the class scores (right).

3.4.1 Convolutional Layer

The Conv layer is the fundamental component of a Convolutional Network and is responsible for the majority of the work that needs to be done in terms of computation. It does this by obtaining image features using tiny squares of data that are presented as input, which preserves the correlation that exists between the pixels. The mathematical operation of convolution takes as its inputs a filter and a matrix representation of a picture. In this step, the result is sent to the feature map [48]. When applied to a picture, the convolution layer's use of filters enables it to execute a variety of functions, including the detection of edges, the sharpening of details, and the blurring of background areas.

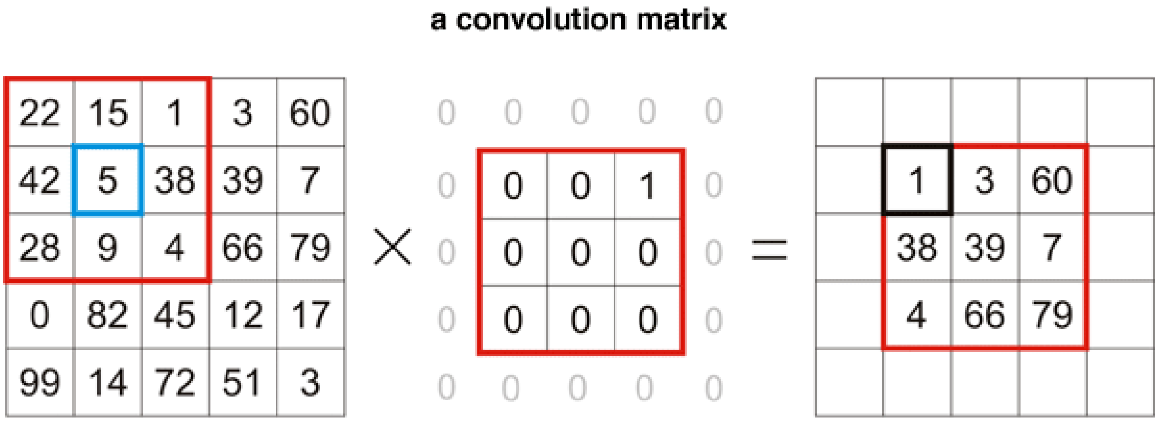


Figure 3.5: Convolutional operation [49]

In order to learn, the Convolutional Layer employs a set of filters. Those filters can be used to extract features from the input photos of many different kinds [50]. By applying those filters to the input photos, a wide variety of features can be retrieved.

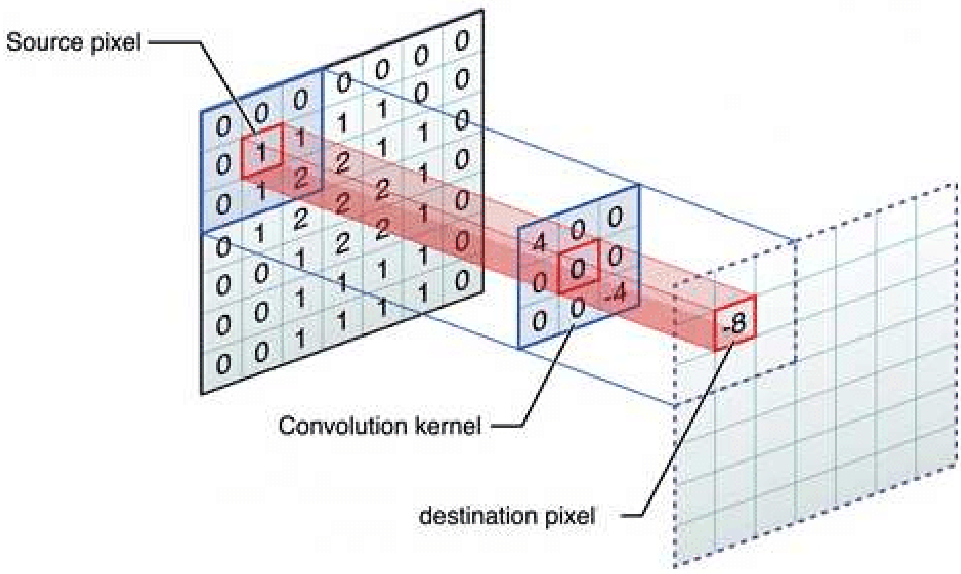


Figure 3.6: Sample generation of a 5 × 5 feature map as the result of the 2D discrete convolution of a 7 × 7 input image with a 3 × 3 filter. [51]

3.4.2 Activation Layer

The convolution layers come next after the activation layer, which is when the network is given its first taste of its non-linear characteristics. Their primary function is to transform the signal that is received at a node's input into the signal that is sent out from that node. The output that was generated is going to be transmitted to the subsequent layer. Each activation function (also known as a non-linearity) begins with a single integer and then applies a predetermined mathematical operation to that number. There is a diverse selection of activation functions available for your selection. This discussion will focus on two different types of activation functions.

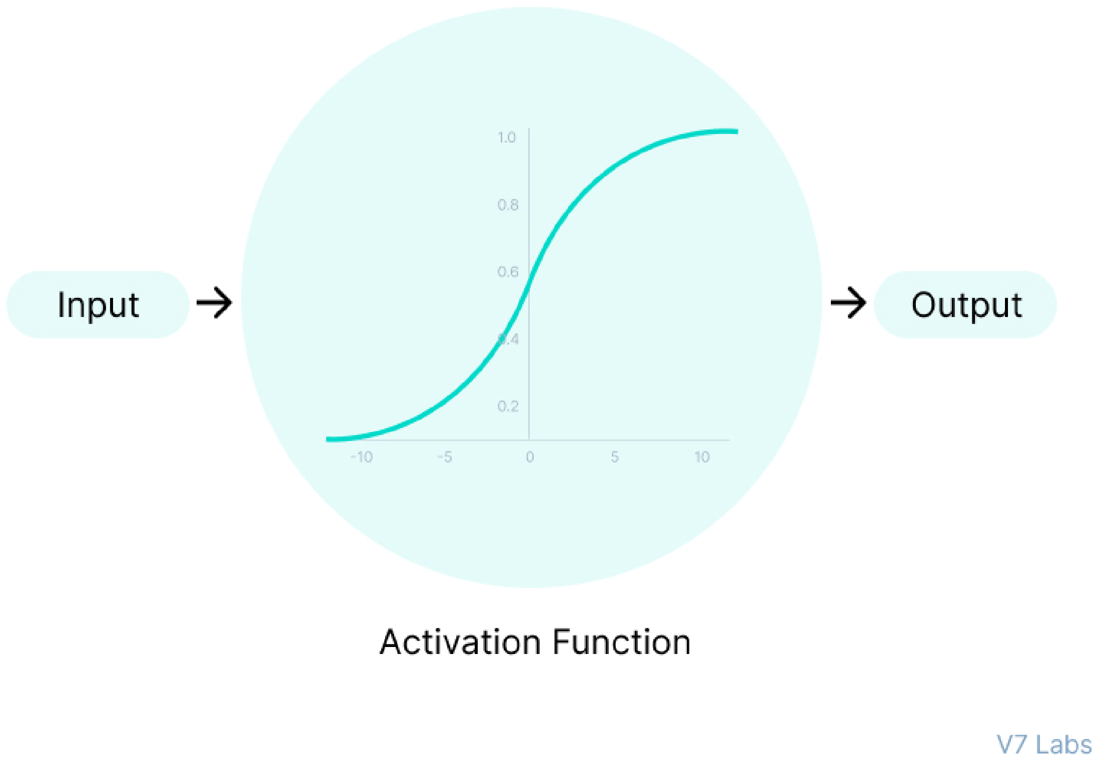


Figure 3.7: Activation Function [52]

- **Sigmoid:** Mathematically, the sigmoid non-linearity takes the form $\sigma(x) = \frac{1}{1+e^{-x}}$. Squashing a real number into the interval between zero and one. In particular, zero is substituted for large negative values, while one is substituted for large positive ones. The sigmoid function has been widely used throughout history due to its convenient representation of a neuron's firing rate, which can range from zero (no firing) to infinity (highest possible firing) (1). While once widely employed, sigmoid non-linearity is now largely obsolete in practice. There are two key downsides to this approach:

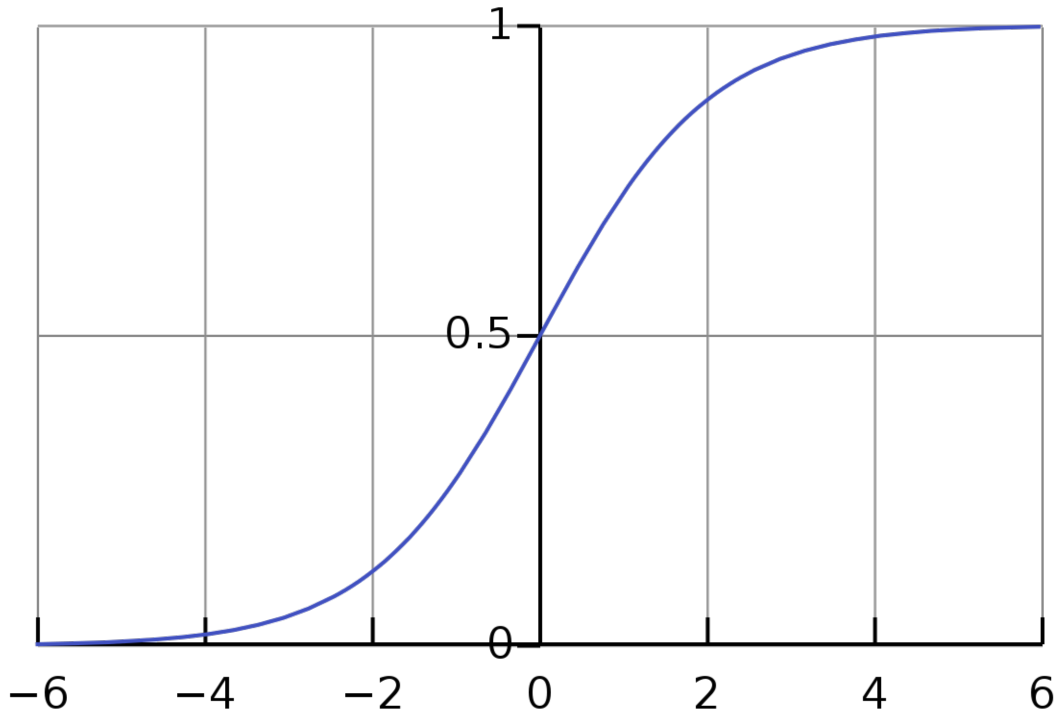


Figure 3.8: Sigmoid Function [53]

- The sigmoid functions' outputs are not centered around zero. This is undesirable because it would result in non-zero-centered data being sent to neurons in the deeper levels of a Neural Network (more on this shortly).
- When the activation of a sigmoid neuron saturates at either tail of 0 or 1, the gradient in these areas is almost zero, which is a characteristic of the sigmoid neuron that should be avoided at all costs. Therefore, if the local gradient is very low, it will essentially "kill" the gradient, and almost little signal will flow through the neuron to its weights and, ultimately, to its data. This occurs because the local gradient effectively "kills" the gradient when it falls below a certain threshold.

- **ReLU:** Over the past few years, the Rectified Linear (ReLU) Unit has seen a significant surge in popularity. It converges substantially faster than the majority of other activation functions and is efficient from a computing standpoint [54]. It makes the computation of the function, $f(\mathrm{x}) = \max\ (0, x)$. To put it another way, the activation occurs when the threshold is set to zero (as seen in the graphic on the left of the previous page). It's important to weigh the benefits and drawbacks of using the ReLUs:
  - The ReLU can be implemented by simply thresholding an activation matrix at zero,

in contrast to sigmoid neurons which need costly computations (exponentials, etc.).

  - When compared to sigmoid functions, it was discovered to significantly hasten the convergence of stochastic gradient descent.
  - Regrettably, ReLU units are susceptible to ”dying” during training. For instance, a ReLU neuron’s weights could update in such a way that it never activates on any datapoint again if a very big gradient were to run through it. If this occurs, the gradient inside the unit will be 0 from then on. In other words, the ReLU units are vulnerable to permanent destruction if they are knocked off the data manifold at any time during the training process.

- Tanh, maxout, leaky ReLU, etc. are examples of alternative activation functions.

3.4.3 Pooling Layer

When building a ConvNet, it is usual practice to intersperse Pooling layers between the Conv layers. Its job is to keep the network from becoming overly-trained by gradually shrinking the size of the spatial representation, which in turn uses fewer resources and prevents overfitting. Each depth slice of the input is processed separately by the Pooling Layer, which then uses the MAX operation to spatially enlarge the data. A pooling layer with filters of size 2x2 applied with a stride of 2 downsamples every depth slice in the input by 2 along both width and height, which results in 75% of the activations being discarded. This is the most frequent form of this type of layer [55]. In this particular scenario, each and every MAX operation would involve taking the maximum value over a set of four numbers (a small 2x2 rectangle located within some depth slice). The dimension of depth has not been altered in any way. In a broader sense, the pooling layer is composed of:

- Allows for the entry of a given volume $W_1 * H_1 * D_1$
- Calls for the following two hyperparameters:
  - Their spatial extent F,
  - The stride S
- Generates a volume proportional to its size $W_1 * H_1 * D_1$ where:
  - $W_2 = \frac{W_1 - F}{S} + 1$

- $H_2 = \frac{H_1 - F}{s} + 1$
- $D_2 = D_1$

- Because it is computing a constant function based on the input, it has no parameters that it introduces.

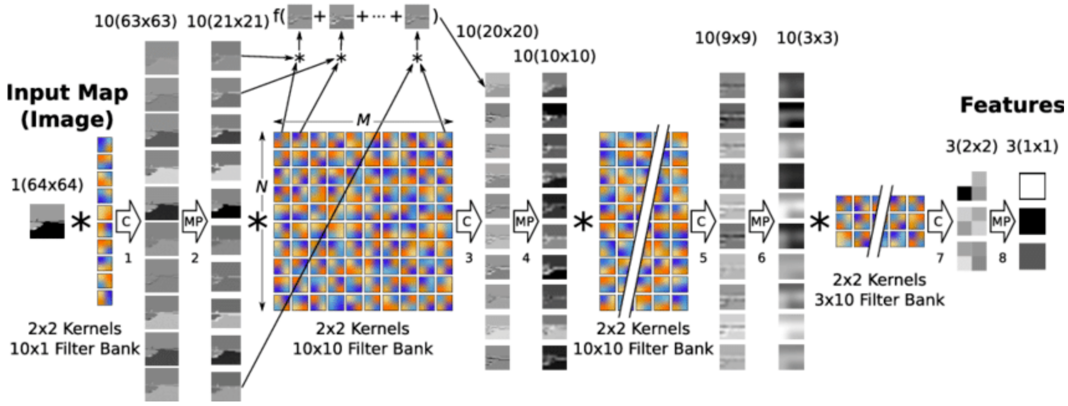


Figure 3.9: Example of Max-pooling CNN [56]

The input volume is spatially downsampled by a factor of the pooling layer's depth, however this is done independently for each depth slice.

3.4.4 Fully-Connected (FC) Layer

Like standard Neural Networks, a fully connected layer's neurons are directly connected to all of the activations in the layer below them. Therefore, a matrix multiplication and a bias offset can be used to calculate their activations. Notably, the neurons in the CONV layer only connect to a local region in the input, and many of the neurons in a CONV volume have the same parameters, but this is the only distinction between the FC [46] and CONV layers. Nevertheless, neurons in both levels continue to compute dot products, therefore the functional form of the neurons in both layers is the same. Because of this, it has been discovered that the FC and CONV layers can be converted into one another:

- The corresponding forward function can be implemented by an FC layer for any given CONV layer. As a result of local connectivity, the weights in many blocks are equal, hence the weight matrix is largely nil (due to parameter sharing).

- It is also possible to transform any FC layer into a CONV layer. For instance, a CONV layer with the parameters F=7,P=0,S=1,K=4096 can be written in the same way as an FC layer with K=4096 that is inspecting an input volume of size 77512. In other words, we are making the filter size equal to the size of the input volume, therefore the outcome will be the same as the first FC layer—namely, 114096. This is because only a single depth column ”fits” across the input volume.

The net output is generated by the final layer of the network, which is fully connected, using an activation function known as the softmax function. The number of classes in the classification issue determines which activation function is used.

3.5 CNN’s Known Architectural Components

Within the realm of convolutional networks, there are a few different architectures that have been given a name. These are the most prevalent ones:

- **LeNet:** Yann LeCun is credited with developing the very first successful implementations of convolutional neural networks in the 1990s. The LeNet architecture [26] is the most well-known of them, and it was utilized to read things like zip codes, numerals, and so on.

- **AlexNet:** AlexNet [57], created by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton, is widely credited as the first work to popularize Convolutional Networks in Computer Vision. While the Network’s architecture was very similar to that of LeNet, it was larger, more complex, and made use of many Convolutional Layers (whereas before, only a single CONV layer was often used, always followed by a POOL layer).

- **GoogLeNet:** Google’s Convolutional Network developed by Szegedy et al. won 2014’s ILSVRC (Large Scale Visual Recognition Challenge). Its key addition was an Inception Module that drastically decreased network parameters (4M vs. 60M for AlexNet). In addition, the Fully Connected layers at the top of the ConvNet are replaced with Average Pooling in this article, allowing for the removal of numerous parameters that do not appear to play a significant role [58].

- **VGGNet:** The network created by Karen Simonyan and Andrew Zisserman, which later became known as the VGGNet [59], finished in second place in the ILSVRC 2014 competition. The most important thing that it contributed was demonstrating that a network’s depth is an important factor in determining how well it performs. Their best and final network has 16 CONV/FC layers and, enticingly, boasts a highly homogenous architecture that, from the beginning to the finish, only executes 3x3 convolutions and 2x2 pooling.

- **ResNet:** The Residual Network [60], which was designed by Kaiming He and colleagues , was the winner of the ILSVRC 2015 competition. It makes extensive use of batch normalization and has a specific kind of connection called a skip connection. In addition to this, there are missing fully connected layers at the end of the network in the architecture. The reader is also directed to Kaiming's lecture (video, slides), as well as some recent experiments to duplicate these networks in Torch. ResNets are now the Convolutional Neural Network models that are considered to be far and away the state of the art, and they are the option of choice when it comes to actually putting ConvNets to use in practice.

3.6 Conclusion

The convolutional neural network, or CNN, is a variant of the standard neural network; however, its inputs are matrices with many dimensions. CNN architectures are often created primarily by a few significant layers, which are then stacked one on top of the other. In order to extract features, the convolution and max-pooling layers are utilized, whilst classification is accomplished with the help of the dense layers.

Chapter 4

Background Research and Literature Evaluation

4.1 Introduction

The definitions of the terminology that are necessary for this research are provided in this chapter. In the last chapter, ”The Second Chapter,” we talked about the biochemical factors that contribute to drowsiness. In the following section of this chapter, we are going to talk about the popular and recent research papers and works that are relevant to this topic, as well as the benefits and downsides of these topics. In conclusion, we will talk about the subject matter of our investigation.

4.2 Exacerbated the Issue of Driver Fatigue

Studies have demonstrated that tiredness and hypovigilance are common while driving on the highway, and that they may have significant ramifications in terms of the likelihood of an accident occurring as a result of their presence. A study employing a driving simulator was carried out to evaluate the impact of the monotony of roadside visual stimulation using a steering wheel movement (SWM) [61] analysis approach. This study was carried out in order to analyze the influence of the monotony of roadside visual stimulation. Sixty-two male participants drove for 40 minutes each throughout two separate sessions. One scenario featured largely boring visual cues along the roadside, while the other featured a variety of visual components designed to break up the routine without altering the physical layout of the road itself. Comparisons of driving performance were made between various situations to see if breaking up the routine had any beneficial effects on subjects and reduced fatigue. The results show that longer periods of time spent driving in a monotonous road environment lead to increased fatigue and alertness declines due to the time spent on task, which is true for both driving periods and the frequency with which big SWM occur. The implications for environmental countermeasures against driver weariness are highlighted [62]. About 20-30% of all traffic accidents in the United States in 2016 were caused by drivers who were too tired to keep their eyes on the road, according to a survey published by the National Highway Traffic Safety Administration. Researchers found that drowsy drivers were more likely to drift out of their lane or instigate a tail chase because of their diminished attentiveness and information processing capacity [63] [64]. As a result, developing a method that accurately identifies signs of driver weariness in order to enhance the safety of vehicular traffic is of utmost importance in the real world.

The method of detecting the driver's drowsiness state by analyzing the driver's facial behavior characteristics has attracted widespread attention of researchers as a result of the rapid development of computer vision technology. This method is used to determine whether or not the driver is fatigued. To identify drowsiness in drivers, researchers extracted a variety of facial tiredness indicators. Different facial fatigue features were extracted using the Deep Belief Network (DBN) and the accuracy of driver sleepiness detection was independently confirmed by Zhao et al. [65] Driver weariness was estimated by Zhang et al. [66] using a rearrangement technique based on local binary patterns (LBPs) [67] and a support vector machine (SVM) [68]. To extract drowsiness-related behavioral variables as face and head movements, Park et al. [69] introduced the Deep Drowsiness Detection (DDD) framework, which makes use of three deep convolutional neural networks (DCNN) [70].

Accidents and injuries that occur as a result of traffic on the roads represent significant challenges for society as a whole as a result of the substantial financial damage that is inflicted on individuals, their families, and entire nations [71]. Nearly 1.25 million people lose their lives each year, according to the World Health Organization (WHO), which reports an average of 3,287 deaths every day [71] [72]. Following a brief introduction to deep learning and its associated algorithms, we will proceed to offer an in-depth analysis of the most recent deep learning-based systems, algorithms, and methodologies for the identification of distraction, fatigue/drowsiness, and aggressiveness in human drivers. Through the presentation of a complete and in-depth comparative examination of all of the most recent detection methods, our goal is to acquire an all-encompassing comprehension of the HIADB detection process [73].

4.3 Literature Review of Existing Models

In this section, we are going to talk about the research papers, current models, related works, its shortcomings, prior works, contributions, and the performance of the existing models of the research papers. Following that, we will have a conversation on the validation accuracy, performance of the existing models, and other related topics. At the end, we are going to discuss the shortcomings of the currently available models as well as provide a summary of the entire chapter as well as the literature study.

Driving will one day be completely risk-free thanks to the advent of highly autonomous vehicles. There are six levels of autonomy for self-driving cars, ranging from 0 to 5, and the technology behind them is established in theory. It has been suggested that advanced driving assistance systems, also known as advanced driving systems (ADS) [74], or advanced driving systems (ADAS), are able to reduce the number of mistakes that drivers make while they are behind the wheel. This would allow for the early detection of potentially hazardous driving situations. Although these systems have numerous benefits to provide and are able to handle some potentially hazardous scenarios, in practice they are unable to recognize potentially hazardous driving dangers. This is despite the fact that they are capable of handling some potentially hazardous circumstances.

This is as a result of the fact that these systems are dependent on sensors, and when subjected to challenging conditions, the accuracy of these sensors drastically decreases [75]. However, when the car is driven in a GPS-deficient area or when the GPS receiver exhibited poor performance, the studies that used accelerometers and gyro-sensors built into smart phones for detecting driving behavior fail to offer accurate results [76] [77] [78] [79] [80]. As a result, it remained challenging to identify aggressive driving.

Moreover, when employed on a mountain route or similar curving road, these systems do not detect aggressive driving behaviors like rapid, erratic turns or frequent braking. The issue with bio-signal based approaches is that they require the use of costly sensors and the installation of those sensors may be unpleasant [81] [82]. Measuring is more challenging with single-camera solutions [83] at night or in tunnels. Dual Near-infrared (NIR) [84] cameras allow for far-ranging detection of driver exhaustion, but they are unable to pick up on outwardly visible signs of drowsiness.

In addition to these studies, the following research articles' literature reviews are listed and will be discussed in more detail below:

∞ **Real-time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks** [85]

In this study, a real-time detection system was built using deep learning. The classification into three classes is carried out at a rate of 14.9 frames per second, with an accuracy of 89.5 percent. The majority of the effort that has been put into developing this model has been concentrated on the following three domains:

- ☐ The manner in which the vehicle is driven.
- ☐ Characteristics of drivers in terms of their psychophysiology
- ☐ Techniques based on Computer Vision [86] for the monitoring of drivers

The model consists of the following two stages: The Multi-Task Cascaded Convolutional Neural Network, or MTCNN, is the first layer that is utilized [87] and DDDN (Driver Drowsiness Detection Network). Here, the main parts are : **Baseline-4:** Observes two eyes, mouth and face, **Baseline-2:** Observes left eye and mouth and **Compressed-2:** Uses a compressed method, named "Distillation". The highest accuracy is achieved while using Baseline-2 (93.8%), while Compressed-2 (89.5%) is used using this model. The previous work done which is described in this paper are given below:

Conventional Approaches for Drowsiness Detection:

It does this by sensing the movement of the steering wheel and the steering wheel itself. This data is then used to determine the driving pattern. In addition to this, it monitors by making use of the information collected by sensors such as EEG, ECG, and EOG. The other method is known as the extraction of facial features using computer vision. The highest level of accuracy it can achieve is 90%.

Detection of Drowsiness by the Use of Deep Learning:

This technique relies heavily on CNN, which represents a significant advancement in the field of computer vision, particularly with regard to tasks such as image classification, object detection, emotion recognition, scene segmentation, and so on. The best accuracy it can get is 78%.

The Deep Learning Compression Algorithms:

In this approach, a massive network referred to as a Teacher network is utilized. This network has demanding compute requirements but has the ability to learn patterns from a substantial dataset. On the other hand, student networks are typically somewhat limited, necessitating a lower level of computational effort, and are able to acquire knowledge via teacher networks.

This study employs three distinct models in order to detect tiredness, but it makes no mention of the low-light environment or the impact that glasses have on accuracy.

∞ **A Real-Time Driving Drowsiness Detection Algorithm with Individual Differences Consideration [88].**

In the first module, a singular tiredness state classifier that was based on SVM was trained by taking Eyes Aspect Ratio as an input. This was done. The amount of drowsy frames per unit of time will be used to calculate a new variable that has been introduced. The model introduced in this research work has a success rate of 94.8 percent. Drowsiness was identified by this model, which was given the moniker DCCNN, mostly by observation of the subject's eyes (Deep Cascaded Convolutional Neural Network). The model also employed SVM (Super Vector Machine). The algorithm is made up of the two modules listed below:

- ☐ Offline Training
- ☐ Online Monitoring

The design is made up of three different subnetworks, each of which has fewer total filters but a higher level of discrimination between those filters. Later in this proposed model, they have employed cardiopulmonary resuscitation (CPR) techniques (Cascaded Pose Regression). Dlib toolbox is utilized in order to acquire Facial Landmarks. Additionally given are the results of comparisons with P80 [PERCLOS], Adaboost, and LSTM. Then eye detection was accomplished through the use of an elliptical fitting approach.

The possibility of poor lighting was not discussed in this article either. This model also has issues with the head posture and movement during the animation. In order to further improve the performance of the algorithm, we will investigate multi-feature methods such as fusing mouth and head posture. Because it is easier to become sleepy when driving at night, the research on detecting tiredness while driving will be carried out at night.

- **A Real-Time System for Monitoring Driver Fatigue [89]**

A brand new eye-detection algorithm that integrates Adaboost and algorithm with template matching has been proposed. They have presented a revised version of the Camshift algorithm. Then they utilized Haar-like traits in order to identify the faces of drivers. Additionally this model performs well in dim light. Utilized the PERCLOS P80 metric in order to carry out the fatigue evaluation. It has a 92.65% degree of accuracy.

The proposed methodology combines Camshift and Adaboost algorithms to produce Haar-like features. In the event that the tracking algorithm is turned off, the Adaboost algorithm will be restarted, and the tracking algorithm will continue to operate until the face image is redirected. Provides a way for matching templates as well as an eye-validation technique. In order to determine the fatigue features, the Lin Algorithm was applied to each iris, and then two ellipses were fitted to each of the iris's contours. For the purpose of computation, we used the P80 metric of PERCLOS, which indicates that the height/width ratio of eye closure is approximately less than 0.2. This model is also experimented under Work done while exposed to natural light. In the hue (H) channel of the HSV color space, the integration projection and elliptical fitting methods are performed in order to extract fatigue features from the eyes of the drivers.

This model offers a respectable accuracy of 92.6 percent. This device, in contrast to others, is functional even when there is little light. However, there has been no mention made regarding whether or not the driver is wearing glass.

∞ **Real-Time Driver-Drowsiness Detection System Using Facial Features[90]**

Proposed a system called DriCare [90]. The model also detects drivers' tiredness status. A brand new algorithm for tracking faces is also shown. A new approach of facial recognition detection based on 68 critical points has been developed. This results in an accuracy rate of 92%.

In this method, a non-contact procedure is used, which results in a lower cost than setting up EEG, ECG, or other equipment. In order to accomplish the objective, there are three significant obstacles to overcome:

- ☐ There is a wide range of heights among drivers.
- ☐ The angles at which their faces are filmed in the video are not the same.
- ☐ Changing positions.

The KCF (Kernelized Correlation Filters) [91] algorithm is also altered in this research for the following reasons:

- ☐ KCF is not a reliable method for face tracking.
- ☐ After leaving the detection area, it will no longer be able to recognize faces.

MC-KCF, also known as Multiple Convolutional Neural Networks-KCF, is the name of the new method that has been introduced. It combines CNN and KCF into one channel. This procedure also finds a way to compensate for the fact that the KCF algorithm is unable to mark the target in the first frame in order to avoid losing it.

It presents the newest version of the KCF protocol. When calculating weariness, both the mouth and the eye are taken into account in this proposed methodology. A lower cost alternative to the contact procedures.

- **Real-Time Driver Fatigue Detection System Based on Multi-Task ConNN [92]**

Presented a model of a Convolutional Neural Network that can perform multiple tasks in this research work. The information obtained from the tongue and the eyes is categorized simultaneously. The PERCLOS test is used to assess how quickly the eyes close, whereas the FOM test measures how often someone yawns. This model achieves an accuracy of 98.81% when applied to the YawdDD [93] and NthuDD [94] datasets.

This study is an expanded version of the work that they have done in the past. The researchers have used the Dlib portion of CPP in this methodology.After that they have calculated based on the face using 68 critical points. The output is displayed using three different degrees:

- ☐ Extremely weary
- ☐ Less tired
- ☐ Not tired

The head condition will be incorporated into an embedded system alongside the eye and mouth conditions. In addition, the limitations of light were not discussed in this article. They did not conduct any tests in real-world conditions.

- **Research on Fatigue Driving Feature Detection Algorithms of Drivers Based on Machine Learning. [13]**

In an effort to identify instances of driver weariness while they are behind the wheel. There is a proposal for a diagnostic model that is based on machine learning in this research domain. It is able to adequately meet the conditions of drivers who are fatigued. In this study, we investigate the use of machine learning for the detection of fatigued driving. Investigates the possibility of a driver fatigue detection system being implemented in the context of extended periods of driving. This model provides a solution to the issue of driver weariness by eliminating the need for sensory detection. In order to get higher levels of accuracy, the cascade approach of SVM is also utilized here. In this study, the fatigue aspects that were adopted have not been finished. Due to the utilization of open-source data sets, the scale of the data sets that were utilized in this paper is insufficient.

It is planned to build LTSM in addition to other long-term memory neural networks that use Deep Learning. Because there is a shortage of data, the procedures have not been tested appropriately.

4.4 Conclusion

This chapter is dedicated to a discussion of the research papers that both you and I have recently read. New conceptual frameworks and methodological approaches, as put up by the researchers, have been brought to our attention. The most recent publications have provided us with insight into the proposed models' shortcomings, as well as future work and gaps in those models' coverage.

Chapter 5

Proposed Methodology and Implementation

5.1 Introduction

In this chapter, we covered a number of significant aspects in our methodology, such as the preprocessing of data, the construction of a CNN, and other topics. In addition to that, we presented a fundamental procedure of our whole methodology. At the end of this chapter, we went over the many assessment criteria that we took into consideration when conducting research.

5.2 Workflow Essentials

Having a fundamental workflow allows us to acquire a better understanding of the primary actions involved in the entire procedure. Throughout the entirety of our job, we followed a somewhat consistent workflow. The following diagram illustrates our primary method of operation.

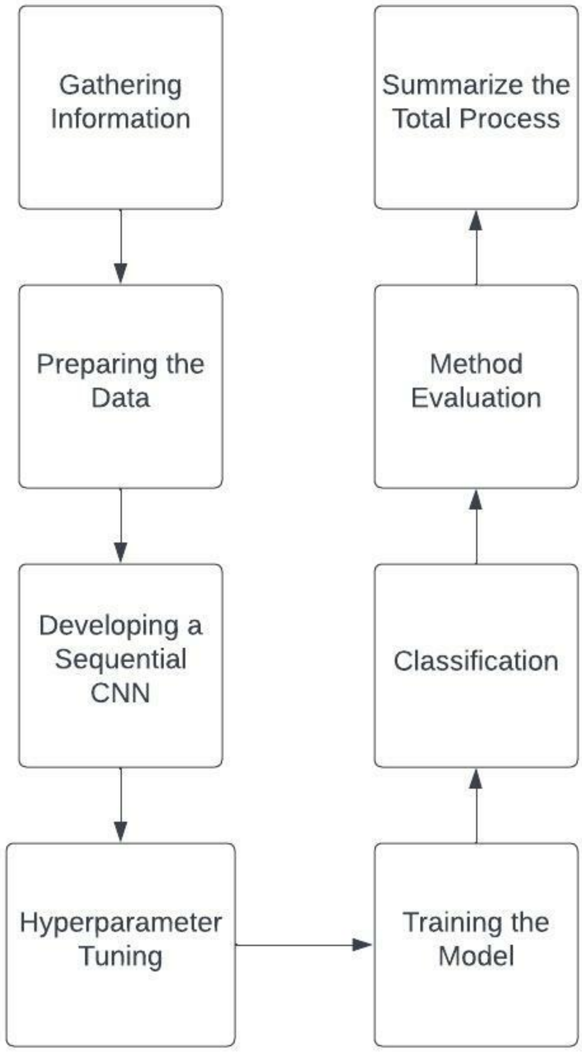


Figure 5.1: The Basic Workflow of our Methodology

You can see that the first step in our research process was gathering information. After that, we performed some preliminary processing on the information that had been gathered in order to prepare it for input into the convolutional neural network (CNN). Following that, we proceeded to design the fundamental framework of our multi-stage CNN model. The next step was for us to perform some fine-tuning on our models by adjusting a number of hyperparameters, such as the batch size, the number of epochs, the number of channels, the filter height, the learning rate, and the dropout probability. After making adjustments to the parameters, we trained our models using the optimum value in the dataset that they were given to work with. After that, we utilized the trained models to make predictions based on samples from the testing datasets. After that, we used the various indicators that we had chosen to evaluate performance to assess how well our models performed (Sensitivity, Specificity, Accuracy, MCC, AUC). In order to have a better knowledge about the capabilities of our model, we compared our outcome to the predictors that were already in place. In order to provide a clearer understanding of the functionality of our model, we have included all of the relevant graphical illustrations in each of these processes.

5.3 Data Preprocessing

Applications of machine learning, particularly in the field of deep learning, continue to become more diverse and numerous at an accelerated rate. The data-centric approaches to model building, such as data augmentation techniques, might be a useful instrument in the fight against the issues that are now being faced in the field of artificial intelligence. Data augmentation is beneficial for improving the performance and results of machine learning models by producing new and varied examples to train datasets. This can be accomplished through the formation of new and diverse examples. If the dataset used in a machine learning model is extensive and meets the requirements, the model will perform better and more correctly. Data collection and labeling can be laborious and expensive operations, but they are essential steps in the development of machine learning models. The reduction of these operational costs is made possible for businesses by the transformations that can be made to datasets through the application of data augmentation techniques. The process of cleaning data is one of the phases involved in developing a data model and is required for models to have a high level of accuracy. If, on the other hand, cleaning the data results in a reduction in the data's representability, the model will be unable to make accurate predictions for the inputs that come from the real world. Data augmentation approaches can make machine learning models more robust by simulating variables the models would encounter in the actual world. This makes the models more applicable to the world in which they are intended to be used. Only these three methods for augmenting the data have been utilized by us here; they are as follows:

- Re-scaling
- Zooming
- Rotation
- Horizontal Flip

Project ”Verified Human Interfaces, Control, and Learning for Semi-Autonomous Systems” [95] is the source of the dataset used in this study. This is a component of the study titled ”Using a Driver's Eye Data to Predict Accident-Causing Drowsiness Levels,” and it was conducted by the National Highway Traffic Safety Administration [96].

5.4 Structure of CNN

Following the completion of any necessary preprocessing, the data was sent on to be processed by a convolutional neural network. In a typical CNN model, the inputs are linked to certain convolution and max-pooling layers, which are then followed by a few fully connected layers that are linked to the output layer. In contrast, the inputs that have been preprocessed are sent to a CNN model that has several channels. The goal of this strategy was to ensure that a sequence gets handled in chunks of varying sizes. It is possible that the best features are not always extracted using the same filter size for each convolution in a sequential model. For this reason, we merged the features obtained by performing feature extraction operations (convolution and max-pooling) across several channels on the input sequence. We made adjustments to the convolution layer's height of the filters and the number of channels by using the notation n and f respectively. The breadth of the filters stayed the same throughout. After that, each of these convolution layers was wired up to a max-pooling layer. Where N is the length of the nucleotide, and f is the height of the filters, the filter size for each of the max-pooling layers was calculated as (N-f+1). After that, the concatenation of the max-pooling layers was done in order to merge the features that were extracted by the convolution and maxpooling layers. The following step involves connecting the concatenated max-pooling layers to the first completely linked layer. Following that, we utilized dropout regularization in an effort to lessen the total number of factors. After that, the last layer was connected, which resulted in a probability distribution of the classes. It was possible to make an educated guess about the ultimate outcome by using the probability distribution.

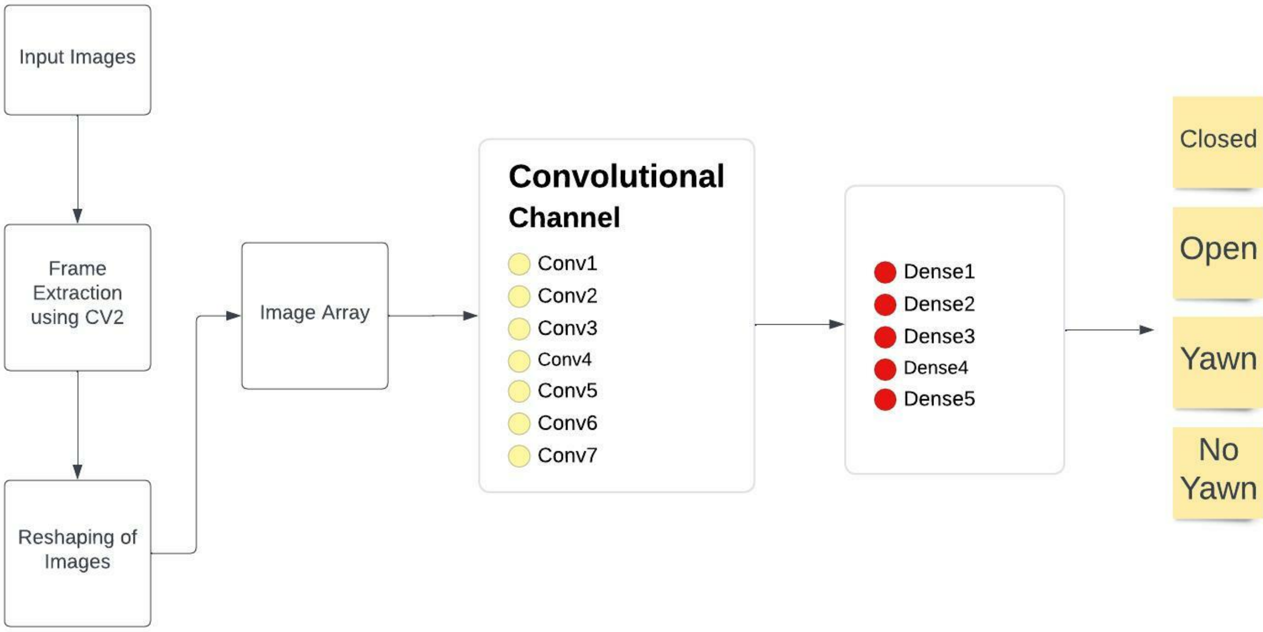


Figure 5.2: Proposed Methodology

The number of convolution layers was decided upon through the use of k-fold cross-validation in conjunction with grid search. The learning rate, dropout frequency, and height of the filters were all determined with the assistance of cross-validation as well. In every layer besides the final one, which used the softmax activation function instead of the relu activation function, relu was used as the activation function. The framework of our model looked roughly like this overall. When looking at different datasets, the only thing that changed was the height of the filters and the number of convolution layers.

A loss function is a way to evaluate how well a model performs in a dataset. This evaluation is done by comparing the model's performance to the original data. As the loss function for our model, we decided to use cross-entropy, which compares the predicted labels to the actual labels. The following is a definition of cross-entropy:

$$Cross - entropyloss = \sum_{i=0}^{n} Y_{p_i} log Y_{a_i}$$

In this case, n is the total number of classes, Ya denotes the actual probability of belonging to class I and Yp denotes the expected probability of belonging to class i. By selecting the appropriate weights and biases, our objective was to get the loss function down to its minimum possible value. On the training data, we utilized back-propagation, which changed the parameters based on the cross-entropy of the data.

5.5 Hyperparameter Tuning

In order to maximize a model's prediction performance, hyperparameter tuning is an absolutely necessary step [97] [98]. In neural networks, tweaking the hyperparameters is an extremely important step that must be taken in order to reach global minima and prevent either overfitting or underfitting. We fine-tuned our model by adjusting a number of hyperparameters by working with the benchmark datasets.

Hyperparameters	Ranges of Values	Selected Values
No. of epochs	[10, 30, 40, 50, 70, 80, 100]	80
No. of channels	[5, 7, 9, 11]	7
Learning rate	[0.0001, 0.0003, 0.0005, 0.0007, 0.001]	0.0005
Dropout Probability	[0.4, 0.45, 0.5, 0.55, 0.60]	0.40

Table 5.1: The ranges and the selected values of hyperparameters of the selected dataset.

With the hyperparameters, we were able to obtain the complete model structure. Let's have a look at a summary of the final structure after the hyperparameters have been optimized before we get into a discussion about the training and testing procedures that we used in our technique. The overall number of trainable parameters was 1,592,700, and the vast majority of those parameters were implemented in the dense layer. The total number of trainable parameters was 1,592,828.

5.6 Metrics for the Method of Evaluation

There are many different assessment measures for convolutional neural networks. We needed four parameters to calculate them: true positive (TP), true negative (TN), false positive (FT), and false negative (FN) (FN). The following is a list of the equations for the evaluation metrics:

5.6.1 Accuracy (ACC)

The extent to which the predictions made on the test data are right is what is meant by ”accuracy”.

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

5.6.2 Precision

One measure of the efficacy of a machine learning model is its predictive accuracy, or precision. In statistics, accuracy is measured by the proportion of correct predictions made to the total number of forecasts that were positive (i.e., the number of true positives plus the number of false positives) [99].

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

5.6.3 Recall

The recall is determined by determining the ratio of the total number of positive samples to the number of positive samples that were correctly identified as positive. This ratio is then multiplied by 100. The model's capacity to recognize positive samples is evaluated using the recall metric. The greater the recall, the greater the number of positive samples found [100].

$$Recall = \frac{TP}{TP + FN}$$

5.6.4 F-1 Score

The F1-score is widely regarded as one of the most essential measures for performance assessment in machine learning. It does an excellent job of summing up the predictive performance of a model by integrating two metrics, recall and precision, that would otherwise compete with one another [101].

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

5.7 Conclusion

This chapter lays out the framework for how we went about doing this task. A process flowchart was also included. We also described how to acquire data, how to preprocess it, the structure of the model, and how to evaluate the effectiveness of the strategy.

In addition, the proposed model has been presented in this chapter. Within this model, the output has been separated into four categories: yawn, no-yawn, open, and closed.

Chapter 6

Analysis of Results and Performance

6.1 Introduction

We analyzed our methods' performance and presented our findings in this chapter. The procedure began with hyperparameter training, then progressed to training and testing. A graphical and tabular presentation of our findings is included at the chapter's conclusion.

6.2 Experimental Setup

Kaggle has been used throughout the entirety of the process to build it. Python 3 is utilized throughout the implementation process. The following constitute the whole hardware configuration:

6.2.1 CPU

Component	CPU-only	CPU (when paired with GPU)
Architecture:	x86_64	x86_6
Model:	79	85
Model name:	Intel(R) Xeon(R) CPU @ 2.20GHz	Intel(R) Xeon(R) CPU @ 2.20GHz
CPU MHz:	2199.998	2000.190

Table 6.1: The CPU used when no GPU is selected, as well as the CPU used when a GPU is selected

6.2.2 GPU

The graphics processing unit (GPU) in question is an NVIDIA Tesla P100, and it has 16 gigabytes of RAM. When researching the Tesla P100, I found that this GPU has a total of 3584 Cuda cores.

6.3 Prediction Made Using the Model That Was Proposed

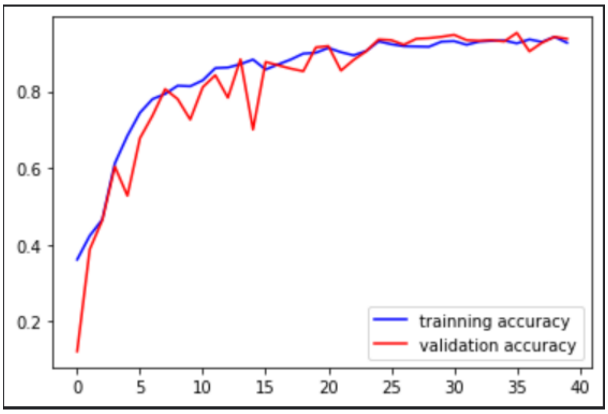
When adequate hyperparameter tuning is performed, as was covered in the chapter before this one, the suggested model performs at its highest level of capability. In this section, we are going to look at a graph that compares the validation accuracy, the training accuracy, the validation loss, and the training loss according to the various epochs.

Validation Accuracy, Training Accuracy, Validation Loss and Training Loss after 40 epochs:

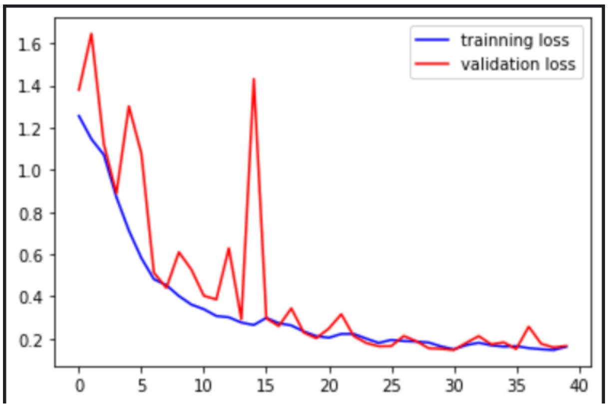
Table 6.2: Evaluation Metric after 40 epochs

Metric / Class	Precision	Recall	f1 Score	Support
Yawn	0.55	0.90	0.68	63
No_Yawn	0.82	0.45	0.58	74
Closed	0.96	0.71	0.82	215
Opened	0.86	0.96	0.87	225

The values of the evaluation metrics termed precision, f1-score, recall, and support are presented in the table that is displayed above. These evaluation metrics are employed in this suggested model to evaluate the performances of the model. The validation accuracy graph, the training accuracy graph, the validation loss graph, and the training loss graph are all displayed in this experiment. The graphical results of using the hyperparameter for 40 iterations can be seen in the first part of the photographs here. After forty epochs, we are able to observe a greater curve during the fifteenth epoch of validation accuracy, and this higher curve shows that the detection did not work. It is possible for it to occur with the noises in the dataset. However, around the fifteenth epoch, the model begins to perform significantly better than the early stages. As a result, the forty-epoch timescale is an excellent candidate for the role of hyperparameter.



(a) Accuracy Graph



(b) Loss Graph

Figure 6.1: Accuracy and Loss Graph

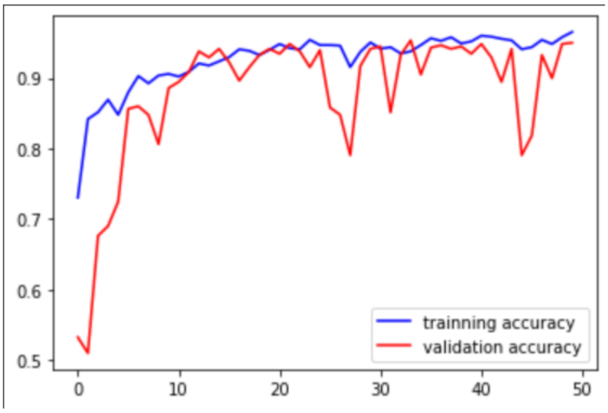
The similar issue can be seen in the Loss graph, which reveals a higher arch in the validation loss curve. This is an indication of the problem. Because the issue arises during the process of validating the accuracy of the model, the very same component of the model also causes the greater validation loss, which is made abundantly evident by the graphical representation. In this situation, the issue that we are having may be resolved by either retuning all of the hyperparameters or increasing the epoch in order to train the model and stabilize the graphs. Both of these options are available to us. On the other hand, this could lead to the issue of the garment being too tight.

Validation Accuracy, Training Accuracy, Validation Loss and Training Loss after 50 epochs:

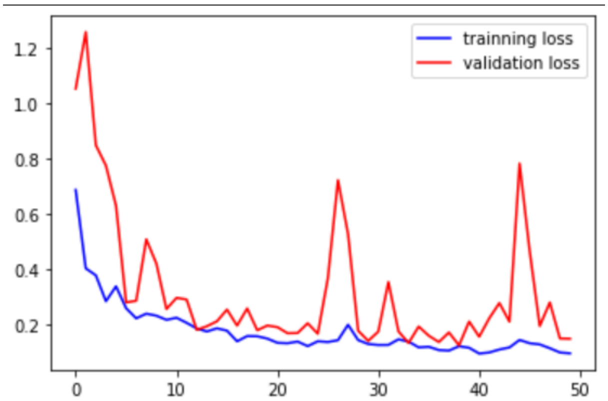
Table 6.3: Evaluation Metric after 50 epochs

Metric / Class	Precision	Recall	f1 Score	Support
Yawn	0.83	0.32	0.46	63
No_Yawn	0.39	0.45	0.56	74
Closed	0.85	0.71	0.74	215
Opened	0.92	0.85	0.88	225

In this particular scenario, we are going to evaluate the performance of the model after a total of fifty epochs have passed. The values of the evaluation measures that were utilized in the phase before this one may be seen in table 1.3. After that, we are able to view the graphical representation of the validation accuracy, validation loss, training accuracy, and training loss of the proposed model in the event that fifty epochs are used.



(a) Accuracy Graph



(b) Loss Graph

Figure 6.2: Accuracy and Loss Graph

The graphical representation of this instance gives us two larger arches in the case of validation accuracy. This indicates that the accuracy reduces just before the 30th epochs, and another one is just between the 40th and 50th epochs, more particularly around the 45th epoch. Due to the fact that the accuracy decreases the most in those two areas, the loss will also be greater in those cases. It is also possible to verify if we look at the graph indicating the amount of loss. In order to fix the issue at hand, we have applied a batch normalization layer in this location in order to standardize the values of the output matrix. In light of the fact that the hyperparameter is not performing particularly well, we can still take into consideration the possibility that more epochs will perform better than fifty epochs.

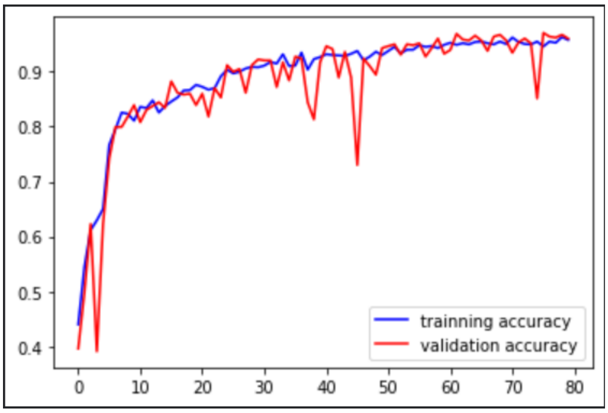
Validation Accuracy, Training Accuracy, Validation Loss and Training Loss after 80 epochs:

Table 6.4: Evaluation Metric after 80 epochs

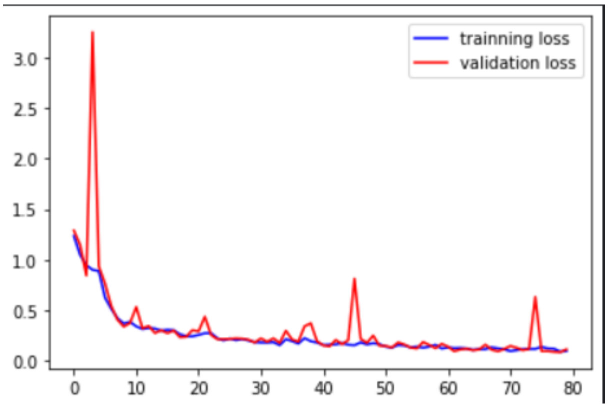
Metric / Class	Precision	Recall	f1 Score	Support
Yawn	0.46	0.71	0.56	63
No_Yawn	0.53	0.88	0.66	74
Closed	0.99	0.52	0.68	215
Opened	0.90	0.99	0.94	225

We have executed the proposed model, and it has provided a graphical depiction of how well the model performed. This is identical to what we did in the previous step. Additionally, the various values of each of the evaluation metrics that are employed in this implementation are provided by this implementation as well. These graphs also depict the degrees of accuracy and loss exhibited by the proposed model during the validation and training processes respectively. In this particular instance, the hyperparamter values, such as batch size and learning rate, were kept the same as they had been in the model for the 40th epoch.

After being implemented with the hyperparameter of eighty epochs, the model's performance is significantly improved over that of its predecessor. In this case, we can see a medium arch during the fifty-first epoch, and we can also see a lengthy arch right after the beginning of the implementation, which is fairly common. Both of these arcs are quite typical. The larger the decline in validation accuracy, the greater the corresponding loss in validation. Furthermore, the arch that is produced during the fifty-first epoch contributes a medium arch to the validation loss. However, after them, the graph reached a very stable state. Therefore, we are able to consider the hyperparameter of eighty epochs to be a superior tuning than the one that was used previously.



(a) Accuracy Graph



(b) Loss Graph

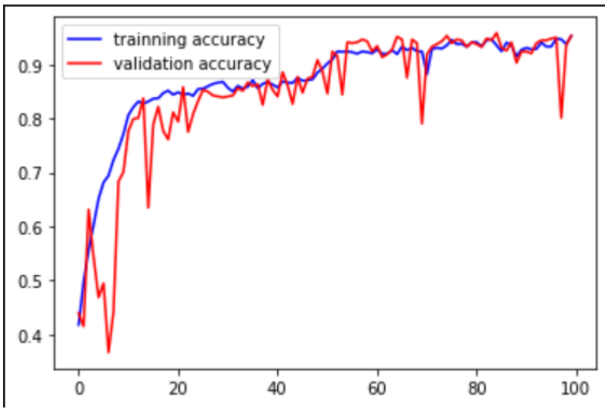
Figure 6.3: Accuracy and Loss Graph

Validation Accuracy, Training Accuracy, Validation Loss and Training Loss after 100 epochs:

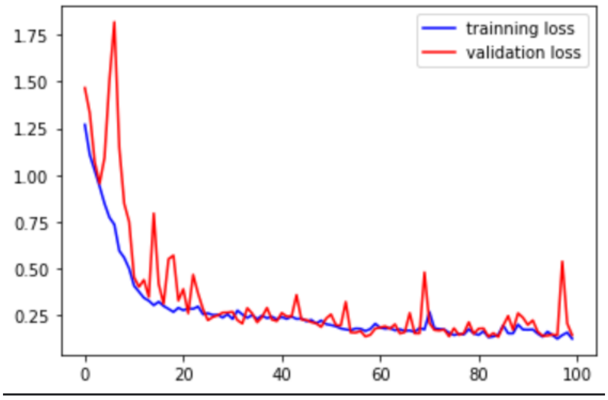
Table 6.5: Evaluation Metric after 100 epochs

Metric / Class	Precision	Recall	f1 Score	Support
Yawn	0.36	0.86	0.50	63
No_Yawn	0.66	0.31	0.42	74
Closed	0.94	0.62	0.75	215
Opened	0.86	0.94	0.90	225

Finally, the hyperparameter of one hundred epochs that was used at the very beginning of the chapter is going to be presented in the final section of the document. After the implementation, it is possible to observe that this hyperparameter is unable to provide superior performance to that of lower epochs. The performance of the graph that was presented after eighty epochs was significantly better than that of the graph that was presented after one hundred epochs. Therefore, it is abundantly evident that we can get rid of this hyperparameter.



(a) Accuracy Graph



(b) Loss Graph

Figure 6.4: Accuracy and Loss Graph

6.4 Performance Analysis

In the course of this investigation, we have developed a multi-class model that divides the results of our analysis into four distinct categories: yawn, no yawn, open, and closed. However, in the research that came before, we discovered that all models could be solved by employing a binary classification model. In light of this, the output format of the model is where the suggested model makes its most significant contribution. The Sequential model uses seven convolutional layer channels and five dense layer channels, and each convolutional layer includes an attached maxpooling layer. The model is described as having seven convolutional layer channels. Batch

Normalization was also implemented in the model before the activation layer before it was finally finalized. After forty iterations, this proposed model continues to operate in a consistent manner. The model has an accuracy rate of 95.34% on average, which is considered to be quite satisfactory.

6.5 Conclusion

In this chapter, we have evaluated the performance of the algorithm after making adjustments to the hyperparameters using a variety of methods. In addition to this, we have analyzed the performance of a variety of assessment criteria, and the graph below demonstrates the validation accuracy, validation loss, training accuracy, and training loss respectively. In the following chapter, we are going to draw a conclusion to the entire thesis work, as well as define the future scope of the recommended approach that has been detailed in this study effort.

Chapter 7

Conclusions and Suggestions for Further Research

7.1 Introduction

The scope of the topic, prior studies, our contribution, experimental analysis, and findings are all briefly described in this chapter. Additionally, we have included a brief study of the potential extensions of our study.

7.2 Summary

The purpose of this line of research was to develop a multi-class model that was based on deep learning so that it could detect tiredness in drivers, which is one of the most common factors that contributes to vehicular mishaps. In light of this fact, we went ahead and altered and pre-processed the input data in order to get them ready for the application of a deep learning model. Instead of using the binary class classification model that is presented in the majority of journal papers, we decided to use a multi-class technique that was based on deep learning. We constructed the multi-stage CNN model by tuning the hyperparameters in the manner that is most likely to be successful. Our goal in deploying a CNN with several stages was to ensure that our model could successfully process a variety of datasets. This goal was accomplished by ensuring that the model could handle multiple datasets. For this reason, we utilized convolution filters of varying sizes across each of the stages. The concatenation of the output of the max-pooling layers gave us a wide variety of features to use in determining the level of fatigue exhibited by a driver. After that, we fine-tuned our model by adjusting a number of hyperparameters, which resulted in an improved model's predictive performance. Following the tuning of the parameters, we trained our models on the datasets and then examined how well they performed using independent datasets. The results of the investigation indicate that the model works well with the dataset that was applied to it.

7.3 Future Scopes

We are confident that the findings of our research will make a substantial contribution to the advancement of technologies that can detect tiredness in drivers and lead to a reduction in the number of accidents that occur on the roads. On the other hand, we are aware that there is a significant amount of room for more research in this field. As time passes, there is an increasing amount of involvement from the scientific community in the study of data pertaining to automobiles. In the course of our investigation, we have looked into a relatively limited dataset. Despite this, there are now more datasets available about the examination of automobiles that can drive themselves. Additionally, in the not-too-distant future, there will be the introduction of additional datasets. There is a possibility that these datasets include a great deal more information than the ones that are currently available. When analyzing datasets with a higher level of complexity, it is possible to discover even more outstanding performance for drowsiness identification in drivers. In addition, the scientific community is moving toward the next generation of driver drowsiness detection as a next-generation technology provides more accurate results due to the small loss in information compared to automobile technology. This is because a next-generation technology can detect drowsiness in drivers more accurately. This is also something that has the potential to be a substantial area of interest. Validating the pathway through real-time detection will be the primary focus of work that needs to be done in the future regarding the identification of driver drowsiness. We anticipate that in the not-too-distant future, this field will see an increase in the number of researchers working in it, as well as an increase in the number of technological innovations made.

REFERENCES

[1] B. C. Tefft, "Acute sleep deprivation and risk of motor vehicle crash involvement," 2016.

[2] X. Li, X. Lian, and F. Liu, "Rear-end road crash characteristics analysis based on chinese in-depth crash study data," pp. 1536–1545, 07 2016.

[3] CDC, "Drowsy driving can happen to anyone," *Centers for Disease Control and Prevention*, Sep 2022.

[4] "Drowsy driving: Dangers and how to avoid it | sleep foundation 2022," *Sleep Foundation*, Feb 2022.

[5] K. v Sonka and J. Volna, "Medical factors of falling asleep behind the wheel," p. 566–569, 2005.

[6] J. Bass and J. S. Takahashi, "Circadian integration of metabolism and energetics," *Science*, vol. 330, p. 1349–1354, Dec 2010.

[7] V. V. Gafarov, E. A. Gromova, V. N. Maksimov, I. V. Gagulin, and A. V. Gafarova, "Biological determinants of sleep disorders," *Circadian Rhythm - New Insights Into the Physiological and Pathological Implications [Working Title]*, Feb 2022.

[8] "Drowsy driving - sleep disorders | ucla health," *Drowsy Driving - Sleep Disorders | UCLA Health*.

[9] D. S. Goldstein, "Differential responses of components of the autonomic nervous system," *Handbook of clinical neurology*, vol. 117, pp. 13–22, 2013.

[10] C. Tsigos and G. P. Chrousos, "Hypothalamic–pituitary–adrenal axis, neuroendocrine factors and stress," *Journal of psychosomatic research*, vol. 53, no. 4, pp. 865–871, 2002.

[11] L. A. Matura, S. Malone, R. Jaime-Lara, and B. Riegel, "A systematic review of biological mechanisms of fatigue in chronic illness," *Biological research for nursing*, vol. 20, no. 4, pp. 410–421, 2018.

[12] "Drowsy driving prevention week® - national sleep foundation 2022," *National Sleep Foundation*, Oct 2022.

[13] H. Zhongwei, O. Shuangjiang, and X. Dengyuan, "Research on fatigue driving feature detection algorithms of drivers based on machine learning," *Systems Science & Control Engineering*, vol. 9, no. 1, pp. 167–172, 2021.

[14] T. S. F. (2011)., "National highway traffic safety administration," 2016.

[15] "Accidents caused by truck driver fatigue - the king law firm 2021," *The King Law Firm*, Jan 2021.

[16] A. F. for Traffic Safety, "2019 traffic safety culture index," 2019.

[17] Q. Wang, J. Yang, M. Ren, and Y.-J. Zheng, "Driver fatigue detection: A survey," *2006 6th World Congress on Intelligent Control and Automation*, vol. 2, pp. 8587–8591, 2006.

[18] "Drowsy driving - sleep disorders | ucla health," *Drowsy Driving - Sleep Disorders | UCLA Health*.

[19] J. Damashek, "The consequences of drowsy driving | hecht, kleeger damashek, p.c.," *Hecht, Kleeger Damashek, P.C.*, Mar 2017.

[20] A. Mukminin, "9 safe driving tips to protect you and your family," *DSF.my*, Jul 2021.

[21] S. Vishwanathan and M. N. Murty, "Ssvm: a simple svm algorithm," vol. 3, pp. 2393–2398, 2002.

[22] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," *arXiv preprint arXiv:1612.02295*, 2016.

[23] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," pp. 11976–11986, 2022.

[24] S. Keren, O. Gheysens, C. S. Levin, and S. S. Gambhir, "A comparison between a time domain and continuous wave small animal optical imaging system," *IEEE transactions on medical imaging*, vol. 27, no. 1, pp. 58–63, 2007.

[25] M. Dundar and D. Landgrebe, "Toward an optimal supervised classifier for the analysis of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 1, pp. 271–277, 2004.

[26] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," pp. 253–256, 2010.

[27] T. Hey and A. Trefethen, "The data deluge: An e-science perspective," *Grid computing: Making the global infrastructure a reality*, vol. 72, pp. 809–824, 2003.

[28] X. Kang, B. Song, and F. Sun, "A deep similarity metric method based on incomplete data for traffic anomaly detection in iot," *Applied Sciences*, vol. 9, p. 135, 01 2019.

[29] L.-X. Wang and F. Wan, "Structured neural networks for constrained model predictive control," *Automatica*, vol. 37, no. 8, pp. 1235–1243, 2001.

[30] H. Huang, S. Liu, and K. Davis, "Energy theft detection via artificial neural networks," pp. 1–6, 10 2018.

[31] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 335–346, 2020.

[32] R. Hecht-Nielsen, "Theory of the backpropagation neural network," pp. 65–93, 1992.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[34] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[35] D.-X. Zhou, "Theory of deep convolutional neural networks: Downsampling," *Neural Networks*, vol. 124, pp. 319–327, 2020.

[36] J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu, "A theory of generative convnet," pp. 2635–2644, 2016.

[37] K. Thayer, S. E. Chasins, and A. J. Ko, "A theory of robust api knowledge," *ACM Transactions on Computing Education (TOCE)*, vol. 21, no. 1, pp. 1–32, 2021.

[38] "A friendly introduction to [deep] neural networks | knime," *KNIME*.

[39] D. Bisla, A. N. Saridena, and A. Choromanska, "A theoretical-empirical approach to estimating sample complexity of dnns," pp. 3270–3280, 2021.

[40] Y.-C. Hsieh, C.-L. Chin, C.-S. Wei, I.-M. Chen, P.-Y. Yeh, and R.-J. Tseng, "Combining vgg16, mask r-cnn and inception v3 to identify the benign and malignant of breast microcalcification clusters," in *2020 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, pp. 1–4, IEEE, 2020.

[41] P. V. LAL, U. SRILAKSHMI, and D. VENKATESWARLU, "Face recognition using deep learning xception cnn method," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 2, 2022.

[42] "What is a convolutional neural network?," *What is a Convolutional Neural Network? - MATLAB Simulink*.

[43] T. Xiao, L. Liu, K. Li, W. Qin, S. Yu, and Z. Li, "Comparison of transferred deep neural networks in ultrasonic breast masses discrimination," *BioMed research international*, vol. 2018, 2018.

[44] S. Yu, K. Wickstrøm, R. Jenssen, and J. C. Príncipe, "Understanding convolutional neural networks with information theory: An initial exploration," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 435–442, 2020.

[45] H. J. Jie and P. Wanda, "Runpool: A dynamic pooling layer for convolution neural network.," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 66–76, 2020.

[46] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2020.

[47] M. Rahaman, M. Mahin, M. Ali, and Hasanuzzaman, "Bhcdr: Real-time bangla handwritten characters and digits recognition using adopted convolutional neural network," 04 2019.

[48] U. Udofia, "Basic overview of convolutional neural network (cnn)," *Retrieved May*, vol. 27, p. 2019, 2018.

[49] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes, “Deeper networks for pavement crack detection,” 07 2017.

[50] T. Wiatowski and H. Bölcskei, “A mathematical theory of deep convolutional neural networks for feature extraction,” *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1845–1866, 2017.

[51] A. Agapitos, M. O’Neill, M. Nicolau, D. Fagan, A. Kattan, A. Brabazon, and K. Curran, “Deep evolution of image representations for handwritten digit recognition,” pp. 2452–2459, 05 2015.

[52] “Activation functions in neural networks [12 types use cases],” *Activation Functions in Neural Networks [12 Types Use Cases]*.

[53] “Sigmoid function - wikipedia 2022,” *Sigmoid function - Wikipedia*, Aug 2022.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[55] X. Lai, L. Zhou, Z. Fu, S. M. Naqvi, and J. Chambers, “Enhanced pooling method for convolutional neural networks based on optimal search theory,” *IET Image Processing*, vol. 13, no. 12, pp. 2152–2161, 2019.

[56] J. Koutník, J. Schmidhuber, and F. Gomez, “Online evolution of deep convolutional network for vision-based reinforcement learning,” pp. 260–269, 07 2014.

[57] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, “The history began from alexnet: A comprehensive survey on deep learning approaches,” *arXiv preprint arXiv:1803.01164*, 2018.

[58] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” pp. 1–9, 2015.

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

[60] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” pp. 770–778, 2016.

[61] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards, and M. Golz, "Steering wheel behavior based estimation of fatigue," vol. 5, no. 2009, 2009.

[62] P. Thiffault and J. Bergeron, "Monotony of road environment and driver fatigue: a simulator study," *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 381–391, 2003.

[63] J. He, W. Choi, Y. Yang, J. Lu, X. Wu, and K. Peng, "Detection of driver drowsiness using wearable devices: A feasibility study of the proximity sensor," *Applied ergonomics*, vol. 65, pp. 473–480, 2017.

[64] Z. Jie, M. Mahmoud, Q. Stafford-Fraser, P. Robinson, E. Dias, and L. Skrypchuk, "Analysis of yawning behaviour in spontaneous expressions of drowsy drivers," pp. 571–576, 2018.

[65] L. Zhao, Z. Wang, X. Wang, and Q. Liu, "Driver drowsiness detection using facial dynamic fusion information and a dbn," *IET Intelligent Transport Systems*, vol. 12, no. 2, pp. 127–133, 2018.

[66] Y. Zhang and C. Hua, "Driver fatigue recognition based on facial expression analysis using local binary patterns," *Optik*, vol. 126, no. 23, pp. 4501–4505, 2015.

[67] A. Hadid, "The local binary pattern approach and its applications to face analysis," pp. 1–9, 2008.

[68] Q. Wen, Z. Yang, Y. Song, and P. Jia, "Automatic stock decision support system based on box theory and svm algorithm," *Expert systems with Applications*, vol. 37, no. 2, pp. 1015–1022, 2010.

[69] S. Park, F. Pan, S. Kang, and C. D. Yoo, "Driver drowsiness detection system based on feature representation learning using various deep networks," pp. 154–164, 2016.

[70] A. Ren, Z. Li, C. Ding, Q. Qiu, Y. Wang, J. Li, X. Qian, and B. Yuan, "Sc-dcnn: Highly-scalable deep convolutional neural network using stochastic computing," *ACM SIGPLAN Notices*, vol. 52, no. 4, pp. 405–418, 2017.

[71] "Road safety facts mdash; association for safe international road travel 2022," *Association for Safe International Road Travel*, Apr 2022.

[72] "Global status report on road safety 2018," *Global status report on road safety 2018*, Jun 2018.

[73] W. Chen, B. Wan, and G. Sun, "Safety analysis and research on risk assessment model of spacecraft assembly," pp. 454–459, 2011.

[74] N. Aranjuelo, L. Unzueta, I. Arganda-Carreras, and O. Otaegui, "Multimodal deep learning for advanced driving systems," pp. 95–105, 2018.

[75] S. Monjezi Kouchak and A. Gaffar, "Estimating the driver status using long short term memory," pp. 67–77, 2019.

[76] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li, "D 3: Abnormal driving behaviors detection and identification using smartphone sensors," pp. 524–532, 2015.

[77] T.-H. Nguyen, D.-N. Lu, D.-N. Nguyen, and H.-N. Nguyen, "Dynamic basic activity sequence matching method in abnormal driving pattern detection using smartphone sensors," *Electronics*, vol. 9, no. 2, p. 217, 2020.

[78] D.-W. Koh and H.-B. Kang, "Smartphone-based modeling and detection of aggressiveness reactions in senior drivers," pp. 12–17, 2015.

[79] R. Chhabra, S. Verma, and C. Rama Krishna, "Detecting aggressive driving behavior using mobile smartphone," pp. 513–521, 2019.

[80] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402–427, 2012.

[81] C. D. Katsis, N. Katertsidis, G. Ganiatsas, and D. I. Fotiadis, "Toward emotion recognition in car-racing drivers: A biosignal processing approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 3, pp. 502–512, 2008.

[82] W.-L. Zheng, K. Gao, G. Li, W. Liu, C. Liu, J.-Q. Liu, G. Wang, and B.-L. Lu, "Vigilance estimation using a wearable eog device in real driving environment," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 1, pp. 170–184, 2019.

[83] B. Hariri, S. Abtahi, S. Shirmohammadi, and L. Martel, "Vision based smart in-car camera system for driver yawning detection," pp. 1–2, 2011.

[84] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE transactions on vehicular technology*, vol. 53, no. 4, pp. 1052–1068, 2004.

[85] B. Reddy, Y.-H. Kim, S. Yun, C. Seo, and J. Jang, "Real-time driver drowsiness detection for embedded system using model compression of deep neural networks," pp. 121–128, 2017.

[86] T. Lindeberg, "Scale-space theory in computer vision," vol. 256, 2013.

[87] A. Ghofrani, R. M. Toroghi, and S. Ghanbari, "Realtime face-detection and emotion recognition using mtcnn and minishufflenet v2," pp. 817–821, 2019.

[88] F. You, X. Li, Y. Gong, H. Wang, and H. Li, "A real-time driving drowsiness detection algorithm with individual differences consideration," *Ieee Access*, vol. 7, pp. 179396–179408, 2019.

[89] Y.-h. Li, F. You, K. Chen, L. Huang, and J.-m. Xu, "A real-time system for monitoring driver fatigue," *Transportation Planning and Technology*, vol. 39, no. 8, pp. 779–790, 2016.

[90] W. Deng and R. Wu, "Real-time driver-drowsiness detection system using facial features," *Ieee Access*, vol. 7, pp. 118727–118738, 2019.

[91] J. Hughes, "Idealist thought and music theory in nineteenth-century germany: Kcf krause, dissonance, and" coming-to-be"," *International review of the aesthetics and sociology of music*, pp. 3–12, 1996.

[92] B. K. Savaş and Y. Becerikli, "Real time driver fatigue detection system based on multi-task connn," *Ieee Access*, vol. 8, pp. 12491–12498, 2020.

[93] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "Yawdd: A yawning detection dataset," pp. 24–28, 2014.

[94] V. Kumar, S. Sharma, *et al.*, "Driver drowsiness detection using modified deep learning architecture," *Evolutionary Intelligence*, pp. 1–10, 2022.

[95] "Vehical: Verified human interfaces, control, and learning for semi-autonomous systems," *VeHICaL: Verified Human Interfaces, Control, and Learning for Semi-Autonomous Systems*.

[96] A. Byrnes and C. Sturton, “On using drivers’ eyes to predict accident-causing drowsiness levels,” pp. 2092–2097, 2018.

[97] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, “Hyper-parameters optimisation of deep cnn architecture for vehicle logo recognition,” *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 939–946, 2018.

[98] W.-Y. Lee, S.-M. Park, and K.-B. Sim, “Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm,” *Optik*, vol. 172, pp. 359–367, 2018.

[99] J. L. Funk and R. D. Rogge, “Testing the ruler with item response theory: increasing precision of measurement for relationship satisfaction with the couples satisfaction index.,” *Journal of family psychology*, vol. 21, no. 4, p. 572, 2007.

[100] M. Goadrich, L. Oliphant, and J. Shavlik, “Learning ensembles of first-order clauses for recall-precision curves: A case study in biomedical information extraction,” pp. 98–115, 2004.

[101] D. Fourure, M. U. Javaid, N. Posocco, and S. Tihon, “Anomaly detection: how to artificially increase your f1-score with a biased evaluation protocol,” pp. 3–18, 2021.