DISTRACTED DRIVER DETECTION USING CONVOLUTIONAL NEURAL NETWORK

THANDAR OO

M.C.Tech. **DECEMBER 2022**

DISTRACTED DRIVER DETECTION USING CONVOLUTIONAL NEURAL NETWORK

By

THANDAR OO
B.C.Tech. (Hons.)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Technology
(M.C.Tech.)

University of Computer Studies, Yangon
DECEMBER 2022

ACKNOWLEDGEMENTS

I would to give my sincere thanks to people who have assisted and guided me during the master study.

First and foremost, I especially thank Dr. Mie Mie Khin, Rector, University of Computer Studies, Yangon, for allowing me to develop my research and giving me supporting during the period of my dissertation.

I am remarkably and especially thankful to Dr. Htar Htar Lwin**,** Pro-Rector**,** Faculty of Computer Systems and Technologies, the University of Computer Studies, Yangon.

Secondly, I would like to express my deepest gratitude to my supervisor Dr. Amy Tun, Professor, Faculty of Computer Systems and Technologies, the University of Computer Studies, Yangon, as Dean of Master's Course, for her caring and encouragement, and providing me with tremendous support and guidance during my master study.

In addition, I would also like to express my respectful gratitude to Daw Aye Aye Khine**,** Associate Professor**,** Head of the Department of English, University of Computer Studies, Yangon. I would like to thank her for valuable supports and revising my dissertation from the language point of view.

Finally, I would like to extend my thanks to all my teachers who taught me throughout the master's degree course, and my friends for their cooperation. The completion of my dissertation would not have been possible without the support and nurturing of my family.

ABSTRACT

The major cause of accidents is due to driver’s distracted actions. Most of the car accidents involve driver distraction under different forms such as talking on the phone, texting, operating the radio, drinking and talking with the passenger and so on. In most cases of distractions, a driver just keeps only one hand or even no hand on the steering wheel. Therefore, detecting driver’s distracted behavior aims at several goal, i.e. the levels of pay attention of a driver to the road and using two hands or not while driving. In this system, deep learning method is used to detect and classify the driver’s action. This system is developed by using Fine-Tuning-AlexNet Convolutional Neural Network to train and classify the driver’s distracted behaviors. This system is also implemented by Python programming language.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

Road traffic accident claims the lives of almost 1.3 million people each year, according to research by the WHO (World Health Organization,2022). Most nations lose 3% of their gross domestic product to road accidents. The rate of fatalities from road accidents is highest in Africa and lowest in Europe. Unqualified and inexperienced drivers, technologically flawed vehicles, poor traffic management, a lack of public awareness, excessive highway vehicle speeds, defective road construction, insufficient enforcement of traffic laws, and distracted driving behaviors are the main causes of traffic accidents.

Distracted driving is happened when a driver is engaged in something else while they are driving. Driving while distracted might raise the likelihood of a car accident. Distracted driving includes a variety of activities, including as texting, chatting on the phone, using a navigation system, and eating while operating a vehicle. Any one of these hazards can put the driver, the passengers, and other road users in danger. The three basic categories of inattentive driving are as follows: (i) visual (keeping the driver's eyes off the road), (ii) manual (keeping the driver's hands off the wheel), and (iii) cognitive (keeping the driver's mind off operating the vehicle).

Deep learning techniques, which identify and categorize a driver's behaviors whether they are distracted or not, play a significant role in helping to eliminate such distracted activities. The best current solutions for many issues in image classification, image recognition, speech recognition, and natural language processing involve neural networks and deep learning. In order to classify between different driving behaviors, this study focuses on the most modern deep learning-based systems, convolutional neural networks, and image classification methods. This system receives a picture of a driver from the dataset as input and then outputs a particular kind of driver action. The behaviors will be put into one of the following groups: 1. driving safely or normally, 2. texting with the right hand, 3. talking on the phone with the right hand, 4. texting with the left hand, 5. talking on a phone with the left hand, 6. operating the radio, 7. drinking, 8. reaching behind, 9. dressing hair and makeup, and talking to the passengers. To build such a proposed system, an appropriate dataset, well-trained transfer learning model and evaluation methods for the neural networks have to be applied.

1.1 Objectives of the Thesis

The main objectives of the thesis are as follows:

- To learn the image classification with deep learning.
- To apply CNN Models.
- To explain Fine-Tuning-AlexNet Architecture.
- To realize distracted driver detection system by fine-tuning for transfer learning.
- To evaluate the classification accuracy of the applied method.

1.2 Motivation of the Thesis

Road crash often leads to injuries, loss of property, and even deaths of people. According to the latest WHO data published in 2020, Road Traffic Accidents Deaths in Myanmar reached 11,004 or 3.05% of total deaths. The age adjusted Death Rate is 20.94 per 100,000 of population ranks Myanmar #71 in the world [12]. More than 80% of road accidents are caused by distracted driving, such as using a mobile phone, talking to passengers, smoking, and drinking while driving. Driver distraction is one of the major problems worldwide. Therefore, it is essential to monitor and analyze the driver's behavior during the driving time to detect the reaction and mitigate the number of road accidents. To detect and classify the driver's distracted behaviors, deep learning method and image classification techniques play an important role. By controlling the driver concerned with his/her distracted action, road accidents may be reduced and save the lives of people.

1.3 Organization of the Thesis

This thesis is organized into five chapters. In chapter one, the distracted driver detection system is introduced. This chapter also described the objectives of the thesis, the motivation of the thesis, and the organization of the thesis.

In chapter two, background theory of deep learning and its methods are presented.

In chapter three, the convolutional neural network and FT-AlexNet model are discussed in detail.

In chapter four, the proposed system design, the distracted driver dataset, the implementation of the system, and experimental results are expressed in detail.

In chapter five, the conclusion of the thesis work is presented. In addition, further extensions of the system are depicted.

CHAPTER 2
BACKGROUND THEORY

In this section, the theory background of deep learning models and their categories are discussed. Models that are based on learning styles are of four kinds in which deep learning models are trained with data; supervised-learning, unsupervised-learning, semi-supervised learning, and reinforcement learning. Models that are based on depth are further categorized into two types; Shallow learning and Deep Learning. Some topics expressed in this section are the basic concepts for deep neural networks, image processing, and image classification system for detecting the driver’s distracted behaviors.

2.1 Basic Concepts of Neural Networks

A neural network is a type of artificial intelligence that gives computers instructions on how to interpret data in a way that is similar to how the human brain does it. Deep learning is a type of machine learning that imitates the human brain by using interconnected neurons or nodes in a layered framework. Computers use an adaptive system that it builds to continuously learn from their mistakes and advance. As a result, artificial neural networks function to more precisely address challenging problems like image categorization or many recognition systems.

With only a little help from humans, neural networks can support the computer's ability to make intelligent decisions. This is due to their capacity to learn and model complicated, non-linear correlations between input and output data. With or without explicit training, neural networks can interpret unstructured input and draw broad conclusions.

There are several significant applications for neural networks, including computer vision, speech recognition, natural language processing, and others. The ability of computers to gather data and insights from pictures and movies is known as computer vision. Computers can differentiate and recognize images that resemble humans using neural networks. Applications for computer vision include picture classification, content filtering, facial recognition, and visual identification in self-driving cars. Despite variable speech patterns, pitch, tone, language, and accent, neural networks can analyze human speech. The capacity to process naturally written material by humans is known as natural language processing (NLP). Computers use neural networks to extract information and meaning from text-based texts and data.

Neural network architecture draws its influence from human brain function. Neurons in the human brain constitute a sophisticated, intricately linked network and communicate with one another via electrical signals to aid in human processing. Similar to this, a problem-solving artificial neural network is constructed of artificial neurons. Artificial neural networks are software programs or algorithms that use computing systems to accomplish mathematical computations, and artificial neurons are software modules, also known as nodes.

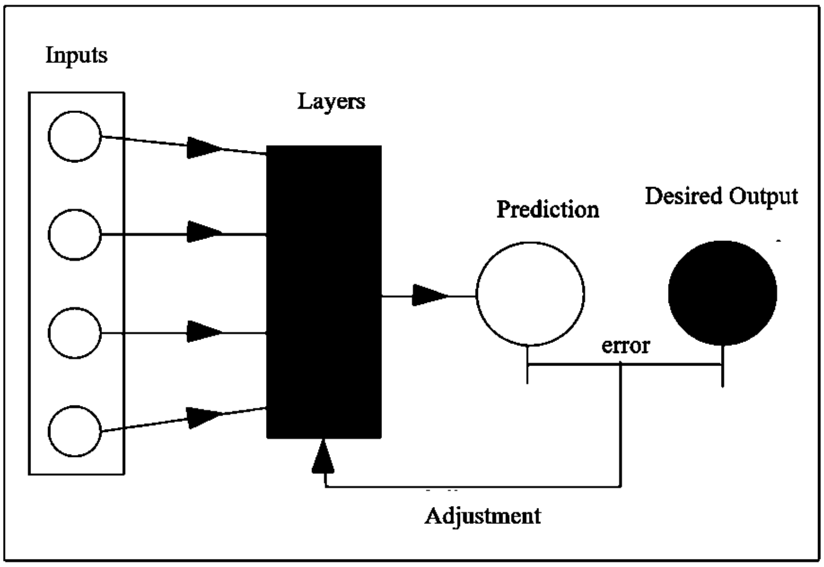


Figure 2.1. Basic Neural Network [1]

The input layer, hidden layer, and output layer are the three layers that make up a basic neural network as shown in Figure 2.1. The input layer is where data from the outside world enters an artificial neural network. Data is processed, analyzed, or categorized by input nodes before being forwarded to the following layer. The input layer or other hidden layers serve as the input for hidden layers. It is possible for artificial neural networks to have a lot of hidden layers. Each hidden layer evaluates the output from the preceding layer, refines it, and then sends it to the following layer. The output layer provides the whole data processing performed by the artificial neural network as the final output. It may include one or more nodes. For binary (yes/no) classification problem, the output layer will have one output node. However, the output layer might consist of more than one output nodes for multi-class classification problem.

A kind of artificial intelligence known as machine learning allows computers to access enormous datasets and train them to learn from them. In order to make wise decisions, machine learning software analyzes historical data for patterns and then applies those patterns to fresh data. Deep learning is a branch of machine learning that processes data using deep neural networks. For machine learning software to function well using traditional approaches, human input is necessary. The collection of pertinent features that the software must examine is manually selected by a data scientist. The software's capabilities are so constrained, making its creation and administration laborious. In contrast, the data scientist merely provides the software with raw data for deep learning. The deep learning network derives the features by itself and learns more independently. It can analyze unstructured datasets like text documents, identify which data attributes to prioritize, and solve more complex problems [3].

2.2. Deep Neural Networks

Artificial intelligence deep learning is represented by neural networks as shown in Figure 2.2. Deep Learning is a branch of computer science's machine learning area that uses numerous layers and large datasets to train models. Deep learning is modeled after how the human brain functions. Deep learning techniques use a variety of intricate systems to mimic human intellect. Deep learning requires pattern recognition, and computers can even perform this task without considerable programming. Deep learning enables machines to recognize objects and carry out any work in a way that is human-like using photos, text, or audio information. The key here is data exposure. If properly educated computers can successfully mimic human performance and occasionally produce accurate results. Deep learning emphasizes iterative learning techniques that expose computers to enormous data sets. This makes it easier for computers to recognize patterns and adjust to change. Machines learn differences and logic through repeated exposure to data sets, which helps them draw trustworthy conclusions from data. Recent advancements in deep learning have improved its dependability for difficult functions.

Small functional neural networks were developed by Soviet mathematician Alexey Ivakhnenko in the middle of the 1960s, and they are regarded as the first significant advance in deep learning.

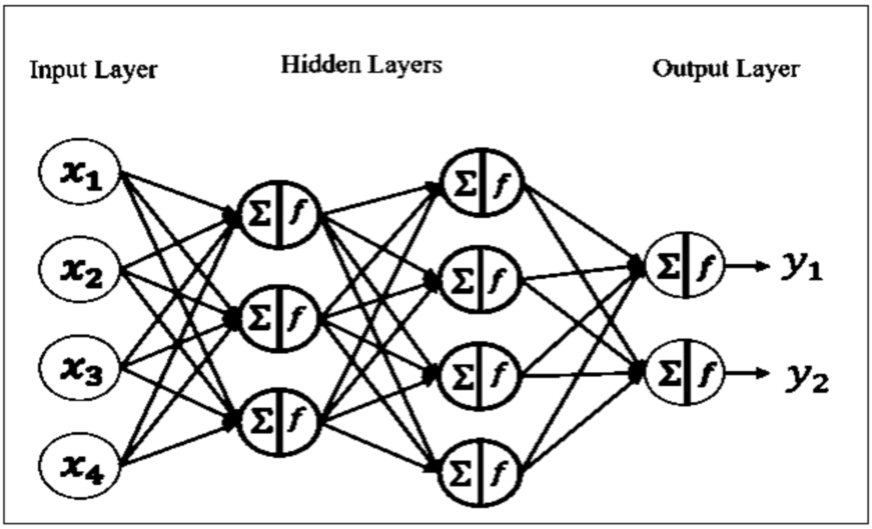


Figure 2.2. Deep Neural Network

Deep Learning is significant because it significantly improves the convenience of people's daily lives, and this tendency will continue to expand. Deep learning is behind a lot of automation in the modern world, whether it is automated parking assistance or face recognition at the airport. In the near future, deep learning will continue to have a significant impact on both the commercial and personal worlds and generate numerous career possibilities.

2.2.1 How does Deep Learning work?

Deep learning uses repeated teaching techniques to educate computers to mimic human intellect. This iterative process is carried out using an artificial neural network at various hierarchical levels. The first levels aid in teaching the machines basic knowledge, and as the levels advance, the knowledge keeps growing. Machines learn more information with each successive stage and blend it with what they already know. A compound input is the last piece of information the system collects during the procedure. This data is organized in multiple tiers and resembles sophisticated logic[5].

Each input node receives a value after receiving the information (in numerical form). Nodes transfer the activation value based on the transfer function and connection strength, and nodes with greater numbers have more activation value. The nodes calculate the whole amount and modify it in accordance with the transfer function once they have received the activation value. Applying the activation function, which aids the neuron in determining if a signal needs to be transmitted, is the following step in the procedure. Weights are applied to the synapses after the activation step in order to create the artificial neural network. Weights are essential for instructing an ANN on how to operate. To determine how far signals can travel, weights can also be changed. While an Artificial Neural Network is being trained, activation weights are often changed. The network reaches the output nodes after the activation process. The step that serves as a conduit between the user and the system is this one. The information is interpreted by the output node so that the user may understand it. Cost functions are used to assess the model's performance by comparing the expected and actual output. To lower loss function, one can select from a variety of cost functions depending on the requirements. The more precise output will be produced with a lower loss function.

Backpropagation, also known as backward propagation, is a technique for determining error function gradients in accordance with neural network weights. By using a backward calculation method, the desired outcome is achieved by removing the wrong weights [14]. On the other hand, forward propagation is a cumulative approach to achieving the desired output. In this approach, the input layer processes the data before sending it over the network. Errors are calculated after outcome values are contrasted with expected results, and the information is propagated backward. The network can be tested to see how it performs after tweaking the weights to the ideal level.

2.2.2 Types of Deep Neural Networks

Neurobiology served as the first inspiration for deep neural network models. By classifying its input patterns, the complexity of many inputs is decreased. Artificial neural network models are made up of units that aggregate several inputs to produce a single output and are motivated by this notion. The basic building block of neural networks, which aim to mimic brain activity, is a nonlinear function of the weighted sum of the inputs, such as max (0, value). These phony neurons are grouped into layers, with the outputs of one layer serving as the input for the subsequent layer in the chain. Deep neural networks use neural network structures. "Deep" functions are those that have more layers and units per layer, as well as higher levels of complexity. The ability to manage large datasets in the cloud made it possible to build more accurate models by using additional and larger layers to capture higher levels of patterns.

The terms training (also known as learning) and inference (also known as prediction) relate to the development phase as opposed to the production or application phase of neural networks. The number of layers and kind of neural network is selected by the developer when designing the architecture of deep network systems, and training sets the weights.

There are many types of deep neural networks and three following types of deep neural networks are popularly used today:

- Multi-Layered Perceptrons (MLPs)
- Recurrent Neural Networks (RNNs)
- Convolutional Neural Networks (CNNs)

2.2.3 MLPs

A class of feedforward artificial neural networks is called a multilayer perceptron (MLP) (ANN). The most fundamental deep neural network, which consists of a number of fully linked layers, is an MLP model. Modern deep learning architectures need a lot of computer resources, but MLP machine learning techniques can get around this. The example of multilayer perceptron is shown in Figure 2.3.

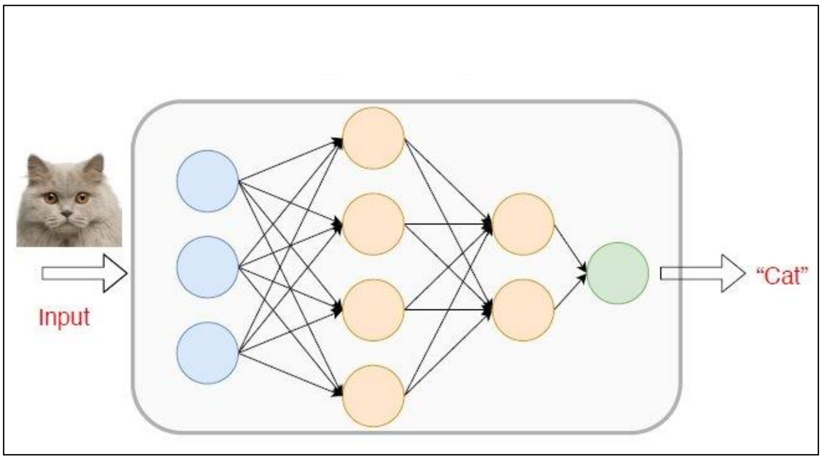


Figure 2.3. Multilayer Perceptron

2.2.4 RNNs

Another type of artificial neural network that utilizes sequential data feeding is a recurrent neural network (RNN). In order to solve the time-series problem of sequential input data, RNNs have been developed. The current input and the prior samples make up the RNN's input. A directed graph is created as a result of the connections between the nodes along a temporal sequence as shown in Figure 2.4. Additionally, each neuron in an RNN has its own internal memory where it stores the computation-related data from earlier samples.

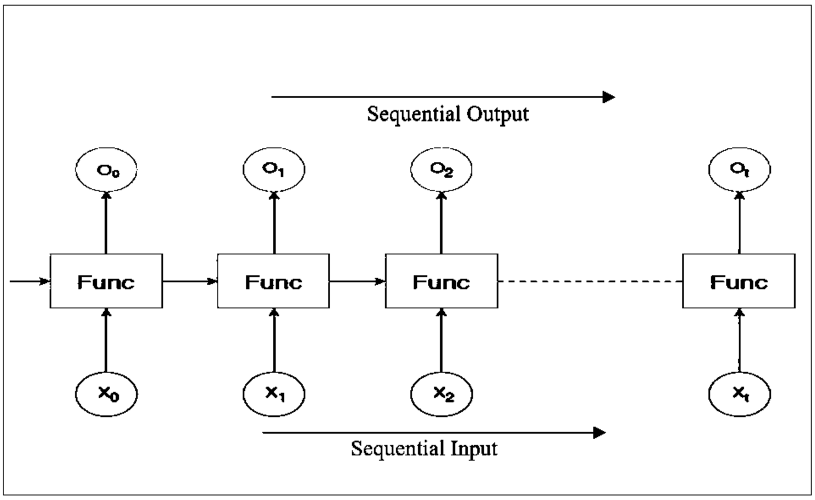


Figure 2.4. Recurrent Neural Network

RNN models are popular in Natural Language Processing (NLP) because they perform better while processing data with variable input length. The goal of the AI in this situation is to create a system that can understand human-spoken natural language through techniques like machine translation, word embedding, and natural language modeling. Each additional layer in RNNs is composed of nonlinear functions that are weighted sums of the outputs and the prior state. As a result, the fundamental building block of an RNN is referred to as a "cell," and each cell is made up of layers and a succession of cells to allow for the sequential processing of recurrent neural network models.

2.2.5 CNNs

Another type of deep neural network is a convolutional neural network as shown in Figure 2.5. The most typical application of CNNs is in computer vision. The AI system learns to automatically extract the properties of these inputs to finish a specified goal, such as picture classification, face identification, or image semantic segmentation, given a sequence of real-world images or videos and using CNN. In contrast to fully linked layers in MLPs, convolution operations are used by one or more convolution layers in CNN models to extract the simple characteristics from the input. Each layer consists of a collection of nonlinear functions that compute weighted sums of spatially close-by subsets of the outputs from the preceding layer at various positions.

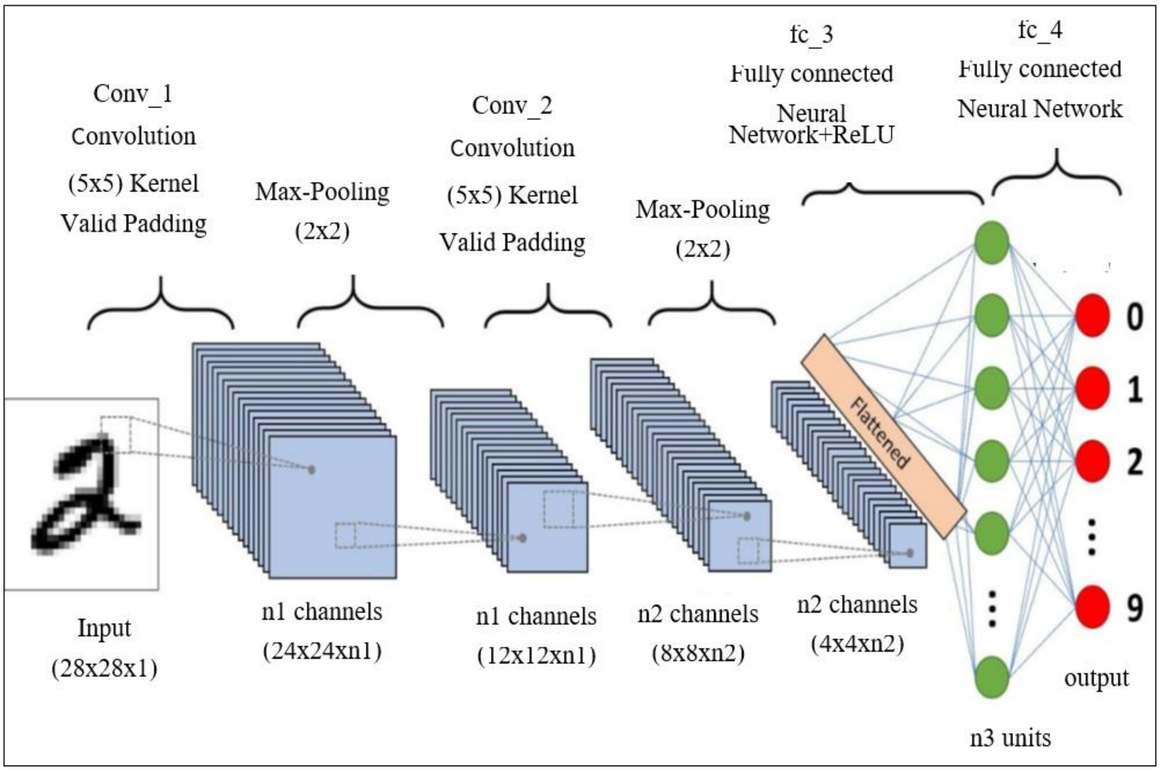


Figure 2.5. Convolutional Neural Network [13]

CNN machine learning models can capture the high-level representation of the input data by applying several convolutional filters, making CNN techniques very common in computer vision tasks. Examples of convolutional neural network applications include object detection and image categorization (using tools like AlexNet, VGG network, ResNet, and MobileNet) (e.g., Fast R-CNN, Mask R-CNN, YOLO, SSD). CNN techniques are as follows:

- **AlexNet**. AlexNet, the first CNN neural network to triumph in the 2012 ImageNet Challenge, has three fully connected layers and five convolutional layers for image categorization. In order to categorize the image with a size of 227*227, AlexNet needs 61 million weights and 724 million MACs (multiply-add computation).
- **VGG-16**. In order to categorize the image with a size of 224*224, the VGG-16 is trained to a deeper structure of 16 layers, comprising of 13 convolutional layers and 3 fully connected layers, using 138 million weights and 15.5G MACs [7].

- **GoogleNet**. GoogleNet proposes an inception module made up of various-sized filters in order to increase accuracy while decreasing the computation required for DNN inference. As a consequence, GoogleNet outperforms VGG-16 in accuracy performance despite processing an image of the same size with only seven million weights and 1.43G MACs.
- **ResNet**. ResNet, a cutting-edge project, uses the "shortcut" structure to achieve accuracy comparable to that of a person and a top-5 error rate under 5%. In order to train a DNN model with a deeper structure, the "shortcut" module is also utilized to address the gradient vanishing issue during training [15].

2.3 Related Work of the System

The "Multiple Scale Faster-RCNN Approach to Driver's Cell-phone Usage and Hands on Steering Wheel Detection" was discussed by T. Hoang Ngan Le, Yutong Zheng, Chenchen Zhu, Khoa Luu, and Marios Savvides of the CyLab Biometrics Center and the Department of Electrical and Computer Engineering at Carnegie Mellon University in Pittsburgh, Pennsylvania, USA [23]. The presentation for a sophisticated deep learning-based technique to automatically determine whether a driver is using a cell phone and whether his or her hands are on the wheel was accessible utilizing this framework. For that study, which deals with using a phone while driving, a speedier Region-Based Convolutional Neural Networks (R-CNN) model was developed. This is done to show that the motorist is using a phone while having both hands on the wheel. This approach uses shallower convolution feature maps from conv3, conv4, and conv5 to create the region proposal network (RPN) that is typically used for ROI pooling. Their research using the VIVA and SHRP-2 databases revealed that their suggested approach had superior accuracy, required fewer tests, and was not dependent on face land marking compared to the state of the art. Additionally, their MS-FRCNN has archived higher accuracy while remaining at a similar cost compared to Faster R-CNN.

On February 26, 2022, Mustafa Aljasim and Rasha Kashef of the Ryerson University's Electrical, Computer, and Biomedical Engineering department in Toronto, Ontario, Canada, implemented the "E2DR: A Deep Learning Ensemble-Based Driver Distraction Detection with Recommendations(E2DR) Model" [9]. In that article, they proposed the E2DR, a novel scalable model that combines two or more deep learning models by stacking ensemble approaches in order to promote generalization, minimize overfitting, and boost accuracy. Using cutting-edge datasets like the State Farm Distracted Drivers dataset, the best-performing E2DR variation, which comprised the ResNet50 and VGG16 models, attained a test accuracy of 92%. Future research can pay attention to how well more than two models integrate using the restricted computer resources available for the trials since it was impossible to evaluate other combinations.

On June 16, 2020, "Detecting Human Driver Inattentive and Aggressive Driving Behavior Using Deep Learning: Recent Advances, Requirements and Open Challenges" was published. It was written by Monagi H. Alkinani, Wazir Zada Khan, and Quratulain Arshad from the College of Computer and Engineering and the Department of Computer Science and Artificial Intelligence at the University of Jeddah in Jeddah, Saudi Arabia [16]. The authors contributed to the first identification and analysis of human driver inattentive driving behavior (HIDB) by classifying it into two main categories: driver distraction (DD) and driver tiredness (DF), often known as drowsiness (DFD). The origins and effects of another risky driving behavior known as aggressive driving behavior were then examined (ADB). In this study, we critically review the most recent deep learning-based systems, algorithms, and techniques for the detection of distraction, fatigue/drowsiness, and aggressiveness in human drivers. A few big and crucial open research challenges are mentioned together with their future orientations.

Automatic driver distraction detection using deep convolutional neural networks, Md. Uzzol Hossain, Md. Ataur Rahman, Md. Manowarul Islam, Arnisha Akhter, Md. Ashraf Uddin, and Bikash Kumar Paul, Department of Computer Science and Engineering, Jagannath University, Dhaka, Bangladesh; Internet Commerce Security Laboratory, Federation University, Australia; Mawlana Bhashani Science and Technology University, Bangladesh. This system uses a face and hand localization technique to recognize distracted driving and to pinpoint the location of distractions like talking, nodding off, or eating. To facilitate transfer learning, four architectures—CNN, VGG-16, ResNet50, and MobileNetV2—have been used. Thousands of images from a publicly accessible dataset that contains ten scenarios of a driver are then used to test the results using a variety of performance criteria to demonstrate the model's effectiveness.

CHAPTER 3
CONVOLUTIONAL NEURAL NETWORK

In this chapter, image preprocessing and image classification methods will be described. Then, the process of convolutional neural network and Fine-Tuning-AlexNet CNN model will be presented in detail.

3.1 Image Preprocessing

The unprocessed photos gathered from the scanning facility, and due to a number of factors, websites are not ideal for immediate processing. These photos contain noise. Consequently, it needs to be preprocessed before being examined. Pre-processing is crucial. Steps utilized in magnetic resonance images include labeling, removing artifacts, enhancing, and divisions. The conversion's preliminary processing, image resizing, noise removal, and quality improvement creates a picture that allows for the accurate detection of minute details [5].

Each pixel's value was deducted from the average RGB value across all pixels. The dataset's mean value is subtracted to center the data. Because our model must be trained by multiplying weights and adding biases to the initial inputs in order to produce activations that are then backpropagated with the gradients, mean subtraction is necessary. To keep the gradients from straying outside of certain ranges, it is crucial that each feature has a similar range. Additionally, because CNNs share parameters, sharing is more difficult if the inputs are not scaled to have values in a similar range. This is because a specific area of the image will have a high value [12].

3.1.1 Rescaling Image

In image preprocessing, the image is scaled down by factor 255 before feeding to the model. Every digital image is formed by pixels having values in the range of 0~255. 0 is black and 255 is white. For colorful image, it contains three maps: Red, Green and Blue, and all the pixel still in the range 0~255. (pixel value ranges according to the storage size, the pixel range is 0~2^bits). Since 255 is the maximin pixel value. Rescale 1./255 is to transform every pixel value from range [0,255] -> [0,1]. The aggregate of them will contribute to the backpropagation update. High-range images produce larger loss, whilst low-range images typically produce a weaker loss. Without scaling, it will take a lot of votes to decide how to update weights for photos with high pixel ranges. For instance, a black-and-white cat image may have a greater pixel range than a pure black cat image, but that does not necessarily imply that it is more crucial for training [16].

3.1.2 Resizing Image

Resizing changes your images size and, optionally, scale to a desired set of dimensions. Resize process is as follows:

- Stretch to: The image can be enlarged to a desired pixel-by-pixel size. Scaling for annotations is proportionate. No source picture data is lost despite the square and deformed images.
- Fill (with center crop) in: Your selected output dimensions are cropped in the center of the created image. For instance, if the outputted resize is set to 416x416 and the source picture is 2600x2080, the outputted resize will be the center 416x416 of the source image. The source image data is lost, but the aspect ratio is preserved.
- Fit within: While retaining the aspect ratio of the source image, the source dimensions are scaled to match the dimensions of the output image. For instance, if the resize option is set to 416x416 and the original image is 2600x2080, the longer dimensions (2600) will be shrunk to 416 and the secondary dimension (2080) will be scaled to 335.48 pixels. The original data and image aspect ratios are preserved, although they are not square.
- Fit (reflect edges) in: While preserving the aspect ratio of the source image, the dimensions of the source dimension are scaled to be the dimensions of the output image, and any newly formed padding is a reflection of the source image. For instance, if the resize option is set to 416x416 and the original image is 2600x2080, the longer dimensions (2600) will be shrunk to 416 and the secondary dimension (2080) will be scaled to 335.48 pixels. Reflected pixels from the source image make up the remaining pixel area (416-335.48, or 80.52 pixels).
- Fit (black edges) in: The source image's dimensions are scaled to match those of the output image while preserving the source image's aspect ratio, and any newly added padding is represented by a black area. For instance, if the resize

option is set to 416x416 and the original image is 2600x2080, the longer dimensions (2600) will be shrunk to 416 and the secondary dimension (2080) will be scaled to 335.48 pixels. Black pixels make up the remaining pixel region (416-335.48, or 80.52 pixels). Images are square, with the original data and aspect ratios preserved [2].

- Fit (white edges) in: The source image's dimensions are scaled to match those of the output image while preserving the source image's aspect ratio, and any newly added padding is represented by a white area. For instance, if the resize option is set to 416x416 and the original image is 2600x2080, the longer dimensions (2600) will be shrunk to 416 and the secondary dimension (2080) will be scaled to 335.48 pixels. White pixels make up the remaining pixel region (416-335.48, or 80.52 pixels). Images are square, have white padding, keep aspect ratios, and retain original data [7].

3.1.3 Normalization

In order to guarantee that each input parameter (in this case, pixel) has a comparable data distribution, data normalization is a crucial step. This speeds up convergence while the network is being trained. When normalizing data, each pixel is first subtracted from its mean before the result is divided by the standard deviation. Such data would have a distribution that resembles a zero-centered Gaussian curve. Since positive pixel numbers are required for picture inputs, we may choose to scale the normalized data in the range [0, 1] or [0, 255] [8].

3.2 Image Classification

The main issues in the field of computer vision include object recognition, segmentation, classification, and localization of images. Among them, picture categorization might be thought of as the core issue. It serves as the foundation for more computer vision issues. Applications for image classification are employed in a variety of fields, including medical imaging, satellite image object identification, traffic control systems, brake light detection, machine vision, and more.

The process of classifying and labeling groups of pixels or vectors within an image based on specific rules is known as image classification. One or more spectral or textural characterizations can be used to apply the categorization law. The two primary kinds of image classification techniques are supervised and unsupervised image classification techniques.

There are two standard methods for classifying images:

- Supervised Classification: This method involves classifying remote sensing data using input from and guidance from the user in the form of training data.
- Unsupervised Classification: This method involves identifying natural groups or structures within a set of remote sensing data automatically [9].

3.2.1 Unsupervised Classification

A fully automatic methodology called unsupervised classification does not use training data. This means that without the need for human participation, machine learning algorithms are utilized to evaluate and cluster unlabeled datasets by identifying hidden patterns or data groups. The specific characteristics of an image are systematically recognized at the image processing stage with the aid of a suitable algorithm. Two of the most often used techniques for classifying images in this context are pattern recognition and image clustering [1].

- K-means is an unsupervised classification algorithm that groups objects into k groups based on their characteristics. It is also called "cluster." K-means clustering is one of the simplest and very popular unsupervised machine learning algorithms.
- The term "Iterative Self-Organizing Data Analysis Technique" (also known as "ISODATA") refers to an unsupervised technique for picture categorization. The ISODATA approach incorporates iterative techniques that divide data components into various classes using Euclidean distance as the similarity metric. The ISODATA approach allows for a varying number of clusters, but the k-means algorithm presumes that the number of clusters is known a priori.

With the aid of their attributes, pixels are grouped in unsupervised categorization. Groups are known as clusters and this process is called clustering. When trained pixels are not available, the unsupervised classification is utilized. Unsupervised classification does not require prior knowledge. It is entirely automated; no human annotation is necessary. This program locates data clusters and assigns cluster labels to them. Unsupervised categorization involves these steps:

- Data clustering

- Based on clusters, every pixel is classified.
- A spectrum class map.
- An analyst labels cluster
- Informational class on maps [9].

3.2.2 Supervised Classification

In order to train the classifier and subsequently classify fresh, unknown data, supervised image classification algorithms require previously classified reference samples (the ground truth). The method of visually selecting training data samples from inside the image and assigning them to pre-selected categories, such as flora, roads, water resources, and buildings, is hence known as the supervised classification technique. To establish statistical measurements that can be used to analyze the overall image, this is done. Maximum Likelihood, Minimum Distance, Artificial Neural Network, and Decision Tree Classifier are a few examples of supervised classification techniques.

Some pixels are known to be grouped in supervised classification, which assigns labels to classes. This procedure is referred to as training. Following that, the classifier uses trained pixels to categorize more photos. Prior information must be gathered by the analyst before the testing process. For each informative class, the analyst picks representative training locations, and the algorithm also creates decision boundaries in this section. The shortest distance to the mean, maximum likelihood, and parallelepiped are frequently used supervised classification techniques. The following are the steps in the supervised classification approach:

- An analyst designates training regions for each informational lesson.
- Signed documents identify (mean, variance, covariance, etc.)
- Each pixel has a classification
- Informational Class on Maps

The classifier benefits from having an analyst or domain expert who can help it understand how the data and classes relate to one another. Using this prior knowledge, it is possible to determine the number of classes and prototype pixels for each class. For partially supervised image classification, a combination of supervised and unsupervised techniques can be used [9].

3.3 Convolutional Neural Network

Due to its ability to process enormous volumes of data, deep learning has become recognized as a very effective technique during the previous few decades. Contrary to conventional approaches, hidden layer technology is significantly more often used for pattern recognition. One of the most popular deep neural networks is convolutional neural networks.

Researchers have struggled to create a system that can comprehend visual input ever since the 1950s, the early years of AI. This area of study eventually became known as computer vision. When a team of researchers from the University of Toronto created an AI model that significantly outperformed the best image recognition algorithms in 2012, computer vision experienced a quantum leap.

The 2012 ImageNet computer vision competition was won by the AI system, known as AlexNet (after its principal designer, Alex Krizhevsky), with an astounding 85 percent accuracy. The test result for the runner-up was a modest 74 percent. Convolutional Neural Networks, an unique kind of neural network that substantially mimics human vision, were at the core of AlexNet. Over time, CNNs have developed into a crucial component of many Computer Vision applications.

Around the 1980s, CNNs were first created and put to use. At the time, a CNN could only recognize handwritten numbers to a certain extent. To read zip codes, pin numbers, etc., it was mostly utilized in the postal industry. The most crucial thing to keep in mind about any deep learning model is that it needs a lot of computational power and data to train. Because of this significant disadvantage at the time, CNNs were restricted to the postal industry and were unable to enter the machine learning field.

In 2012, Alex Krizhevsky came to the conclusion that multi-layered neural networks, a subset of deep learning, should be revived. Researchers were able to resurrect CNNs because to the availability of big data sets, including ImageNet datasets with millions of annotated images and a wealth of processing power.

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. It makes use of a unique method known as convolution, or neural network matrix multiplications. A third function that expresses how the shapes of the two functions are changed by one another is created via convolution, a mathematical action on two functions. The

ConvNet's function is to compress the images into a more manageable format without sacrificing elements that are essential for obtaining an accurate forecast.

3.3.1 Basic CNN Architecture

The fundamentals, such as what an image is and how it is displayed, are discussed before looking at how CNN operates. The basic architecture of CNN is shown in Figure 3.1. A grayscale image is the same as an RGB image but only has one plane, whereas an RGB image is nothing more than a matrix of pixel values.

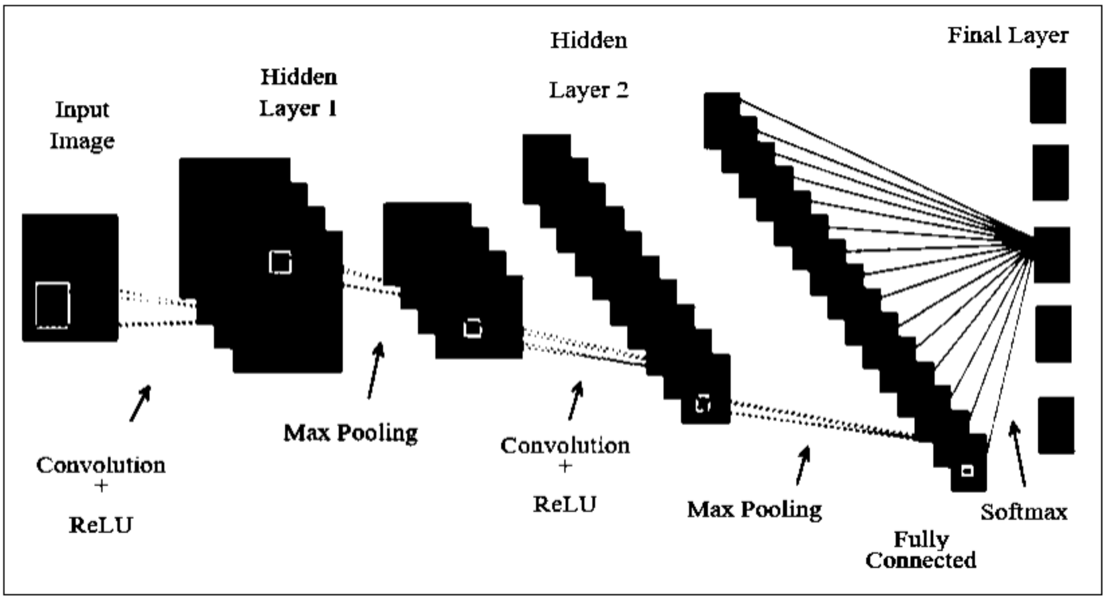


Figure 3.1. Basic CNN Architecture [24]

Convolutional neural networks are made up of many layers of artificial neurons. Artificial neurons are mathematical operations that, in a manner similar to that of their biological counterparts, compute the weighted sum of a number of inputs and output an activation value. When presented with an input image, each layer of a ConvNet creates a number of activation functions that are then passed on to the following layer.

Typically, the first layer extracts fundamental features like edges that run horizontally or diagonally. The following layer receives this output and detects more intricate features like corners or multiple edges [6]. Deeper networks are able to recognize ever more intricate elements like faces and objects. The classification layer generates a series of confidence ratings (numbers between 0 and 1) that indicate how likely it is for the image to belong to a "class" based on the activation map of the final convolution layer.

Similar to the Convolutional Layer, the Pooling layer is in responsibility of reducing the spatial size of the Convolved Feature. Lowering the dimensions will result in less CPU processing power being required to process the data. The two types of pooling are average pooling and maximal pooling. Max pooling is the highest value of a pixel from a kernel-covered region of the image. In addition to totally eliminating the noisy activations, it also does de-noising and dimensionality reduction. On the other hand, average pooling returns the average of all the values from the region of the picture covered by the Kernel. Average Pooling only reduces noise through dimensionality reduction.

CNNs are being employed in a variety of computer vision applications, including augmented reality, facial recognition, picture search, and editing. Although convolutional neural networks have limitations, there is no doubting that they have revolutionized artificial intelligence [12].

3.3.2 Input Layer

The input layer serves as the source for the entire CNN. It typically represents the image's pixel matrix in neural networks used for image processing. The raw pixel values of the photos are stored in the input layer. If colored images with a 640*480 pixel resolution are scaled down to 224*224 pixels to shorten training time [12].

3.3.3 Convolutional Layer

The central component of a CNN is the convolutional layer, which is also where the majority of computation takes place. It needs input data, a filter, and a feature map, among other things. The input will be a color image with three dimensions—height, width, and depth—that match to RGB in an image. This image is made up of a matrix of pixels.

Moving over the image's receptive fields, a feature detector—also known as a kernel or a filter—checks to see if the feature is there. Convolution describes this process. A two-dimensional (2-D) array of weights serving as the feature detector represents a portion of the image. Typically, the filter size is a 3x3 matrix. A portion of the image is subjected to the filter, and the dot product between the input pixels and the filter is computed. The output array is then fed with this dot product. Once the kernel has swept through the entire image, the filter shifts by a stride and repeats the operation as shown in Figure 3.2.

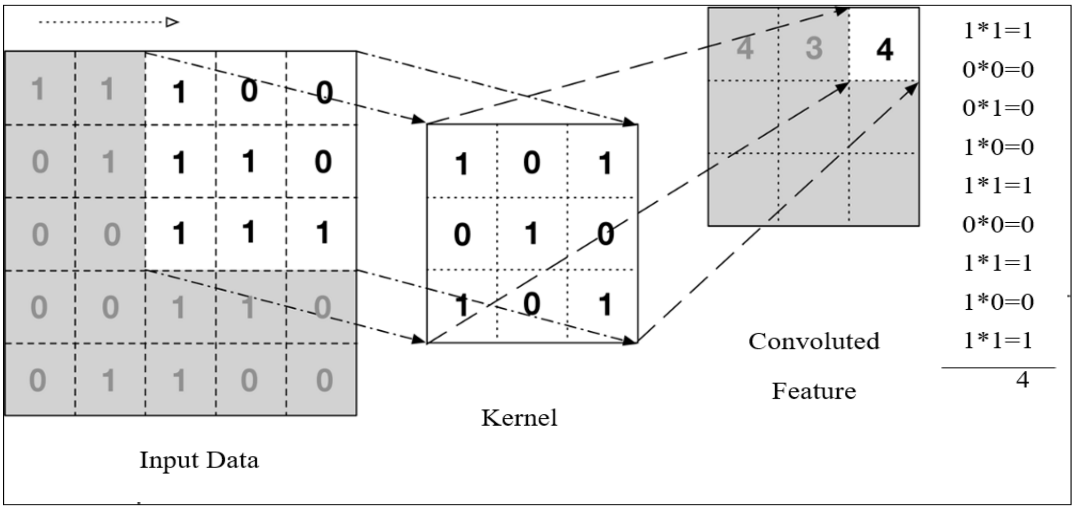


Figure 3.2. Convolutional Layer

A feature map, activation map, or convolved feature is the final result of the series of dot products from the input and filter. The number of filters, stride, and zero padding are the three hyperparameters that must be specified before to training in order to control the output volume size. The formula is used to know the output feature map (dimensions) after convoluting. This formula is as follows:

$$((n + 2p-f)/s) + 1 \qquad (3.1)$$

where, n=size of the image (eg. If 32x32x3 image, n=32),

f= size of the filter (3x3, f = 3)

p= the padding

s= the factor by which you want to slide (stride)

Zero padding is used to preserve loss information in convolution [11].

3.3.4 Pooling Layer

Down-sampling is the pooling layer, does dimensionality reduction by minimizing the number of parameters in the input. To minimize inefficient computing, the pooling layers lowered the input volume's 2D dimensions. Each depth slice's input data is subjected to a tiny filter in order to achieve this. There are numerous types of pooling filters, including average pooling, max pooling. Here are pooling filters:

- Max pooling: The filter chooses the pixel with the highest value to send to the output array as it advances across the input as shown in Figure 3.3.
- Average pooling: The filter calculates the average value inside the receptive field as it passes across the input and sends that value to the output [11].

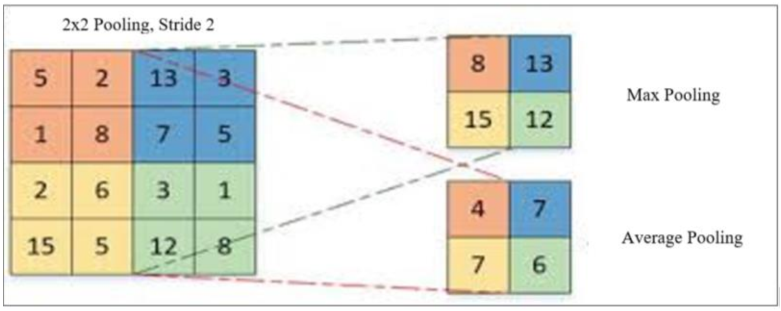


Figure 3.3. Pooling Layer

3.3.5 Fully Connected Layer

After feature extraction, the data is needed to classify into various classes, this can be done using a fully connected (FC) neural network. Most widely used machine learning models have a final set of fully linked layers that combine the data gathered by earlier levels to create the final output. A fully connected layer, which is a component of the convolutional network, uses the results of the convolution/pooling process to make a classification determination. The fully connected layers connect every neuron in one layer to every neuron in the other layer as shown in Figure 3.4.

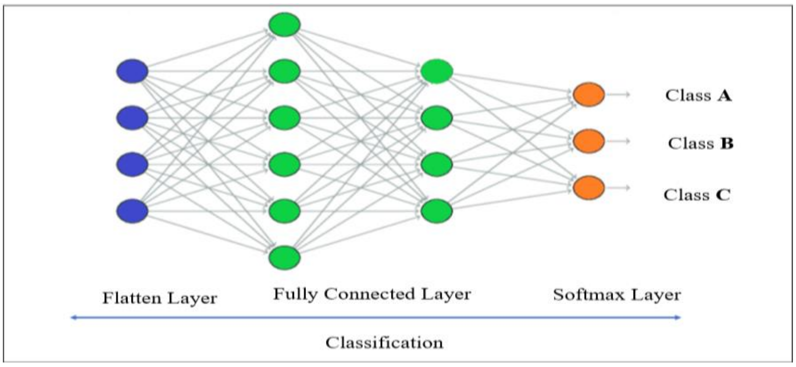


Figure 3.4. Fully Connected Layer [22]

Fully connected neural networks, the standard neural network architecture in which all neurons connect to all neurons in the next layer, are not to be confused with fully connected layers of a CNN. For computer vision, convolutional neural networks enable deep learning. It was discovered that computer vision tasks were inefficient for the traditional neural network architecture. A neural network's primary input is made up of images (they can have hundreds or thousands of pixels and up to 3 color channels).

This calls for a very large number of connections and network parameters in a traditional fully connected network.

An image is made up of smaller details, or features, which can be used by a convolutional neural network to develop a system for assessing each feature separately and making decisions about the image as a whole. The convolutional network includes a fully connected layer that makes a classification determination using the results of the convolution/pooling process [14].

3.3.6 Advantages of using CNN

The following are some benefits of utilizing a convolutional neural network to identify driver distraction:

- CNN will automatically identify the key elements required for the detection process without human intervention.
- CNN uses pooling and certain convolution processes, making it computationally efficient.
- It is possible to forecast or recognize the distraction in an efficient manner.
- CNN will offer a higher level of accuracy for image identification.
- A lightweight construction that plays a crucial role in many applications is used to remove noise.
- CNN is simple to learn and the deployment process may be completed more quickly.
- Both parameter sharing and the dimensionality reduction are considered. [10].

3.4 Activation Functions

The activation function decides, whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. The role of the Activation Function is to derive output from a set of input values fed to a node (or a layer). The activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks [4]. There are three types of activation functions are:

- Binary Step Function,
- Linear Function and

- Non-linear function.

Non-Linear activation is mostly used in CNN such as ReLU, Softmax, and Sigmoid. In this system, ReLU and Softmax activation function is used in the Alexnet model.

Properties of activation functions are as follows:

- Non-linearity: the derivative is not a constant. By doing this, the multilayer network can be prevented from becoming a single-layer linear network.
- Differentiability: This term describes how easily a gradient in optimization may be computed.
- Straightforward: Calculation speed will decrease with a complicated activation function.
- Saturation: Saturation is the issue where the gradient is close to zero in specific intervals (that is, the gradient disappears), which prevents the parameters from being updated.
- Monotonic: The derivative's sign remains constant. It is possible to ensure that the single-layer network is a convex function when the activation function is monotonic.
- Fewer parameters: The majority of activation functions don't have any [19].

3.4.1 Softmax Function

Softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels. Softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs. This function is ideally used in the output layer of the classifier to attain the probabilities and to define the class of each input.

$$\mathrm{S(yi)}= \frac{\exp(yi)}{\sum_{j=1}^{n}\exp(yi)} \qquad (3.2)$$

where, y = an input vector to the softmax function, S

y_i = i^{th} element of the input vector

exp (y_i) = standard exponential function applied on y_i

$\sum_{j=1}^{n}\exp(yi)$ = normalization term

n = numbers of classes.

3.4.2 ReLU Function

The Rectified Linear Unit activation function (ReLU for short) is a non-linear function or piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero**.** The rectified linear unit activation function overcomes the vanishing gradient problem, allowing models to learn faster and perform better. The plot of the ReLU function is shown in Figure 3.5. Mathematically, it is expressed as:

$$f(x) = \max(0,x) \tag{3.3}$$

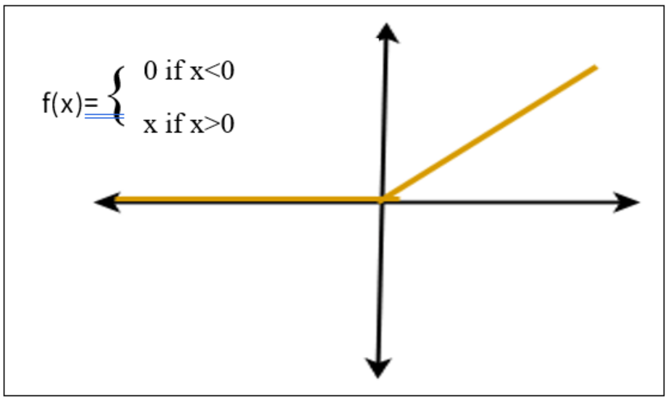


Figure 3.5. ReLU Function

3.5 Fine-Tuning-AlexNet Architecture

In this system, the Alexnet Model, trained on the ImageNet dataset, the winner of the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition, is fine-tuned by replacing the last fully connected layer and retraining the output parameters. Fine-Tuning-AlexNet architecture is composed of five convolutional layers with a combination of max pooling followed by three fully connected layers and uses ReLU activation in each of these layers except the output layer. To prevent overfitting, it also used the dropout layers. The input image size is 227x227x3.

In the first convolution layer, 96 filters of size 11x11 with stride 4 are used to extract features. The output feature map is 55x55x96. The first Max pooling layer of size 3x3 and stride 2 is used. Then the resulting feature map is the size of 27x27x96. In the second convolution, the filter size is reduced to 5x5 and has 256 such filters. The stride is 1 and padding 2. The output size is 27x27x256. Then, a max-pooling layer of size 3x3 with stride 2 is used. The resulting feature map is 13x13x256. FT-AlexNet architecture is shown in Figure 3.6.

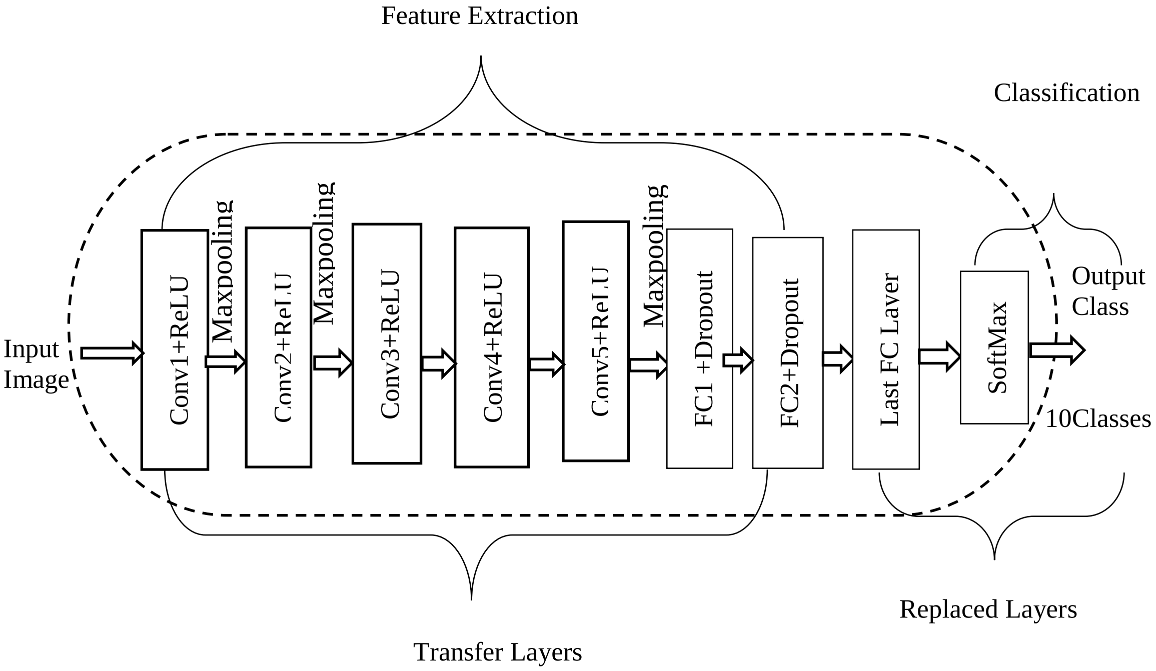


Figure 3.6. Fine-Tuning-AlexNet (FT-AlexNet) Arhitecture

The third convolution uses 384 filters that are 3x3, stride 1, and padding 1, respectively. The final feature map has the dimensions 13x13x384.

The fourth convolution utilizes 384 3x3 filters. The padding and stride together are 1. The output size, 13x13x384, stays the same.

The final convolution layer has 256 of these filters and is 3x3 in size. The final feature map has the dimensions 13x13x256. It uses the third max-pooling layer, which is a 3x3 stride layer. The feature map is 6x6x256 as a result. The first dropout layer is used which the drop-out rate is set to be 0.5. The first fully connected layer is with a ReLU activation function. The size of the output is 4096. Next layer comes another dropout layer with the dropout rate fixed at 0.5. Then, second fully connected layer is with 4096 neurons and ReLU activation. The last fully connected layer or output layer with 10 neurons is used in the FT-AlexNet Model and the activation function is Softmax. The network has a total of 58322314 (about 58 million) trainable parameters.

3.6 Evaluation Method

To measure the performance of distracted driver detection and classification process, this system is tested by using 102150 images that contains the 22,424 training images and 79,726 testing images, however, 20% of test data (about 15,945 images) are used.The performance evaluation methods are as follows:

- Accuracy: It is the percentage of correct classification of test dataset.

$$\text{Accuracy} = \frac{\text{Tp} + \text{Tn}}{\text{Tp} + \text{Tn} + \text{Fn} + \text{Fp}} \tag{3.4}$$

- Precision: It is the positive predictive value.

$$\text{Precision} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}} \tag{3.5}$$

- Recall: It is the ratio of correctly recognized images to the number of relevant images in dataset.

$$\text{Recall} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}} \tag{3.6}$$

- F-score: It is basically harmonic mean of recall and precision.

$$\text{F - Score} = \frac{2(\text{Precision} + \text{Recall})}{\text{Precision} + \text{Recall}} \tag{3.7}$$

- Error rate: It is the total numbers all incorrect predicted images to the total number of images in dataset.

$$\text{Error Rate} = \frac{\text{Fp} + \text{Fn}}{\text{Tp} + \text{Tn} + \text{Fn} + \text{Fp}} \tag{3.8}$$

A true positive (TP) is an outcome where the model correctly predicts the positive class. A true negative (TN) is an outcome where the model correctly predicts the negative class. A false positive (FP) is an outcome where the model incorrectly predicts the positive class. A false negative (FN) is an outcome where the model incorrectly predicts the negative class.

CHAPTER 4

THE PROPOSED DISTRACTED DRIVER DETECTION SYSTEM

In this chapter, the proposed distracted driver detection system will be explained with system flow diagram, the distracted driver dataset, the implementation of the system and experimental result of the system.

4.1 Proposed System Design

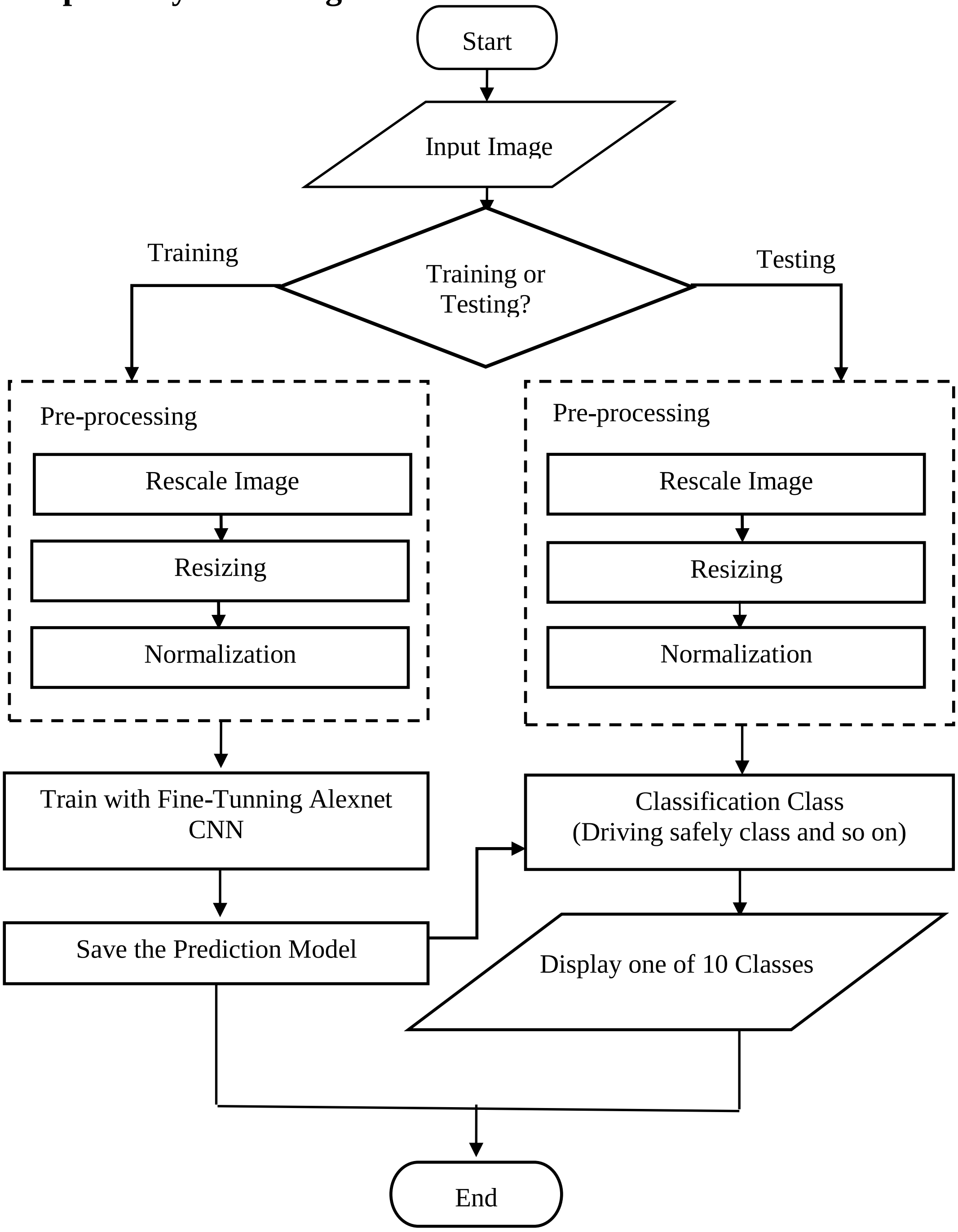


Figure 4.1. System Flow Diagram

This system is proposed as the districted driver detection system using Fine-Tuning-AlexNet Convolutional Neural Network. System flow diagram is shown in Figure 4.1. In the AlexNet architecture, the last fully connected layer with 1000 neurons is used to classify for multiple classes. However, in this system, Fine-Tuning-AlexNet architecture in which the last fully connected layer with 10 neurons is used to classify the 10 driving classes. Firstly, the user must load the districted driver image. Then, the user can choose the training or testing phase. To classify the districted driver class, this system performs the training process.

Before training process, it is needed to perform the pre-processing step that includes “rescaling”, “resizing” and “normalization” processes. Then, this system trains the Fine-Tuning AlexNet CNN model. In this network, five convolution layers and three fully connected layer with dropout and SoftMax activation function are used. The network is trained using the training data. After finishing training process, this system saves the prediction model.

In the testing process, the user must input the desired image. Then, this system performs pre-processing step. Then, this system classifies the tested image by using the prediction model that is obtained from the CNN. Finally, this system displays the relevant districted driver class.

4.2 Processing Steps of the Distracted Driver Detection System

The problem statement of the system is to detect the image of driver’s behavior which is distracted or not. The processing steps of the districted driver detection system are as follows:

- Step-1: Import necessary Libraries
- Step-2: Loading the data and Basic Exploratory Data Analysis
- Step-3: Preparing the training data and testing data
- Step-4: Building the Model
- Step-5: Train the Model
- Step-6: Evaluating the classification accuracy

4.3 Distracted Driver Dataset

This system uses the “StateFarm” dataset as districted driver dataset that is provided by “Kaggle from StateFarm” dataset which is created in 2016[10]. The dataset includes 10 classes, and each image is classified among distracted driver classes. The dataset includes 102,150 images of drivers that are separated into 10 categories and all images are 640x480 pixels images for 26 different drivers.

The dataset is split into train and test sets. The train set contains 22,424 images and the test set contains 79,726 images, therefore, 20% of the test set (about 15,945 images) are used randomly for testing.

A total of 10 classes are used for classifying driver's behavior, as these were identified as the most common activities performed which led to distraction while driving. For each class, about 1595 testing images are used for testing. Out of the 10 classes, 1st class indicates safe driving while the remaining 9 classes indicate distraction while driving.

The categories include the following driver actions:

- Class 1: Driving safely,

  Safe Driving as shown in Figure 4.2 is defined as the act of having hands on the steering wheel and looking in front while driving. The driver also must not engage in any of the activities mentioned below.

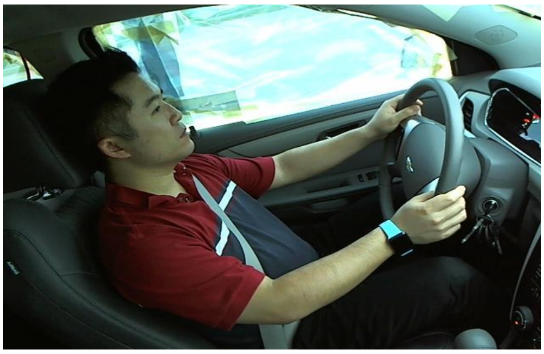


Figure 4.2. "Driving safely" Class

- Class 2: Texting with the right hand,

  If the driver is engaged in texting on his phone with his right hand while driving, then the act will be classified in this category. The act is shown in Figure 4.3.

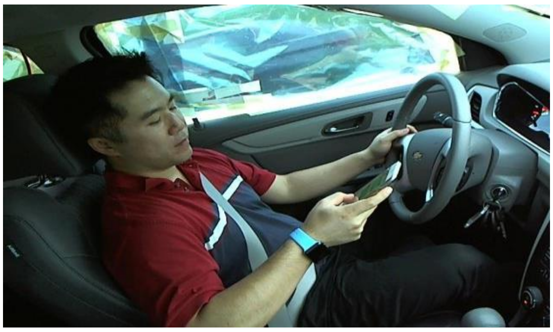


Figure 4.3. "Texting with the right hand" Class

- Class 3: Talking on the phone with the right hand,

  Performing the activity of making a call with the right hand as shown in Figure 4.4 while driving is also a class in the distraction category.

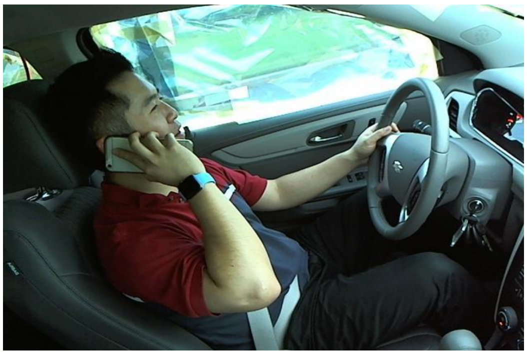


Figure 4.4. "Talking on the phone with the right hand" Class

- Class 4: Texting with the left hand,

  In this category the driver is texting with his left hand while driving. The act is shown in Figure 4.5.

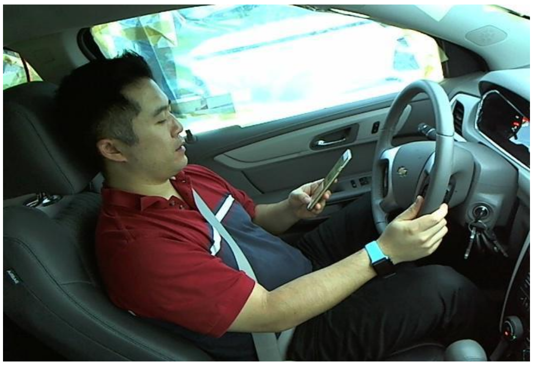


Figure 4.5. "Texting with the left hand" Class

- Class 5: Talking on the phone with the left hand,

  If the driver is making a call using his left hand, as indicated in Figure 4.6, the driver will fall into this category.

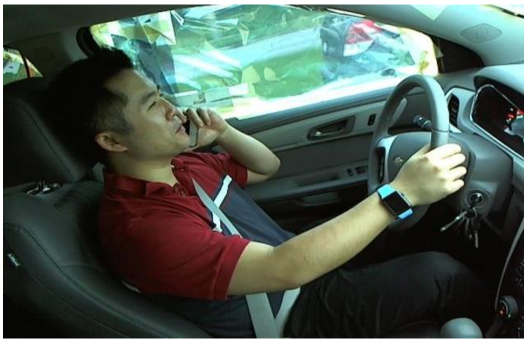


Figure 4.6. "Talking on the phone with the left hand" Class

- Class 6: Operating the radio,

  Operating the radio while driving is also identified as a possible cause of distraction. Figure 4.7 depicts a driver performing the mentioned task

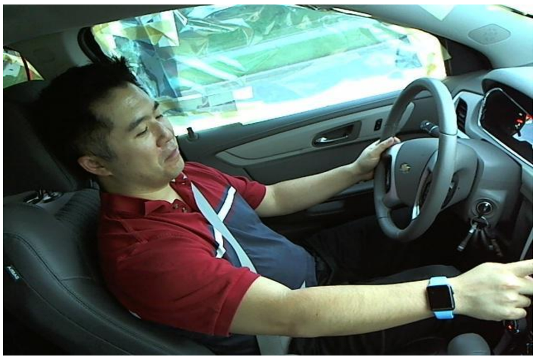


Figure 4.7. "Operating the radio" Class

- Class 7: Drinking,

  Drinking while driving as shown in Figure 4.8 is another class included and is a distraction driver frequently engage in.

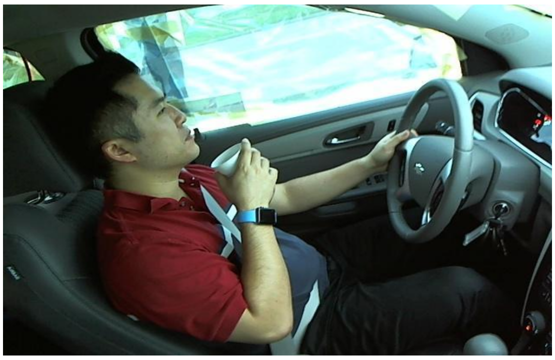


Figure 4.8. "Drinking" Class

- Class 8: Reaching behind,

  In this category the driver turns his back towards the steering wheel as shown in Figure 4.9 and reaches the back seat of the car while driving.

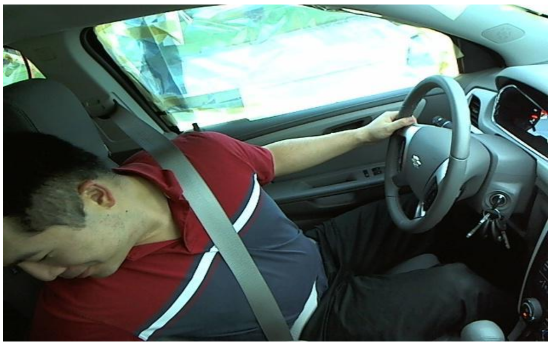


Figure 4.9. "Reaching behind" Class

- Class 9: Dressing the hair and makeup activities, and
  Applying makeup is another category that has been included. This act is shown in Figure 4.10.

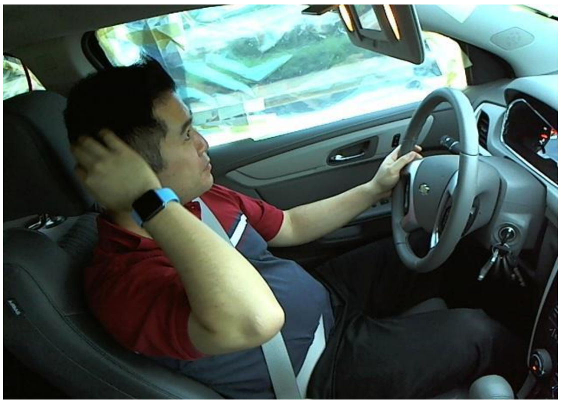


Figure 4.10. "Dressing the hair and makeup activities" Class

- Class 10: Talking to passengers
  In this category again the driver turns his gaze away from the windshield and looks towards his fellow passenger as shown in Figure 4.11. Taking his eyes off the road classifies this action as a distracting task.

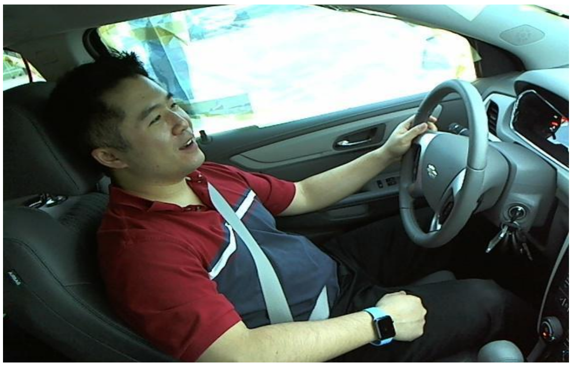


Figure 4.11. "Talking to passengers" Class

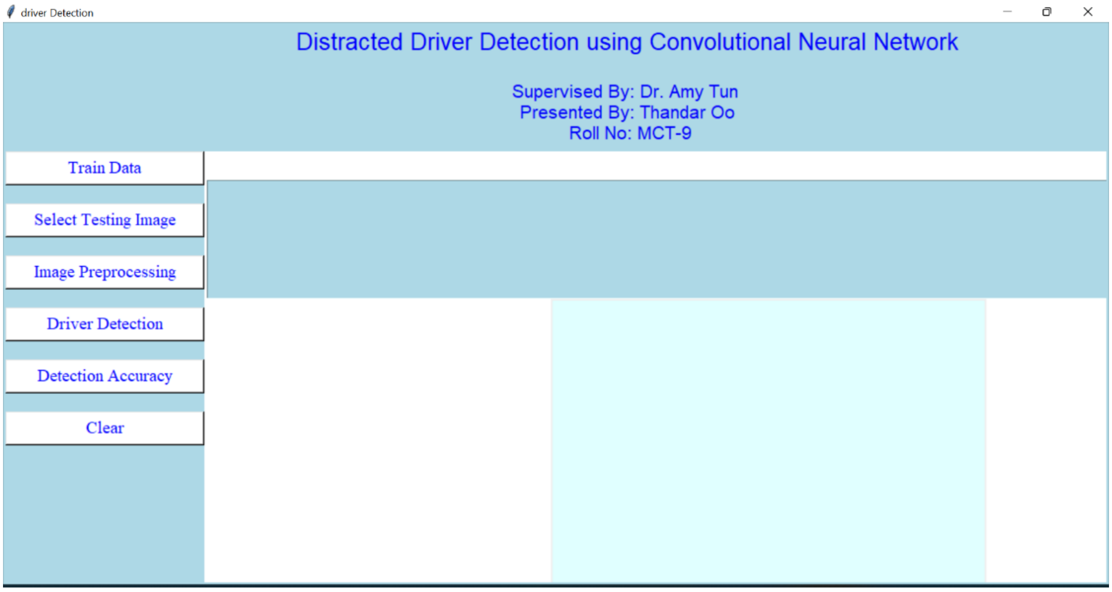
The chosen classes are frequently a source of diversion for driving and even cause accidents and fatalities, and loss of property. In actuality, some of these tasks, like placing a call, are illegal in several nations while driving, and a person can be arrested if found.

4.4 Implementation of the Distracted Driver Detection System

The proposed system is implemented by using Python programming language. In the home page of the system, there are six buttons. These are as follows:

- “Train Data”,
- “Select Testing Image”,
- “Image Preprocessing”,
- “Driver Detection”,
- “Detection Accuracy” and
- “Clear” button.

Home page of the system is shown in Figure 4.12.



driver Detection

Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

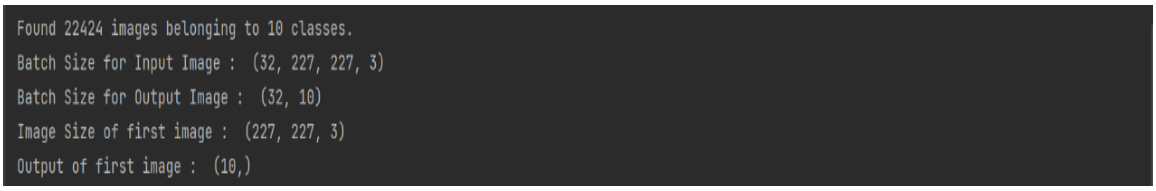
Figure 4.12. Home Page of the System

4.4.1. Training Process

To train data, the user must use “Train Data” button from the Homepage of the system. In this process, this system trained “22,424” images belonging to 10 classes. These classes are “Driving safely”, “Texting with the right hand”, “Talking on the phone with the right hand”, “Texting with the left hand”, “Talking on the phone with the left hand”, “Operating the radio”, “Drinking”, “Reaching behind”, “Dressing the hair and makeup activities”, and “Talking to passengers”.

4.4.2. Image Preprocessing

In the image preprocessing phase, this system performs the “rescaling”, “resizing” and “normalization” processes. Image preprocessing step is shown in Figure 4.13. In this sample, the batch size for input image before pre-processing step is “32, 227, 227, 3”. But, after finishing pre-processing step, this system produces the normalized image that has the “32, 10” batch size.

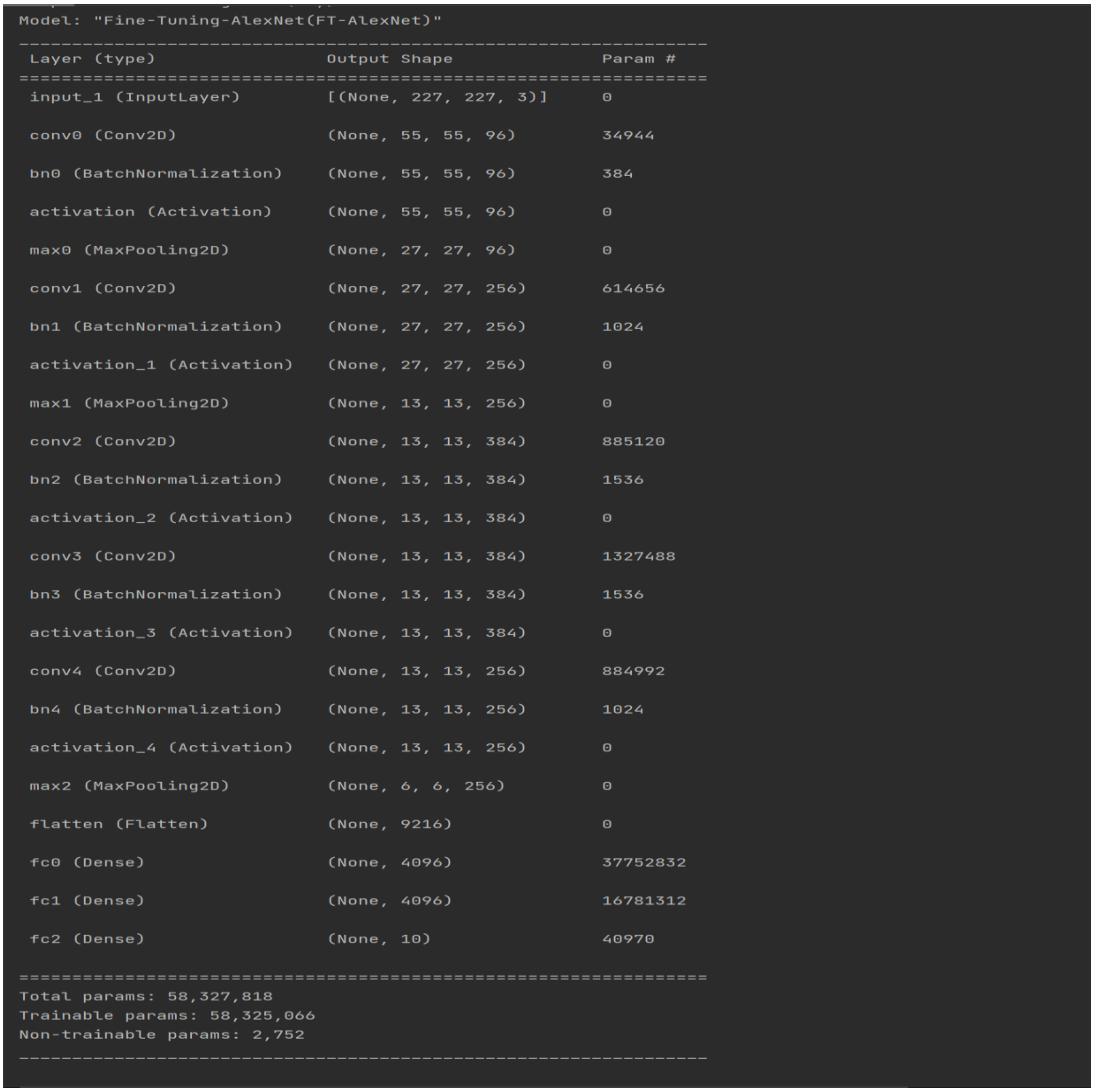


```
Found 22424 images belonging to 10 classes.
Batch Size for Input Image :  (32, 227, 227, 3)
Batch Size for Output Image :  (32, 10)
Image Size of first image :  (227, 227, 3)
Output of first image :  (10,)
```

Figure 4.13. Image Preprocessing for Training Data

4.4.3. Fine-Tuning-AlexNet CNN Architecture

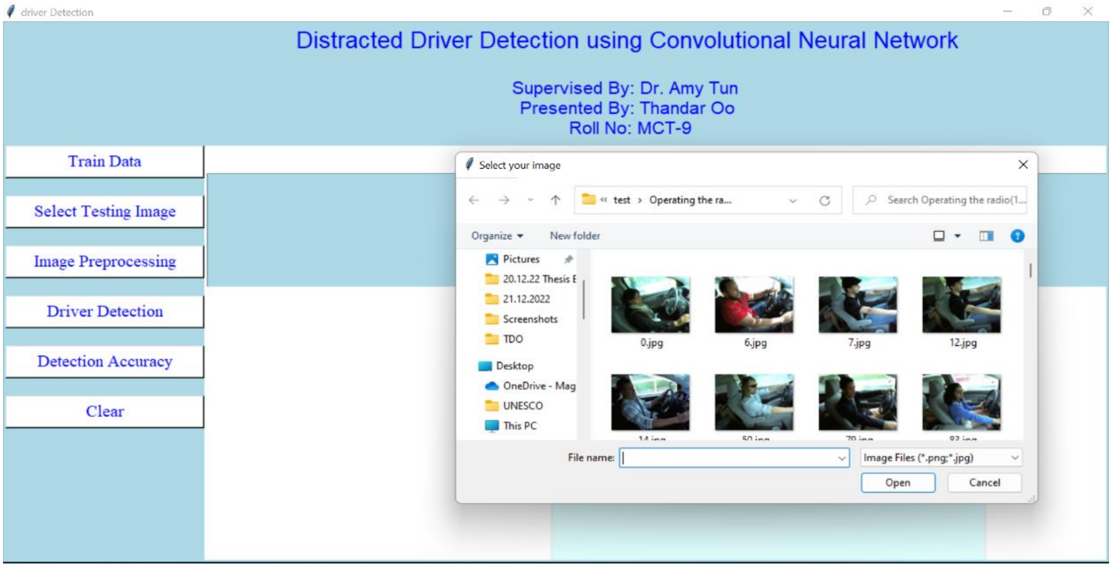
This system trains the “22,424” distracted driver images by using Fine-Tuning-AlexNet CNN model. This model architecture is shown in Figure 4.14.



```
Model: "Fine-Tuning-AlexNet(FT-AlexNet)"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 227, 227, 3)]     0

 conv0 (Conv2D)              (None, 55, 55, 96)        34944

 bn0 (BatchNormalization)    (None, 55, 55, 96)        384

 activation (Activation)     (None, 55, 55, 96)        0

 max0 (MaxPooling2D)         (None, 27, 27, 96)        0

 conv1 (Conv2D)              (None, 27, 27, 256)       614656

 bn1 (BatchNormalization)    (None, 27, 27, 256)       1024

 activation_1 (Activation)   (None, 27, 27, 256)       0

 max1 (MaxPooling2D)         (None, 13, 13, 256)       0

 conv2 (Conv2D)              (None, 13, 13, 384)       885120

 bn2 (BatchNormalization)    (None, 13, 13, 384)       1536

 activation_2 (Activation)   (None, 13, 13, 384)       0

 conv3 (Conv2D)              (None, 13, 13, 384)       1327488

 bn3 (BatchNormalization)    (None, 13, 13, 384)       1536

 activation_3 (Activation)   (None, 13, 13, 384)       0

 conv4 (Conv2D)              (None, 13, 13, 256)       884992

 bn4 (BatchNormalization)    (None, 13, 13, 256)       1024

 activation_4 (Activation)   (None, 13, 13, 256)       0

 max2 (MaxPooling2D)         (None, 6, 6, 256)         0

 flatten (Flatten)           (None, 9216)              0

 fc0 (Dense)                 (None, 4096)              37752832

 fc1 (Dense)                 (None, 4096)              16781312

 fc2 (Dense)                 (None, 10)                40970

=================================================================
Total params: 58,327,818
Trainable params: 58,325,066
Non-trainable params: 2,752
_________________________________________________________________
```

Figure 4.14. Fine-Tuning-AlexNet CNN Architecture

4.4.4. Testing Process


driver Detection

Distracted Driver Detection using Convolutional Neural Network

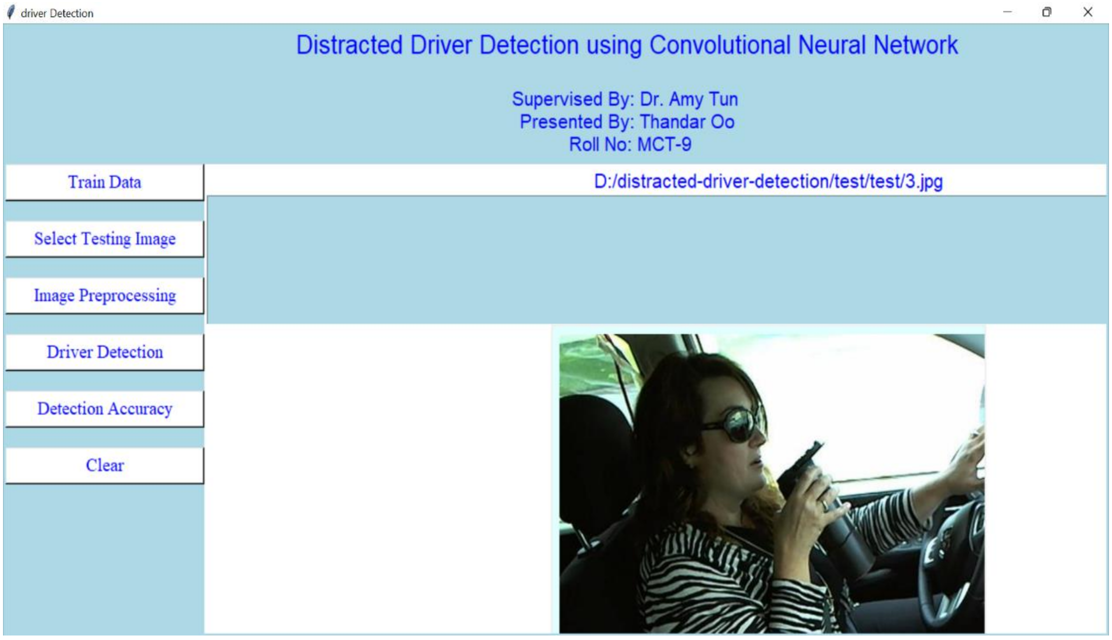
Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

Select your image

File name:
Image Files (*.png;*.jpg)

Figure 4.15. Test Image Selection

For classification, the user must choose randomly one testing image from testing dataset. To test image, the user must first use the "Select Testing Image" button. Test image selection process is shown in Figure 4.15. After finishing "test image selection process", this system shows the test image and the path of this image (D:/distracted…). The test image is shown in Figure 4.16.


driver Detection

Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

D:/distracted-driver-detection/test/test/3.jpg

Figure 4.16. Test Image

4.4.5. Normalized Testing Image

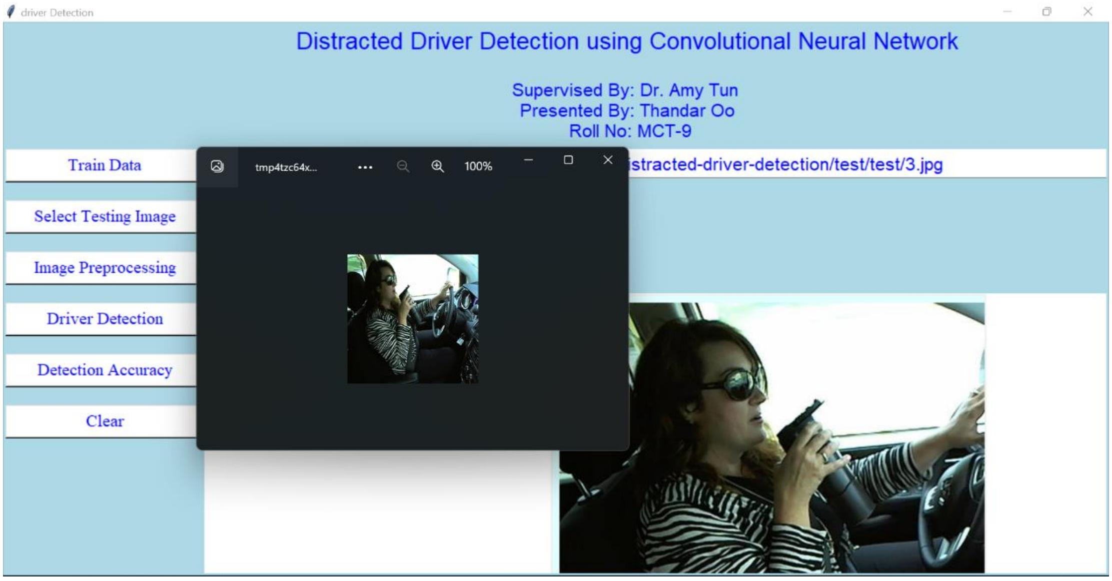
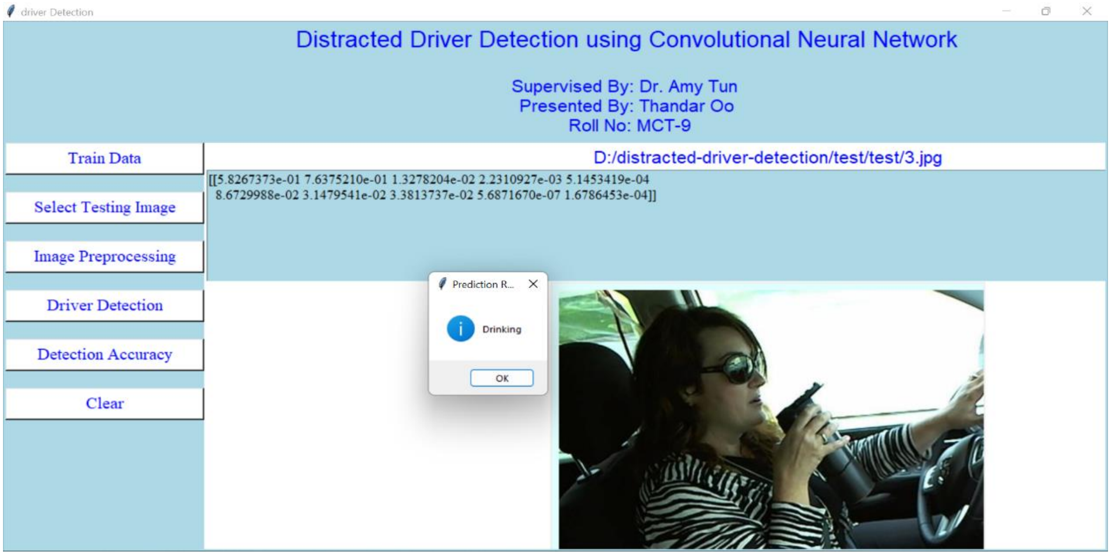


Figure 4.17. Normalized Testing Image

The normalized testing image is shown in Figure 4.17. In the testing phase, the user must use the “image preprocessing” button for resizing and normalization. Image preprocessing is the important step for every classification system.

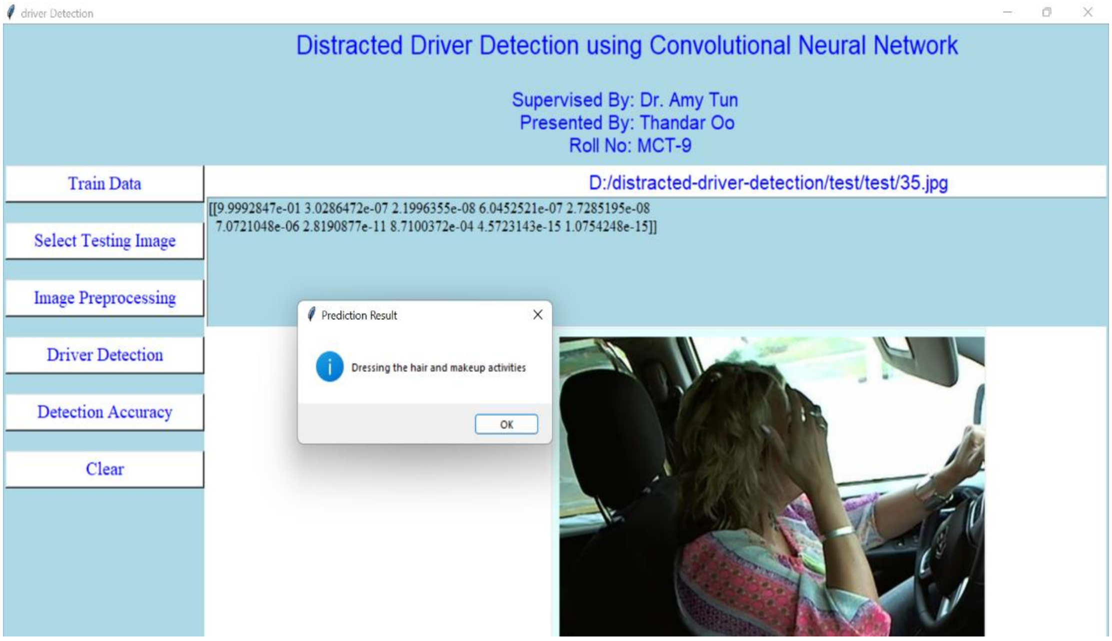
4.4.6. Distracted Driver Detection and Classification

By using the FT-AlexNet CNN model, this system detects driver and classifies this driver into the distracted driver class. Classification for “Drinking” distracted driver is shown in Figure 4.18.



```
[[5.8267373e-01 7.6375210e-01 1.3278204e-02 2.2310927e-03 5.1453419e-04
  8.6729988e-02 3.1479541e-02 3.3813737e-02 5.6871670e-07 1.6786453e-04]]
```

Figure 4.18. Classification for “Drinking” Distracted Driver



Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

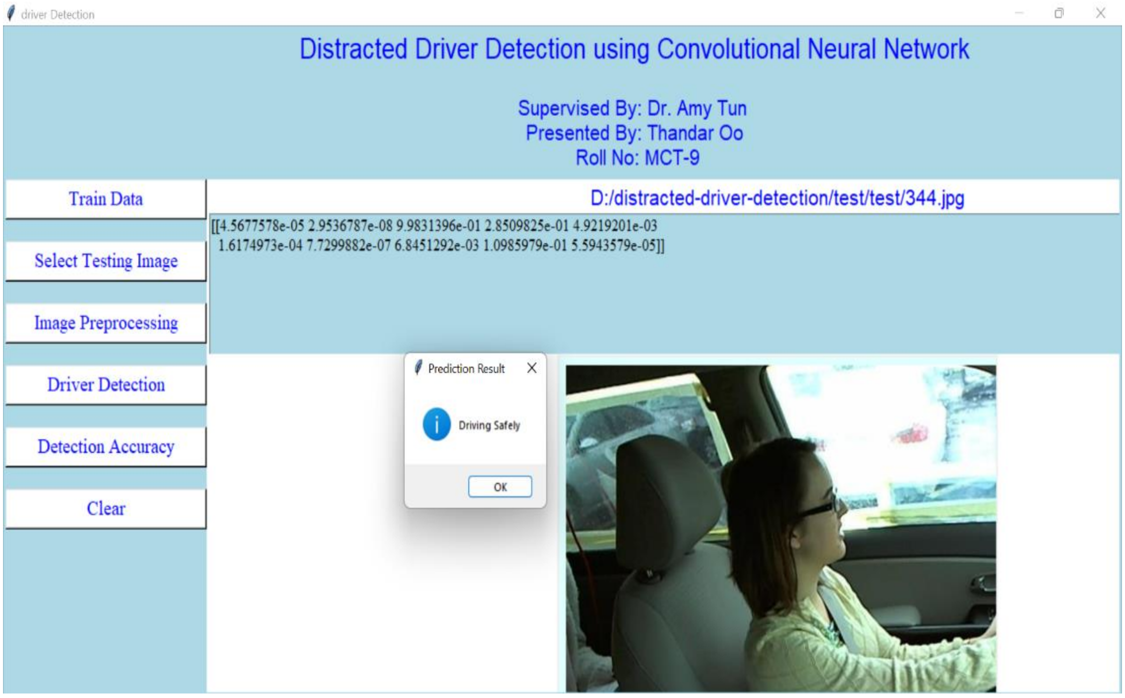
D:/distracted-driver-detection/test/test/35.jpg

```
[[9.9992847e-01 3.0286472e-07 2.1996355e-08 6.0452521e-07 2.7285195e-08
  7.0721048e-06 2.8190877e-11 8.7100372e-04 4.5723143e-15 1.0754248e-15]]
```

Dressing the hair and makeup activities

Figure 4.19. Classification for “Dressing the hair and makeup activities” Distracted Driver

Classification for “Dressing the hair and makeup activities” distracted driver is shown in Figure 4.19 and classification for “Driving safely” driver class is shown in Figure 4.20.



Distracted Driver Detection using Convolutional Neural Network

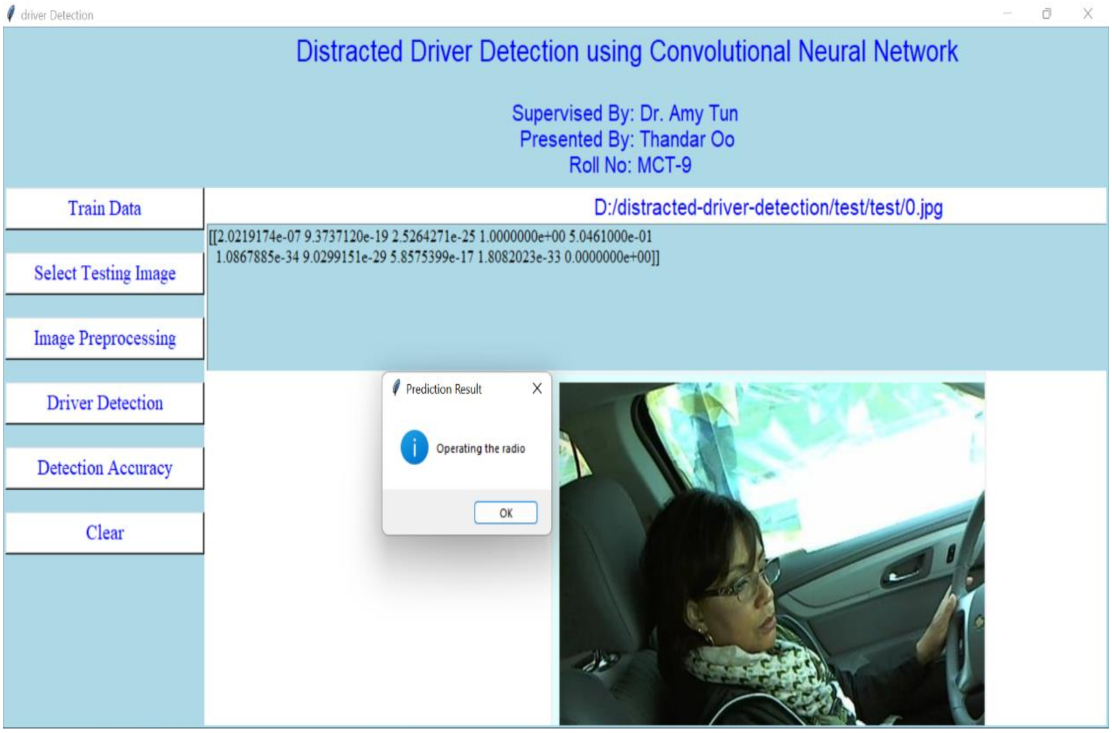
Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

D:/distracted-driver-detection/test/test/344.jpg

```
[[4.5677578e-05 2.9536787e-08 9.9831396e-01 2.8509825e-01 4.9219201e-03
  1.6174973e-04 7.7299882e-07 6.8451292e-03 1.0985979e-01 5.5943579e-05]]
```

Driving Safely

Figure 4.20. Classification for “Driving Safely” Driver



Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

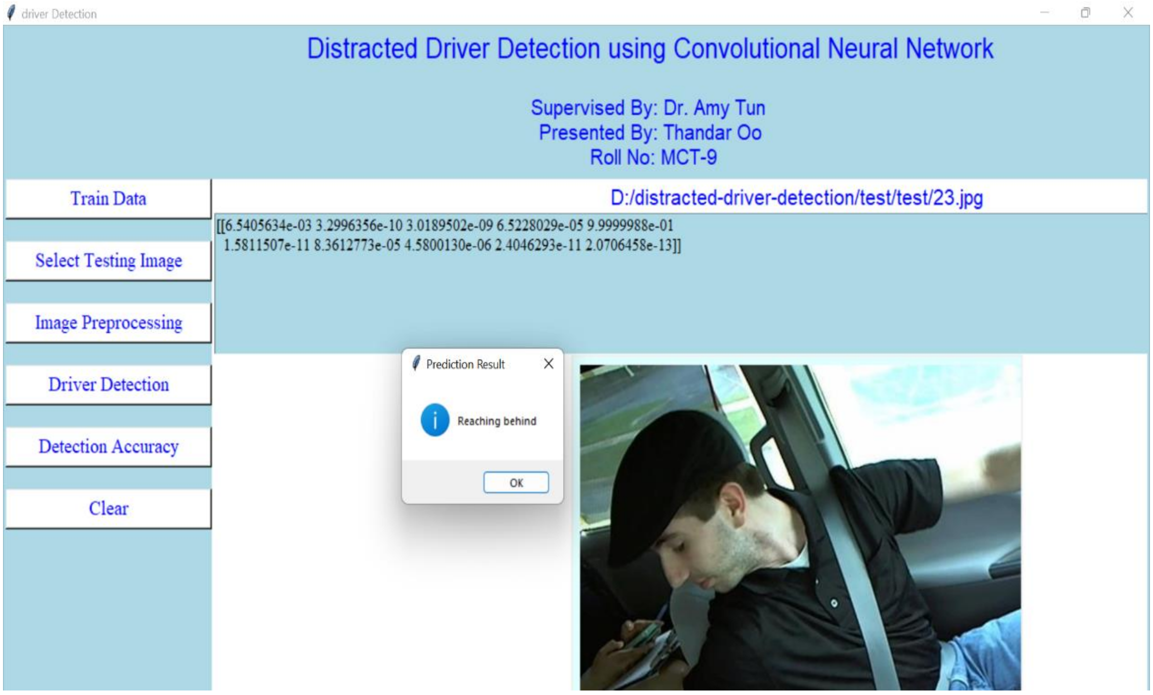
D:/distracted-driver-detection/test/test/0.jpg

```
[[2.0219174e-07 9.3737120e-19 2.5264271e-25 1.0000000e+00 5.0461000e-01
  1.0867885e-34 9.0299151e-29 5.8575399e-17 1.8082023e-33 0.0000000e+00]]
```

Operating the radio

Figure 4.21. Classification for "Operating the Radio" Distracted Driver

Classification for "Operating the radio" distracted driver is shown in Figure 4.21 and classification for "Reaching behind" distracted driver is shown in Figure 4.22.



Distracted Driver Detection using Convolutional Neural Network

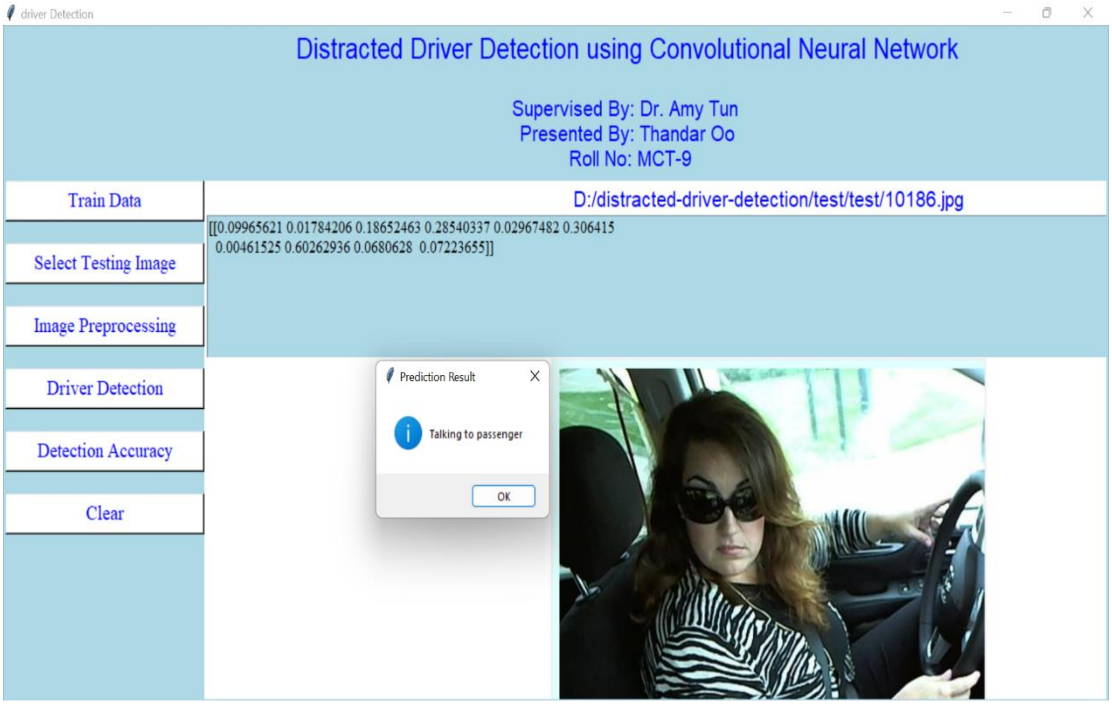
Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

D:/distracted-driver-detection/test/test/23.jpg

```
[[6.5405634e-03 3.2996356e-10 3.0189502e-09 6.5228029e-05 9.9999988e-01
  1.5811507e-11 8.3612773e-05 4.5800130e-06 2.4046293e-11 2.0706458e-13]]
```

Reaching behind

Figure 4.22. Classification for "Reaching Behind" Distracted Driver



driver Detection

Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

D:/distracted-driver-detection/test/test/10186.jpg

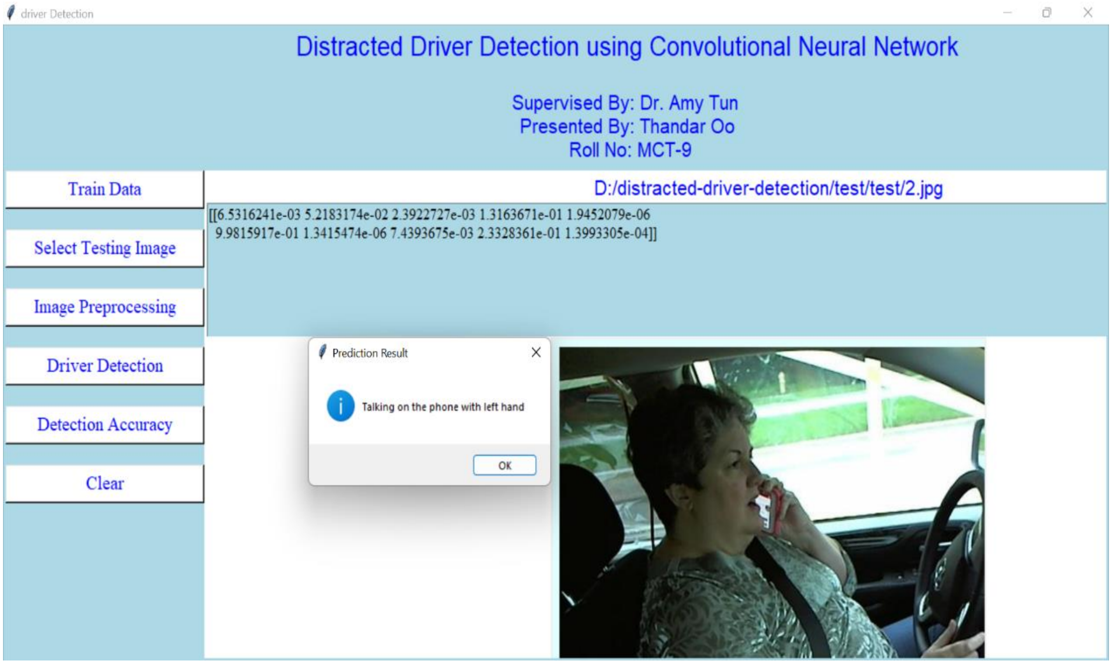
```
[[0.09965621 0.01784206 0.18652463 0.28540337 0.02967482 0.306415
  0.00461525 0.60262936 0.0680628  0.07223655]]
```

Prediction Result

Talking to passenger

Figure 4.23. Classification for “Talking to Passenger” Distracted Driver

Classification for “Talking to passenger” distracted driver is shown in Figure 4.23 and classification for “Talking on the phone with left hand” distracted driver is shown in Figure 4.24.



driver Detection

Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

D:/distracted-driver-detection/test/test/2.jpg

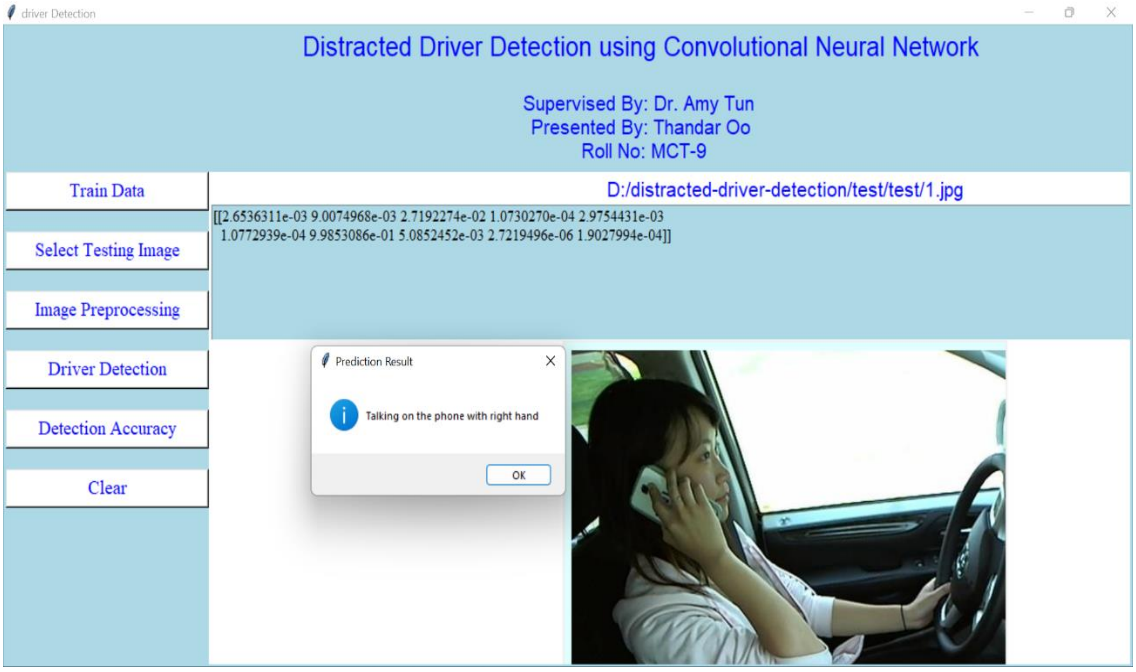
```
[[6.5316241e-03 5.2183174e-02 2.3922727e-03 1.3163671e-01 1.9452079e-06
  9.9815917e-01 1.3415474e-06 7.4393675e-03 2.3328361e-01 1.3993305e-04]]
```

Prediction Result

Talking on the phone with left hand

Figure 4.24. Classification for “Talking on the Phone with Left Hand” Distracted Driver

Classification for “Talking on the phone with right hand” distracted driver is shown in Figure 4.25 and classification for “Texting with left hand” distracted driver is shown in Figure 4.26.



driver Detection

Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

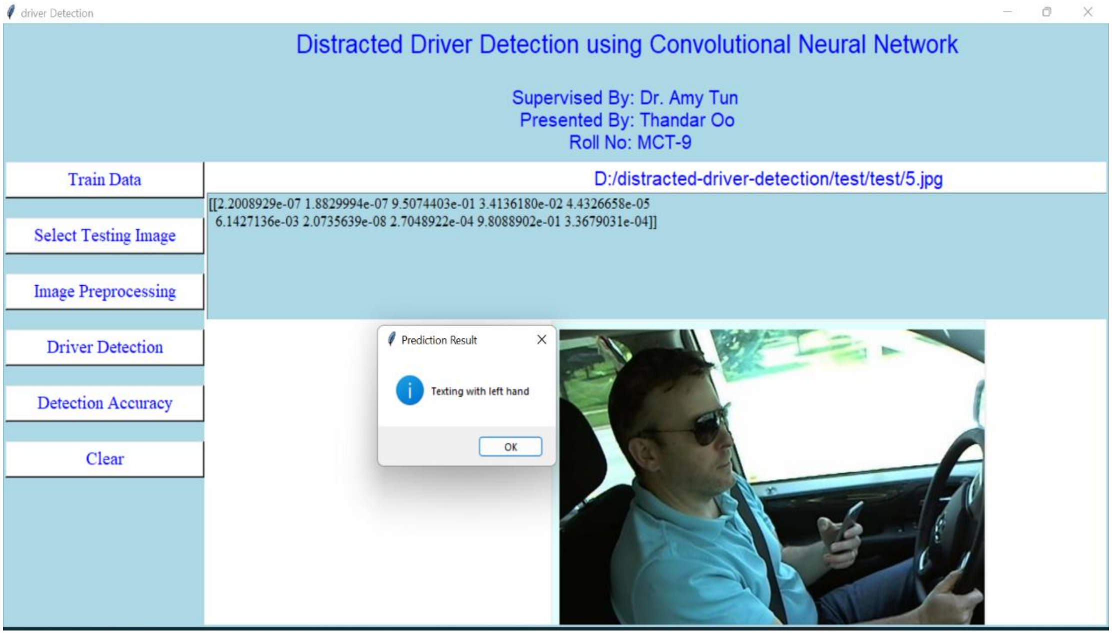
D:/distracted-driver-detection/test/test/1.jpg

```
[[2.6536311e-03 9.0074968e-03 2.7192274e-02 1.0730270e-04 2.9754431e-03
  1.0772939e-04 9.9853086e-01 5.0852452e-03 2.7219496e-06 1.9027994e-04]]
```

Prediction Result

Talking on the phone with right hand

Figure 4.25. Classification for “Talking on the Phone with Right Hand” Distracted Driver



driver Detection

Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

D:/distracted-driver-detection/test/test/5.jpg

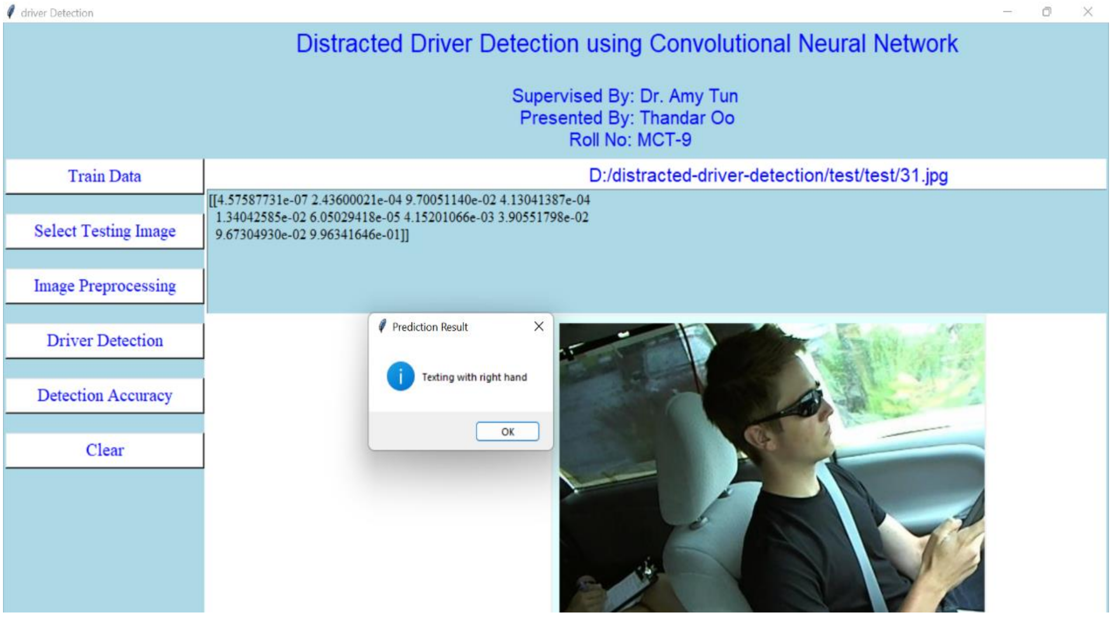
```
[[2.2008929e-07 1.8829994e-07 9.5074403e-01 3.4136180e-02 4.4326658e-05
  6.1427136e-03 2.0735639e-08 2.7048922e-04 9.8088902e-01 3.3679031e-04]]
```

Prediction Result

Texting with left hand

Figure 4.26. Classification for “Texting with Left Hand” Distracted Driver

Classification for “Texting with right hand” distracted driver is shown in Figure 4.27.



Distracted Driver Detection using Convolutional Neural Network

Supervised By: Dr. Amy Tun
Presented By: Thandar Oo
Roll No: MCT-9

D:/distracted-driver-detection/test/test/31.jpg

```
[[4.57587731e-07 2.43600021e-04 9.70051140e-02 4.13041387e-04
  1.34042585e-02 6.05029418e-05 4.15201066e-03 3.90551798e-02
  9.67304930e-02 9.96341646e-01]]
```

Texting with right hand

Figure 4.27. Classification for “Texting with Right Hand” Distracted Driver

4.5 Experimental Results

In this system, accuracy, precision, recall, f-score and error rate methods are used to measure the performance of the system. By testing the prediction model that are trained using 22,424 images, this system obtains the “0.9984” correctness and the “0.0016” loss. Table 4.1 shows the performance evaluation result of the system. An experimental result of the system is shown in Figure 4.28.

Table 4.1 Performance Evaluation Result of the System

Name	Results of the System
Precision	0.8851
Recall	0.9617
Error Rate	0.0016
Accuracy	0.9984
F-score	0.9216

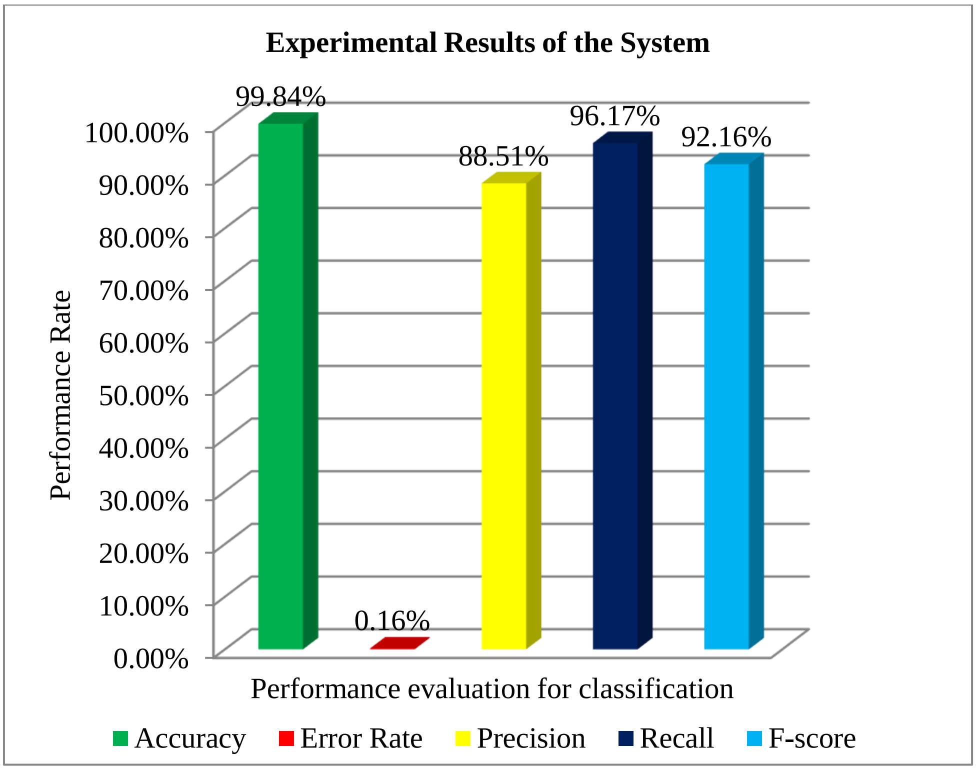


Figure 4.28. Experimental Results of the System

Table 4.2 shows the performance evaluation results such as the accuracy, error rate, precision, recall and f-score results about each driver's actions that include 10 classes.

Figure 4.29 shows the accuracy results about 10 classes. According to the accuracy result, this system highest correctly classifies about "operating the radio" class. But, this system produces the lowest accuracy result by classifying "Talking to passengers" class. Therefore, this "Talking to passengers" class has the highest error rate result among the other nine classes. Figure 4.30 shows the error rate results about 10 classes.

According to the performance evaluation results, the "Drinking" class has the highest precision and recall results. Figure 4.31 and 4.32 shows the precision and recall results about 10 classes. By measuring the performance of the system according to the F-score method, this system obtains the lowest F-score results about "reaching behind" class and the highest F-score results about "drinking" class. Figure 4.33 shows the F-score results about 10 classes.

According to the performance evaluation results, the "Drinking" class has the highest precision and recall results. Figure 4.31 and 4.32 shows the precision and recall

results about 10 classes. By measuring the performance of the system according to the F-score method, this system obtains the lowest F-score results about "reaching behind" class and the highest F-score results about "drinking" class. Figure 4.33 shows the F-score results about 10 classes.

Table 4.2 Performance Evaluation Results about 10 Classes

Class ID	Class Name	Accuracy	Error Rate	Precision	Recall	F-Score
Class 1	Driving safely	98	2	89	98	93.28
Class 2	Texting with the right hand	94	6	88	96	91.8
Class 3	Talking on the phone with right hand	98	2	92	98	94.9
Class 4	Texting with the left hand	95	5	90	95	92.4
Class 5	Talking on the phone with left hand	98	2	89.1	98	93.38
Class 6	Operating the radio	98.5	1.5	88	97	92.28
Class 7	Drinking	97.5	2.5	92	98.7	95.2
Class 8	Reaching behind	90	10	79	90	84.14
Class 9	Dressing hair and makeup activities	98	2	91	98	94.37
Class 10	Talking to passengers	87	13	87	93	89.9

According to the performance evaluation results, the "Drinking" class has the highest precision and recall results. Figure 4.31 and 4.32 shows the precision and recall results about 10 classes. By measuring the performance of the system according to the F-score method, this system obtains the lowest F-score results about "reaching behind" class and the highest F-score results about the "drinking" class. Figure 4.33 shows the F-score results about 10 classes.

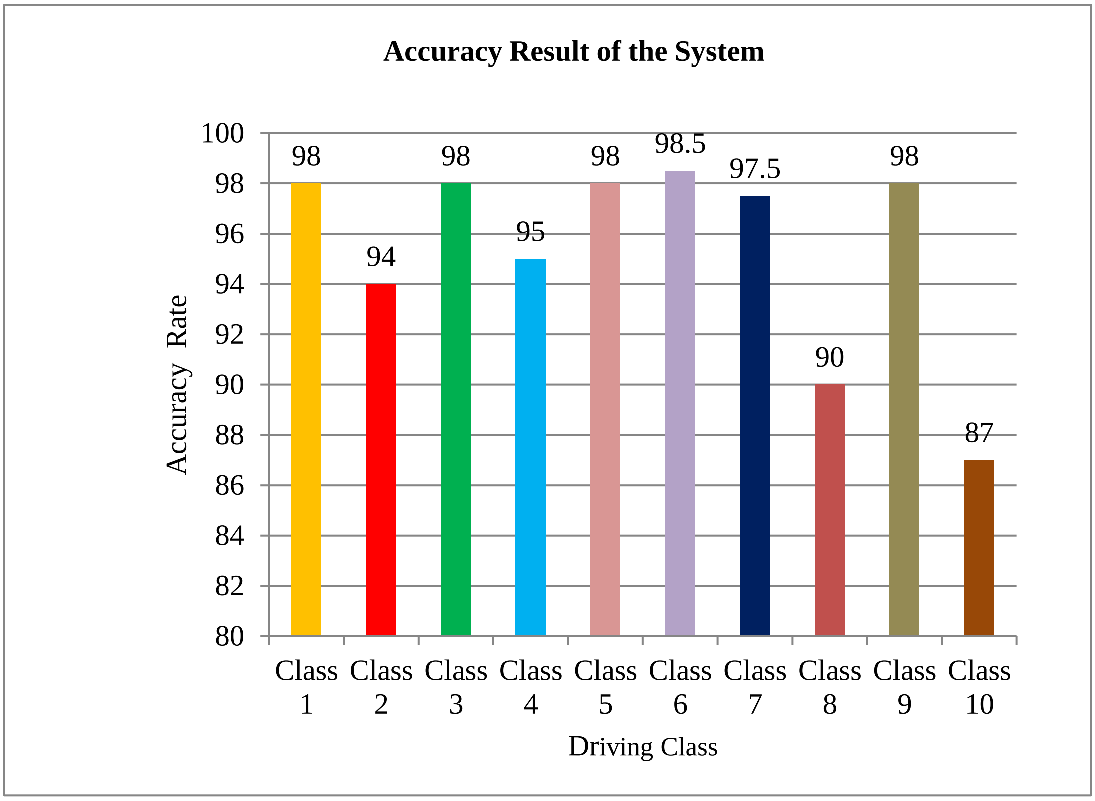


Figure 4.29. Accuracy Results

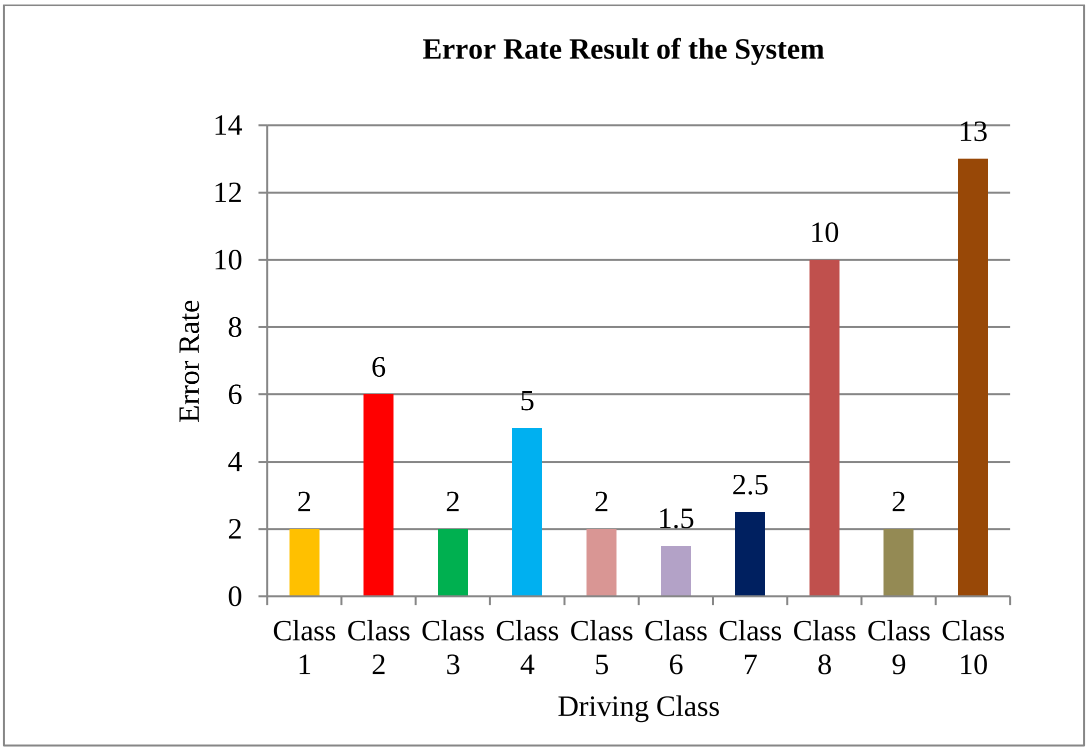


Figure 4.30. Error Rate Results

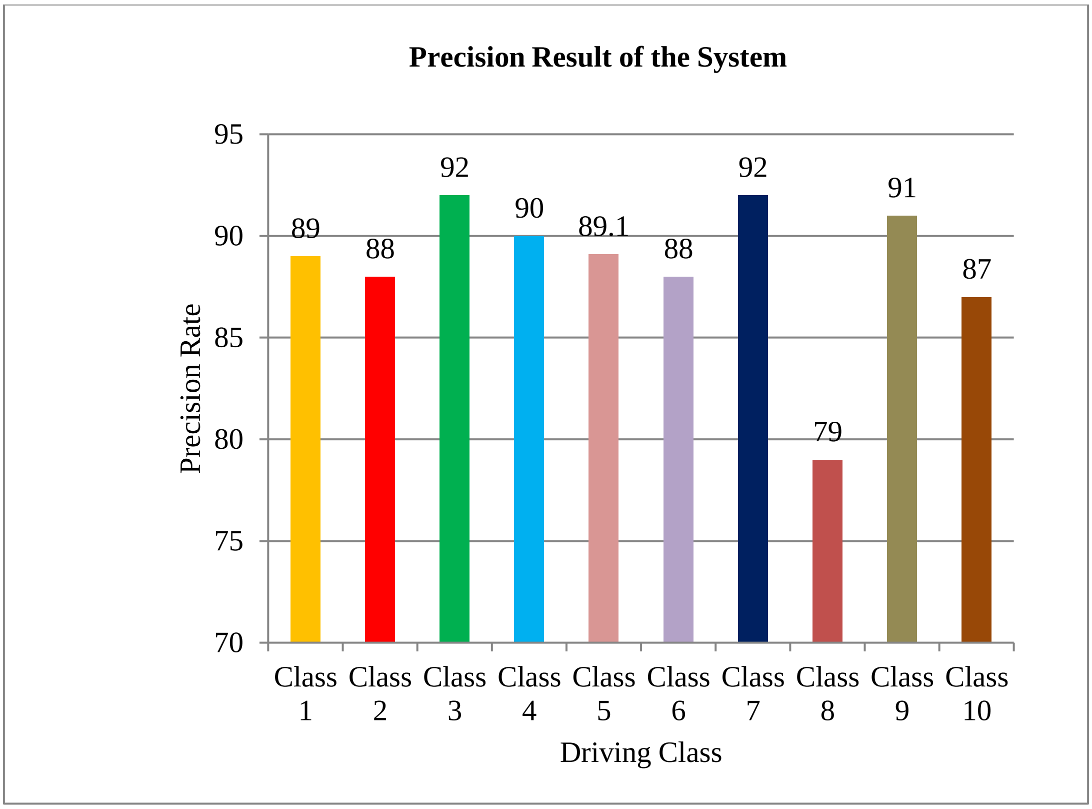


Figure 4.31. Precision Results

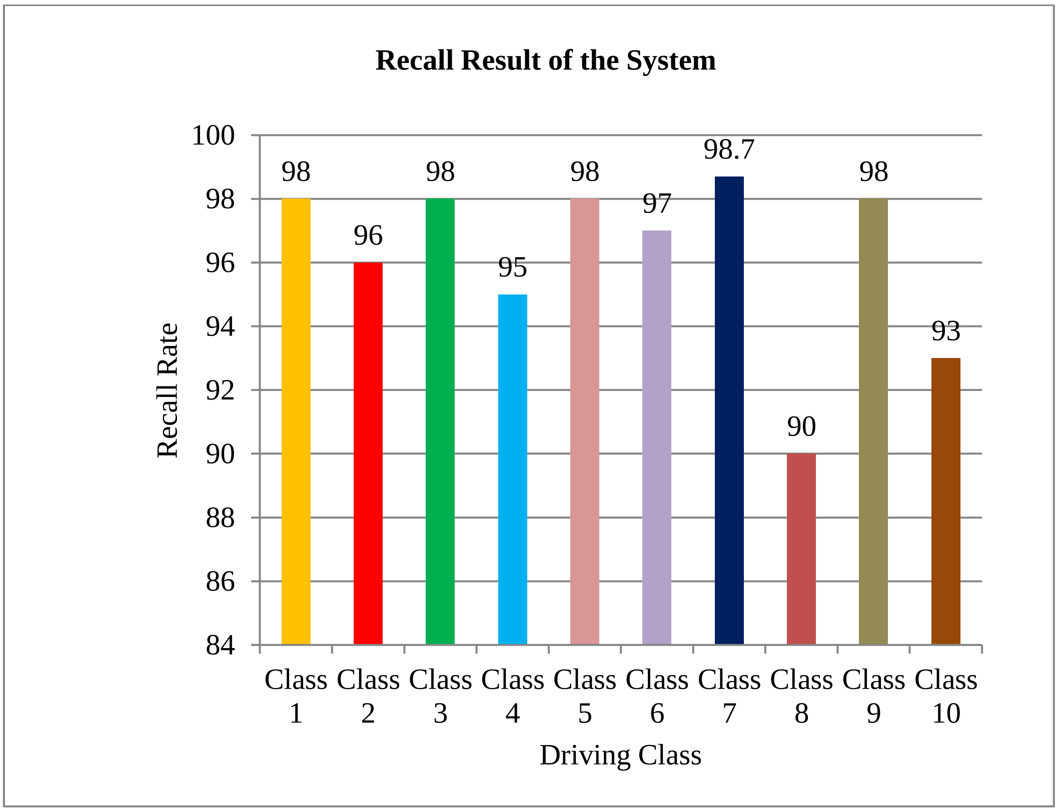


Figure 4.32. Recall Results

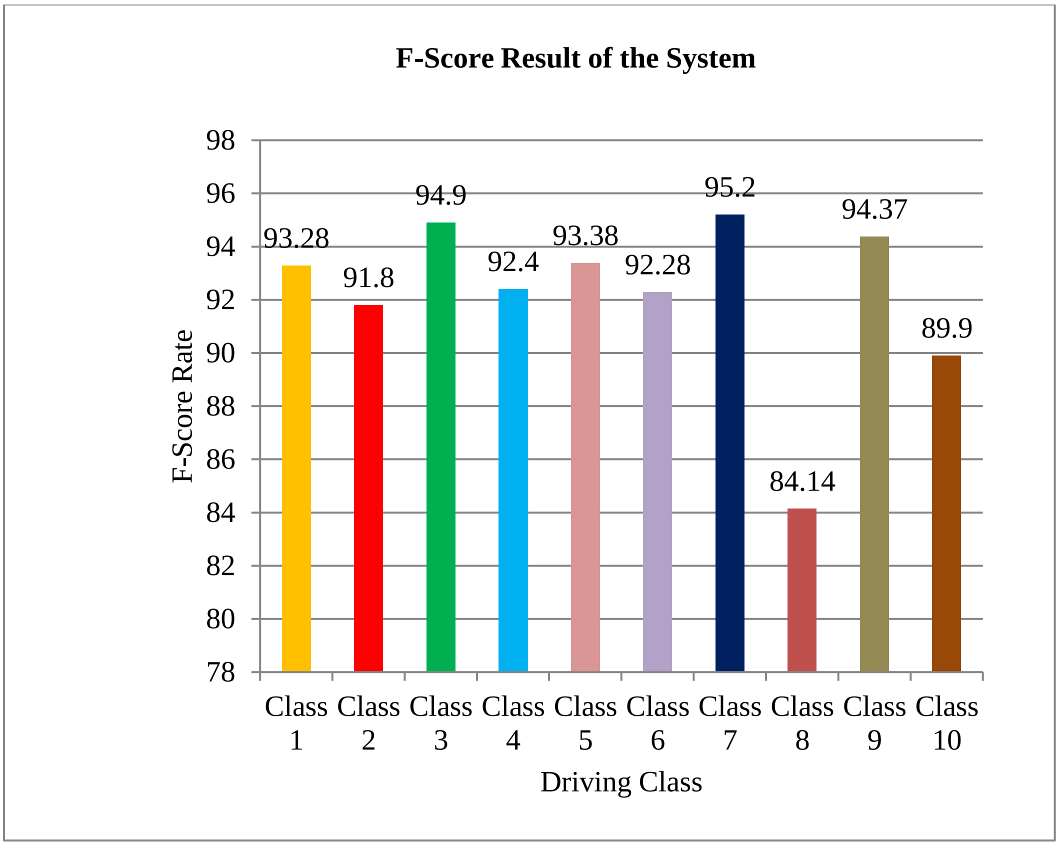


Figure 4.33. F-Score Results about 10 Classes

CHAPTER 5
CONCLUSION AND FURTHER EXTENSION

In recent years, the number of road accidents, due to distracted driving, has been on a rise. This makes it imperative to take measures to curb the number of road fatalities. The major cause of these accidents is driver error.

Distracted driving is a dangerous act, and one of the prime contributing factors to road traffic accidents. The number of incidents involving vehicles is steadily rising, and one of the main causes of this is driver distraction. Distractions for drivers can include talking on the phone, texting while talking on the phone, eating, drinking, and conversing with other passengers. In the current age, the technology is significantly improved and deep learning approaches have played an important role. Deep neural networks are widely used for image classification, object detection and many recognitions system.

5.1 Advantages

To prevent accidents, this system is proposed as the distracted driver detection system by using the Fine-Tuning-Alexnet CNN architecture that solves the issue of road accidents because of driver distractions. This system classified the image of the driver's behaviors, with higher performance. In addition, it may also be used for other image classification areas. The StateFarm dataset was created to aid in such research and is accessible to the general public on Kaggle. Each activity in this dataset is categorized, and its associated photos are listed separately. This dataset has a total of 10 actions (classes).

The dataset contains training and testing data for 26 different drivers. These images were applied to deep learning to learn the image features and train the system with FT-AlexNet CNN model. The proposed FT-AlexNet CNN model achieves an accuracy of 99.84% and loss of 0.0051%.

5.2 Limitations and Further Extension

The current approach is effective, however, in the future, the researcher can plan to generate own dataset with using this practice to attempt again and use the more extensive techniques on a greater size. Although the accuracy of the proposed system is 99% and loss is 0.16% for most images of the dataset, the accuracy for the images of two driving classes (talking to passenger class and reaching behind class) is less than for other classes. The existing process should be improved upon and customized. Future work is also anticipated to entail the creation of real-time driver attention monitoring technology and the use of wireless means to enforce citations for driving while distracted are issued to drivers. The researcher can develop a system like that would notice the distracted driver and then text the driver's phone with a traffic ticket for the infraction.

REFERENCES

[1] A. Anand, "Image Classification", ResearchGate, 2017.

[2] A. Jahn, "Keras Image Preprocessing: Scaling Image Pixels for Training", Senior Software Developer Fullstack/Cloud, Linkedin, 2017.

[3] B. Nikhil, "Image Data Pre-processing for Neural Networks", Chatbot Conference, 2017.

[4] Bhuvaneshwari and E. G. M. Kanaga, "Convolutional Neural Network for Addiction Detection using Improved Activation Function", 5th International Conference on Computing Methodologies and Communication, IEEE, 2022.

[5] F. Sajid, A. R. Javed and A. Basharat, "An Efficient Deep Learning Framework for Distracted Driver Detection", IEEE, 2021.

[6] H. Tang, "Image Classification based on CNN: Models and Modules", International Conference on Big Data, Information and Computer Network, IEEE, 2022.

[7] https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/

[8] https://docs.roboflow.com/image-transformations/image-preprocessing

[9] https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/

[10] https://www.kaggle.com/c/state-farm-distracted-driver-detection/

[11] https://www.techscience.com/cmc/v68n1/41832/html

[12] https://www.worldlifeexpectancy.com/myanmar-road-traffic-accidents/

[13] J. M. Mase, P. Chapman and P. F. Grazziela, "A Hybrid Deep Learning Approach for Driver Distraction Detection", International Conference on Information and Communication Technology Convergence, IEEE, 2020.

[14] J. R. Rekkala, "Mobile Usage Detection of Driver using CNN (Convolutional Neural Network), California State University, Northridge, December, 2021.

[15] M. Aljasim and R. Kashef, “E2DR: A Deep Learning Ensemble-Based Driver Distraction Detection with Recommendations Model”, MDPI Journal, vol. 22, no 5, 2022.

[16] M. H. Alkinani and W. Z. Khan, “Detecting Human Driver Inattentive and Aggressive Driving Behavior Using Deep Learning: Recent Advances, Requirements and Open Challenges”, IEEE, 2020.

[17] M. U. Hossain, M. A. Rahman, M. M. Islam, A. Akhter, M. A. Uddin, B. K. Paul, “Automatic driver distraction detection using deep convolutional neural networks”, Elsevier, vol. 14, 2022.

[18] P. M. Chawan, S. Satardekar and D. Shah, "Distracted Driver Detection and Classification", International Journal of Engineering Research and Application, vol. 8, no. 4, pp. 60-64, 2018.

[19] R. Mangayarkarasi, C. Vanmathi and M. Vishwakarma, "A Compartive Study on Driver Distraction Detection using a Deep Learning Model", International Conference on Intelligent Computing Instrumentation and Control Technologies, IEEE, 2022.

[20] R. Zeng and Y. Zhang, "Comparative Analysis on Different Convolutional Neural Network (CNN) for Classification", IEEE 5th International Conference on Information Systems and Computer Aided Education, 2022.

[21] S. Masood, A. Rai and A. Aggarwal, "Detecting Distraction of Drivers using Convolutional Neural Network", Pattern Recognition, Elsevier, 2017.

[22] S. Perumal and T. Velmurugan, "Preprocessing by Contrast Enhancement Techniques for Medical Images", International Journal of Pure and Applied Mathematics, vol. 118, no. 18, pp. 1314-3395, 2018.

[23] T. H. N. Le, Y. Zheng, C. Zhu, K. Luu and M. Savvides, “Multiple Scale Faster-RCNN Approach to Driver’s Cell-phone Usage and Hands on Steering Wheel Detection”, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016.

[24] V. Abdul Jamsheed, B. Janet and U. S. Reddy, "Real Time Detection of Driver Distraction using CNN", Third International Conference on Smart Systems and Inventive Technology, IEEE, 2020.

[25] W. Hao and W. Yizhou, "The Role of Activation Function in CNN", International Conference on Information Technology and Computer Application (ITCA), IEEE, 2020.

PUBLICATIONS

[1] Thandar Oo and Amy Tun, “Distracted Driver Detection based on Convolutional Neural Network”, University of Computer Studies, Yangon, Myanmar, 2022.