REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

EMBEDDED SYSTEM DESIGN FOR DRIVE SECURITY

Kubilay ATAŞ

MASTER OF SCIENCE THESIS
Department of Electronics and Communication Engineering
Program of Electronics

Supervisor
Assoc. Prof. Dr. Revna ACAR VURAL

June, 2022

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

EMBEDDED SYSTEM DESIGN FOR DRIVE SECURITY

A thesis submitted by Kubilay ATAŞ in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 30.06.2022 in Department of Electronics and Communication Engineering, Program of Electronics.

Assoc. Prof. Dr. Revna ACAR VURAL
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Revna ACAR VURAL, Supervisor
Yildiz Technical University _____

Prof. Dr. Burcu ERKMEN, Member
Yildiz Technical University _____

Asst. Prof. Dr. İpek ŞEN, Member
İstanbul Bilgi University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Embedded System Design for Drive Security supervised by my supervisor, Assoc. Prof. Dr. Revna ACAR VURAL. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Kubilay ATAŞ

Signature

To my dear wife
and my altruist family

ACKNOWLEDGEMENTS

I am grateful to have received my education at Yildiz Technical University, which is infused with a modern awareness of educational practices and staffed with eminent scientists who contribute to academic advancement by keeping abreast of developments in the relevant scientific fields. I would want to express my gratitude to each and every academician who has assisted me throughout the course of my graduate studies.

I would like to take this opportunity to thank the dedicated administrative team at the Graduate School of Natural and Applied Sciences as well as the Faculty of Electrical and Electronics for all of the support they have provided. I am thankful to my supervisor, Assoc. Prof. Dr. Revna ACAR VURAL, who helped me take my first steps into the academic world and guided me in many different ways. Her support and guidance have been invaluable to me. I would like to thank my thesis jury, who listened attentively to my presentations and greatly contributed to my work. I would also like to thank my esteemed professor Prof. Dr. Tülay YILDIRIM, who sparked my interest in machine learning, and my dear friend Kıvanç GÜÇKIRAN, who shared his knowledge on technical studies in this field and was always willing to lend a helping hand. Because of his assistance in the data collecting, I am indebted to my friend Arda TEZCAN. Moreover, I am thankful that I was able to make some of my closest friends, Mete BERBEROĞLU and İsmail GÜZELTEPE, when I was pursuing my Master's degree.

Finally, I would like to express my appreciation to my loyal parents, who have always been there for me, as well as to my wife, Ceren ZENGİN ATAŞ, who has made a significant contribution to my work by helping me to label images.

Kubilay ATAŞ

TABLE OF CONTENTS

LIST OF ABBREVIATIONS vii

LIST OF FIGURES viii

LIST OF TABLES x

ABSTRACT xi

ÖZET xii

1 INTRODUCTION 1

1.1 Literature Review 1

1.2 Objective of the Thesis 3

1.3 Hypothesis 4

2 FUNDAMENTAL CONCEPTS 5

2.1 Convolutional Neural Networks 5

2.1.1 Basic Components of CNN 7

2.2 Recurrent Neural Networks 12

2.2.1 Long-Short Term Memory (LSTM) Networks 13

2.3 The Yolov5 Network 14

2.4 Nvidia Jetson Nano as an Embedded System 15

2.5 Embedded System Peripherals 16

2.5.1 Power Supply 16

2.5.2 Local Storage Device 17

2.5.3 Camera 18

2.5.4 USB Audio Card 18

2.5.5 HDMI Capture Card 19

2.5.6 ZTE 3G Internet Modem 19

3 APPLIED METHODOLOGY AND USED MATERIALS 21

3.1 Dataset Creation 21

3.1.1 Collection of Images 21

3.1.2 Labeling of the Images 23

3.2 Proposed Framework 27

4 EXPERIMENTAL RESULTS **33**

4.1 Experiments of Object Detector (YOLOv5) 33

4.2 Experiments of Driver Gaze Classifier 38

4.3 Experiments of LSTM network 42

4.4 Real-time Experiments 43

5 RESULTS AND DISCUSSION **50**

REFERENCES **51**

A ETHICS COMMITTEE APPROVAL **55**

PUBLICATIONS FROM THE THESIS **58**

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CONV	Convolution Layer
DAS	Data Acquisition System
FC	Fully Connected Layer
LSTM	Long-Short Term Memory
NHTSA	National Highway Traffic Safety Administration
NIR	Near Infra-Red
ReLu	Rectified Linear Unit
RNN	Recurrent Neural Network
SDM	Supervised Descend Method
SHRP2	The Second Strategic Highway Research Program
YOLO	You Only Look Once

LIST OF FIGURES

Figure 2.1 The structure of Le-Net-5 [23] 5
Figure 2.2 The structure of AlexNet [30] 6
Figure 2.3 The structure of VGGNet [33] 6
Figure 2.4 The structure of ResNet (152 Layer) [35] 7
Figure 2.5 A 2-D convolution example without kernel flipping. 9
Figure 2.6 An illustration of a pooling operation: (a) is the pooling input; (b) is the output of a max-pooling operation; and (c) is the output of an average pooling operation. 10
Figure 2.7 Convolution operation with the stride value set to 2 and applied to zero padding. 11
Figure 2.8 A simple RNN cell's structure with sequential connections [46]. . . 12
Figure 2.9 Representation of an LSTM cell's structure [46]. 13
Figure 2.10 The structure of the YOLOv5 network. 15
Figure 2.11 The provided Nvidia Jetson Nano 4GB single board computer with accesories for installing to a car. 17
Figure 2.12 The schematic of the DC-DC buck converter circuit. 17
Figure 2.13 PNY 128GB SD card was used for the embedded system's local storage. 18
Figure 2.14 A view of Raspberry Pi Camera v2.1 18
Figure 2.15 USB external sound card used for audible alerts. 19
Figure 2.16 A view of the HDMI capture card used in system testing. 20
Figure 2.17 A view of the ZTE MF65M 3G modem that is utilized for the system's internet access and send/receive SMS messages. 20
Figure 3.1 A view of collected images from participants 22
Figure 3.2 The scheme of the data collecting 24
Figure 3.3 The scheme of the preprocessing and data augmentation 26
Figure 3.4 Driver Gaze Classifier data preparation for training. 26
Figure 3.5 The concept of proposed framework. 29
Figure 3.6 The web-based remote monitoring dashboard. 30
Figure 4.1 Experiment#1, the precision-recall curves of all classes.(The larger the area under the graph, the better the results.) 33

Figure 4.2 Experiment#2, the precision-recall curves of reduced classes (Focused on driver, the larger the area under the graph, the better the results). 34
Figure 4.3 The confusion matrix of the Experiment#2 in YOLOv5 (High values on the diagonal indicate high classification success). 35
Figure 4.4 Experiment#3, the precision-recall curves of all classes.(The larger the area under the graph, the better the results.) 36
Figure 4.5 Experiment#4, the precision-recall curves of reduced classes (Focused on driver, the larger the area under the graph, the better the results). 36
Figure 4.6 The confusion matrix of the Experiment#4 in YOLOv5 (High values on the diagonal indicate high classification success). 37
Figure 4.7 The detection examples of YOLOv5 Object Detector. 39
Figure 4.8 The train performance for 50 Epochs of Driver Gaze Classifier. . . . 40
Figure 4.9 The confusion matrix of Resnet50 network for Driver Gaze Classifier. 40
Figure 4.10 Prediction of driver gaze classification after detection of driver face. The estimation of the direction the driver is facing is automatically added to the image by the software. 41
Figure 4.11 RMSE loss graph of the results of the LSTM network trained for 100 epochs 43
Figure 4.12 An image from the first real-time test. 44
Figure 4.13 An image from the first real-time test (Phone Call: True). 45
Figure 4.14 An image from the first real-time test (Texting: True). 45
Figure 4.15 An image from the first real-time test (Smoking: True). 46
Figure 4.16 Docking to the windshield the system for the second real-time test. 46
Figure 4.17 An image from the second real-time test. 47
Figure 4.18 An image from the second real-time test. 47
Figure 4.19 An image from the second real-time test. 48
Figure 4.20 An image from the second real-time test. 48

LIST OF TABLES

Table 2.1 Technical specifications of Nvidia Jetson Nano 4GB 16
Table 2.2 Technical specifications of Local Storage Device 18
Table 2.3 Technical specifications of CSI Raspberry Pi Camera v2.1 19
Table 2.4 Technical specifications of ZTE MF65M 3G modem 20
Table 3.1 Summerize of collected data from participants. P represents "Participants", and G represents "Gender" 23
Table 3.2 The distribution of labeled Object Detector data 25
Table 4.1 Training results of ResNet networks trained for Driver Gaze Classifier implementation. 40
Table 4.2 Training results of ResNet50 3-fold cross validation for Driver Gaze Classifier. 41
Table 4.3 Default center coordinate values of the objects that are not detected in scene. 42

ABSTRACT

Embedded System Design for Drive Security

Kubilay ATAŞ

Department of Electronics and Communication Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Revna ACAR VURAL

The effect of fatalities and injuries on families and society encourages authorities to study the causes of the increase in traffic accidents. It is essential to underline that the majority of motor vehicle collisions are caused by driver distraction. It is a well-known fact that cigarette smoking and cell phone use contribute to driver distraction. Yolov5, one of the object detection algorithms, is used in the study to detect mobile phone use and smoking in a vehicle's interior. After detecting the predefined classes, the center coordinates of these components are supplied to an LSTM network to detect smoking and mobile phone use together temporal information. On the other hand, a classifier that is focused on the driver's face is offered information to the LSTM network by determining whether or not the driver is currently looking at the road. The training data is created by manually labeling the images collected via a mobile phone placed on the windshield. With this data, the artificial neural network-based systems used in the proposed method are trained and the results are analyzed. As a result, the system that performs well is realized in the physical world via an embedded GPU-supported single-board computer.

Keywords: Driver distraction, smoking detection, cellphone usage detection, Yolov5

YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ÖZET

Sürüş Güvenliği İçin Gömülü Sistem Tasarımı

Kubilay ATAŞ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Revna ACAR VURAL

Ölüm ve yaralanmaların aileler ve toplum üzerindeki etkisi, yetkilileri trafik kazalarındaki artışın nedenlerini araştırmaya teşvik etmektedir. Trafik kazalarının çoğunluğunun, sürücünün dikkat dağınıklığından kaynaklandığının altını çizmek gerekir. Sigara ve cep telefonu kullanımının, sürücünün dikkatinin dağılmasına neden olduğu bilinen bir gerçektir. Çalışmada, nesne tespit algoritmalarından biri olan Yolov5, bir aracın içinde cep telefonu kullanımını ve sigara içilmesini tespit etmek için kullanılmıştır. Önceden tanımlanmış sınıflar tespit edildikten sonra, bu bileşenlerin merkez koordinatları, sigara ve cep telefonu kullanımını tespit etmek için bir LSTM ağına verilir. Öte yandan, sürücünün yüzüne odaklanan bir sınıflandırıcı, sürücünün şu anda yola bakıp bakmadığını belirleyerek LSTM ağına bilgi sunmaktadır. Eğitim verileri, ön cama yerleştirilen bir cep telefonu aracılığıyla toplanan görüntülerin manuel olarak etiketlenmesiyle oluşturulmuştur. Bu veriler ile önerilen yöntemde kullanılan yapay sinir ağı tabanlı sistemler eğitilmiş ve sonuçlar analiz edilmiştir. Sonuç olarak, iyi performans gösteren sistem, fiziksel dünyada GPU destekli bir tek kartlı bilgisayar aracılığıyla gerçekleştirilmiştir.

Anahtar Kelimeler: Sürücü dikkat dağınıklığı, sigara kullanımı tespiti, cep telefonu tespiti, Yolov5

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

1
INTRODUCTION

Each year, millions of people are killed or injured in automobile accidents [1]. The effects of these fatalities and injuries on families and society are unpredictable and significant. Tragically, these losses have irreparable effects on people's lives. In addition to the emotional hurts created by these losses, families and society often suffer financial consequences.

Authorities are increasingly concerned about the danger posed by distractions in driving. The majority of causal factors of accidents due to driver attention attributes the fault to the increased use of mobile cellphones by drivers. The term "distraction" refers to the fact that the driver is engaged in some other activity or is momentarily concentrating on something, someone, or something that is not directly related to the primary duty of driving [2]. According to [3], in 2020, 8% of crashes that resulted in fatalities, 14% of accidents that resulted in injuries, and 13% of all police-reported crashes are caused by distractions.

Furthermore, smoking might have the same effect as using a cell phone. Driving while smoking is considered to be one of the most distracting activities. It can be included among the smoking-related causes of driver distraction, such activities as looking for a cigarette and lighter before smoking, lighting a cigarette, throwing ash into the ashtray, and extinguishing a cigarette. Developing a solution to prevent these distracting events has been the motivating factor behind this work.

1.1 Literature Review

Driving a vehicle is a quite challenging activity. For many years, road safety experts and other partners have identified driver distraction as a major threat to road safety, as it considerably increases the accident risk. 25 percent of police-reported crashes involve some sort of driving inattention, such as the driver being distracted, fatigued, or otherwise lost in thought [4, 5], according to NHTSA statistics [6]. This problem has

been carefully researched for years by mobility and transport specialists. Numerous researchers have come up with definitions for the term distraction. For example, Lee et al. [7] described driver distraction as "the diversion of attention away from safe driving activities to a competitive activity." Distraction happens when a driver "is delayed in the recognition of information needed to successfully complete the driving task due to any occurrence, activity, material, or person inside or outside the vehicle that forces or provokes the driver's diverting attention away from the driving process." [4].

Despite the notion that distraction is a driver's cognitive state, studies distinguish between several distraction detection methodologies, including visual, manual, and cognitive distraction [8–10]. However, activities may also consist of a combination of two or even three sorts of distractions, i.e., holding a cellphone for reading (manual and visual distraction), gazing the cellphone for texting, and contemplating a text message to send (manual, visual, and cognitive distraction). In the light of these concepts, researchers have carried out various studies that include detecting the distraction of drivers and giving warnings or taking action. In [11], three primary ways to such detection are discussed: observation of driver cognition, tracking of driver maneuvering and lane-keeping behavior, and detection of the driver's participation in a secondary job itself. Numerous research focuses on the detection and modeling of fatigue and/or stress as significant causes of inattention [12, 13]. Nevertheless, as demonstrated in [14], visual distraction also impairs driving performance. In the thesis, it is focused on the use of mobile phones and smoking cigarettes, which include manual and visual distraction types, which are among the measurement methods of driver distraction. In addition, making a cell phone call while driving is one of the situations that causes the cognitive capacity to be separated from the driving activity.

Recent research has focused on the broader topic of driver mobile phone usage detection in the outside-vehicle environment, as opposed to the inside-vehicle environment. The authors of [15] utilized data obtained by a highway traffic imaging system to detect driver cell phone use. The cameras were mounted in an elevated location with their lenses facing the roadway. They searched for faces on the right part of the windscreen after detecting the windshields of the vehicles. The photos collected in this manner were utilized to train the classification algorithm. Even though several of the photographs contained only half of a face, the final results were satisfactory. The authors of [16] collected data using a system called DAS that was attached to the vehicle's windshield. Using the SDM, they identified the places where mobile phones are expected to be located and cropped these sections. The algorithm extracts facial landmarks from photos and trims the left and right ear areas. They built feature vectors from gathered crops; they used these vectors to train a classifier to assess whether or not a cell phone is present. Consequently, the strategy with the highest

accuracy score runs at a maximum of 7.5 FPS with 93 percent accuracy.

In addition to the detection of cell phones [17], Le et al. also employed another method, which including tracking the driver's hands, to decide whether or not the driver's hands were on the directing wheel. They used videos and photos of a steering wheel, faces, and hands that were released in SHRP2 [18] and VIVA competitions [19]. In the research using the proposed MSFRCNN network [17], the driving wheel and hands are detected separately in distinct branches. Then, the number of hands in the area where the steering wheel is recognized is determined using a threshold where the joint region is larger than 5 percent of the hand area. Concerning cell phone usage recognition, they utilized the same technique, but due to the occlusion problem, they chose to focus on the identification of patterns with faces and hands during the cell phone conversation rather than the cell phone devices. With their suggested MS-FRCNN network, they have been successful in detecting mobile phone use with an accuracy of 94.2 percent at a frame rate of 0.06 FPS.

Concerning smoking detection, Artan et al. [20] collected data from an NIR surveillance camera mounted on an elevated structure and facing towards traffic. Using this information, they initially identified the windshield region of the automobiles. Then, a pre-trained single-shot detector is employed to detect the driver on the right side of the identified windshield region. Due to heat of the burning cigarette, the NIR image displays hot spots. Once the driver's location has been determined, the system finds for a white blob that indicates cigarettes in flames. Due to the glare influence of other things in the driver's region, such as glass, jewelry, and watches, the accuracy performance of the suggested method is superior to its sensitivity performance.

1.2 Objective of the Thesis

The goal of this study is to detect the driver's use of a cellphone and/or a cigarette while driving by analyzing photos taken with the help of an in-vehicle camera. By installing a closed-circuit monitoring system, it is hoped that the driver becomes aware of this behavior and tends to be more cautious as a result of receiving an auditory warning when mobile phone use and/or smoking, the most significant factors that cause driver distraction, are detected. In addition, the created data set will be used to construct the infrastructure for detecting smoking among passengers and drivers, among other future studies that can be proposed. The project also wants to set up the infrastructure for a commercial product that can provide proof of driver-caused traffic accidents to car rental companies, companies that have used car rental services, and

businesses that want to keep track of how their fleet of vehicles is being used.

1.3 Hypothesis

Modern artificial intelligence networks can now tell when a driver is using a cell phone or smoking, both of which can be distracting, just by processing images from a camera. For this, it is sufficient to prepare well-constructed training data, an object detection algorithm whose performance has been proven in the literature, and a framework in which an LSTM network that can process temporal information will work in harmony. The designed framework can be run successfully on a single-board computer with GPU support. This newly created system can be installed in the vehicle and run effectively in real-time.

2
FUNDAMENTAL CONCEPTS

2.1 Convolutional Neural Networks

CNNs are a subset of deep learning widely used for the visual perception of computers. It is inspired by the natural sight mechanism of biological creatures. Hubel & Wiesel found in 1968 that cells in the visual cortex of spider monkeys are in charge of detecting light in receptive fields [21]. Fukushima [22], influenced by the work of Hubel & Wiesel's, proposed the neocognitron in 1982, which could be taken into account as the precursor of CNN. In 1998, LeCun et al. announced Le-Net-5 [23] that able to classify hand-written characters establishing the modern framework of CNN, the structure of which can be seen in Figure 2.1. However, this and its similar networks [24] were not able to perform effectively on more difficult issues such as large-scale image and video categorization, due to requiring huge training data and lack of computational capacity at the time. The MNIST data [25]

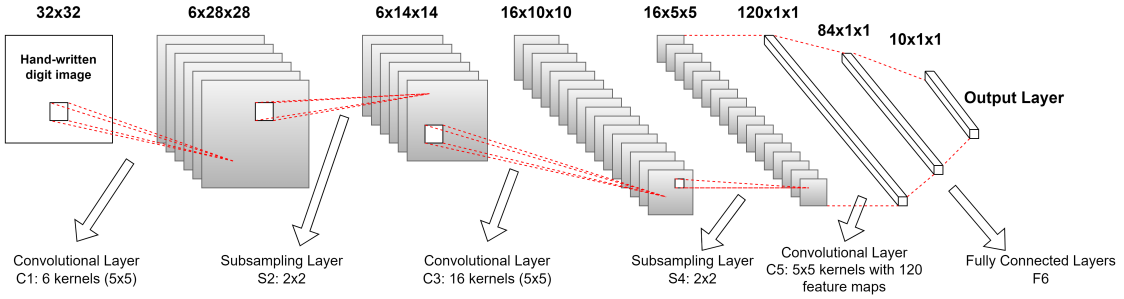


Figure 2.1 The structure of Le-Net-5 [23]

including hand-written characters published by LeCun has been used to evaluate several methods to improve classification accuracy [26–28]. With the development of technology, computers have gained parallel processing capabilities and AI has become more attractive to researchers, thus CNNs with deeper structures and can be able to more accurate classifications have been developed. The Imagenet data set, released in 2009, provided a new environment for testing and comparing CNN algorithms, meeting the data requirements for large-scale image classification [29]. Within the scope of large-scale image categorization, Krizhevsky et al. proposed a classic CNN

architecture and showed significant improvements over previous methods on the image classification task. As illustrated in Figure 2.2, the overall architecture of their method, AlexNet [30], is similar to LeNet-5 but has a more complex structure. In pursuit of this, the development of CNN algorithms accelerated with the release of the first image recognition competition in 2015 [31]. Many approaches have been investigated to improve AlexNet's performance with its success. ZFNet [32], VGGNet [33], GoogleNet [34], and ResNet [35] are four representative efforts among them. A common trend in the evolution of architectures is that networks become deeper. Figures 2.1, 2.2, 2.3, and 2.4 depict the architectural structures of LeNet, AlexNet, VGGNet, and ResNet, respectively. With 152 layers, ResNet, which won the 2015 ILSVRC competition, is deeper than VGGNet. With 19 layers, VGGNet is also more complex than AlexNet. AlexNet, on the other hand, has 12 layers and is therefore more complex than LeNet, which has only 8 layers. With more nonlinearity and depth, the network can better approximate the target function and improve feature representations. However, it increases the network's complexity, making optimization more difficult and overfitting more likely. Researchers have come up with a variety of approaches to dealing with these issues along the road [36–39].

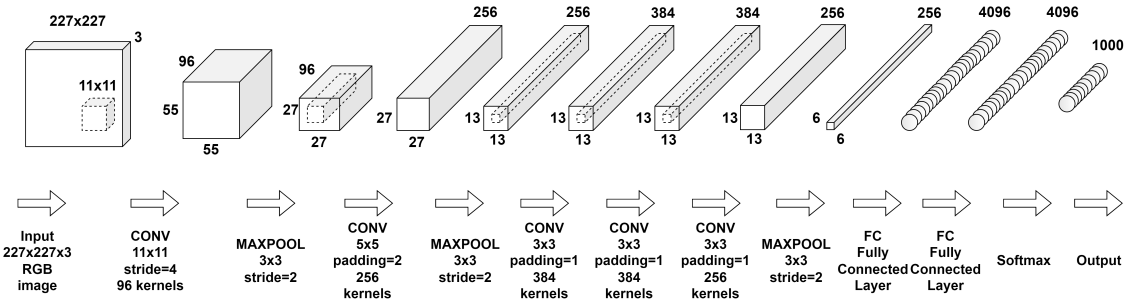


Figure 2.2 The structure of AlexNet [30]

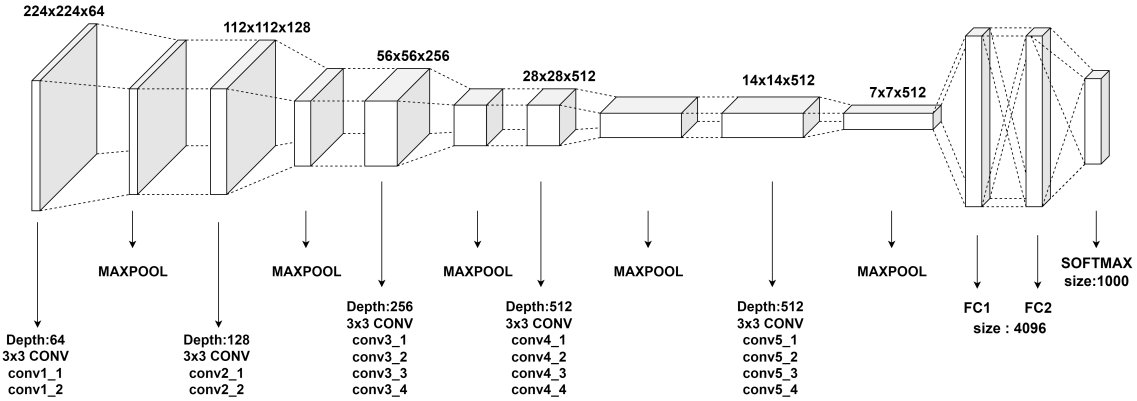


Figure 2.3 The structure of VGGNet [33]

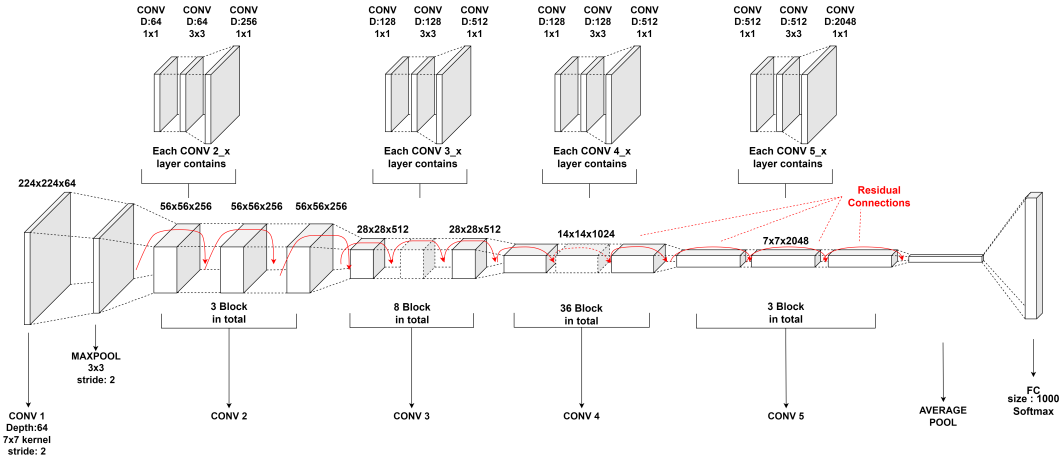


Figure 2.4 The structure of ResNet (152 Layer) [35]

2.1.1 Basic Components of CNN

There are various CNN architectural variants described in the literature. However, their fundamental components are identical. Using the well-known LeNet-5 as an example, it has three types of layers: convolutional layer, pooling layer, and fully-connected layer. The convolutional layer seeks to learn representations of input features. As depicted in Figure 2.1, the convolution layer consists of multiple convolution kernels that are used to generate various feature maps. Specifically, each neuron in a feature map is connected to an area of adjacent neurons in the layer beneath it. In the previous layer, this neighborhood is referred to as the neuron's receptive field. The new feature map can be made by first convolving the input with a learned kernel and then applying an element-wise nonlinear activation function to the outputs of the convolving. It should be known that a kernel is shared across all spatial locations of input in order to produce each feature map. Using many distinct kernels, the entire feature maps are generated.

2.1.1.1 Convolutional Layer

Convolution is an operator on two functions of a genuine parameter in its most basic form. Assume we have a single-valued function $x(t)$ and a parameter t that specifies the time on which this function is dependent. We might think that taking the average of many measurements would reduce the noise in an $x(t)$ signal with noisy information. Since more recent measures are more informative, we'll want to use a weighted average that gives recent values greater weight. A weighting function $w(a)$, where a represents the antiquity of a measurement, would be used to do this. When such a weighted average operation is performed for each t, a new function $s(t)$ that estimates the smoothed values of $x(t)$ is obtained. Convolution is the name for this operation, which is given in equation (2.1). As shown in equation (2.2), the

convolution operation is usually symbolized by an asterisk.

$$s(t) = \int x(a)w(t-a)da \tag{2.1}$$

$$s(t) = (x * w)(t) \tag{2.2}$$

In the jargon of convolutional networks, the primary argument to the convolution is commonly referred to as the input (x) and the secondary argument as the kernel (w). Also, the output is referred to as the "feature map" most of the time. Typically, while working with data on a computer, time is divided into short segments, and measurements provide data at regular intervals. Thus, the time index t can be a vector containing only integer values, and x can be a vector holding measurements with integer values. In this instance, equation (2.1), which is depicted by equation (2.2), must be modified so that it resembles equation (2.3).

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.3}$$

In computer vision applications of machine learning, the input is typically a two or more dimensional array of data (image), and the kernel is typically a two or more dimensional array of parameters that are adapted by the learning model. If a two-dimensional image I is used as input, it is likely that a two-dimensional array will also be required for the kernel K. Equation (2.4) is the result of modifying equation (2.3) to work with a two-dimensional input and a two-dimensional kernel. Moreover, because convolution is commutative, it can be expressed as equation (2.5) and is typically simpler to implement in computer software. Negative expressions in the term K mean that the kernel will be applied by flipping it. Because most kernels are symmetrical, most software libraries use a process called "cross-correlation" on the kernel and image, as shown in equation (2.6), without flipping it. See Figure 2.5 for an illustration of convolution on a two-dimensional input image with a non-flipping kernel.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.4}$$

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{2.5}$$

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{2.6}$$

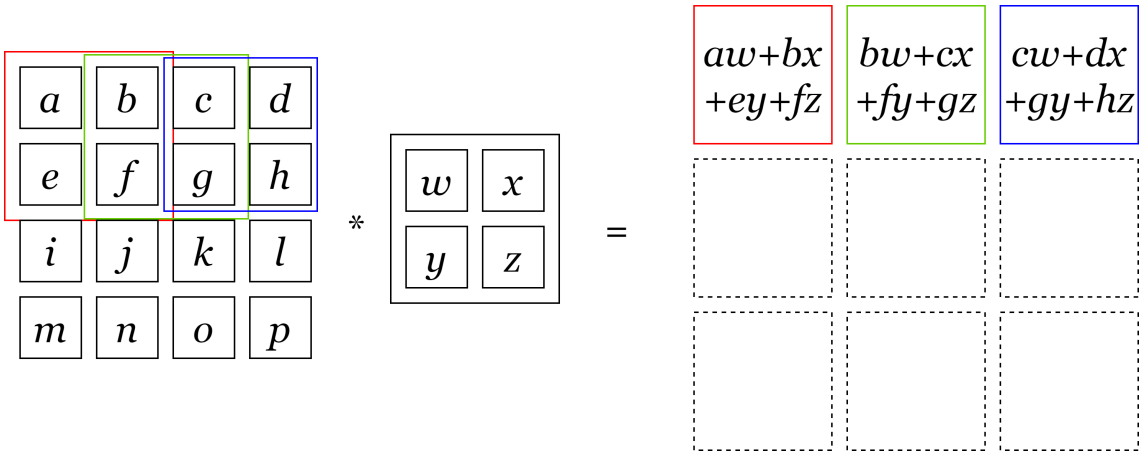


Figure 2.5 A 2-D convolution example without kernel flipping.

2.1.1.2 Pooling Layer

A classic convolutional network layer comprises three stages. In the initial stage, the layer executes many parallel convolutions to generate a set of linear activations. Each linear activation is passed via a nonlinear activation function, such as the ReLu activation function, in the second stage. Occasionally, this stage is referred to as the detector stage. In the third stage, the output of the layer is changed using a pooling function.

A pooling function replaces the net's output at a particular location with a summary statistic of the outputs in the vicinity. The max-pooling technique [40], for instance, returns the largest output within a rectangle area. Other common pooling functions include the mean of a rectangular neighborhood, the L^2 norm of a rectangular neighborhood, and a distance-based weighted average. In all of these cases, pooling helps make the representation almost unaffected by small changes to the input. Invariance to translation denotes that if the input is translated by a slight amount, the most of the pooled outputs preserve their previous meaning. Figure 2 illustrates, in a general sense, how the pooling process can be adapted to either the output of a convolution layer or a picture. The mixed pooling method is yet another type of pooling method. This method combines the maximum pooling method with the average pooling method [41]. Equation (2.7) can be used to express the function that is performed by mixed pooling. Where λ is a value chosen at random, which can either be 0 or 1, indicating whether the maximum pooling method or the average pooling method will be utilized. The output of the pooling operator at point (i, j) in the kth feature map is denoted by the letter $y_{i,j,k}$. Also, α represents the value of the feature at the coordinates (m, n) in the pooling region R_{ij} of the kth feature map. During the phase of the process known as forward propagation, the value λ is stored so that it

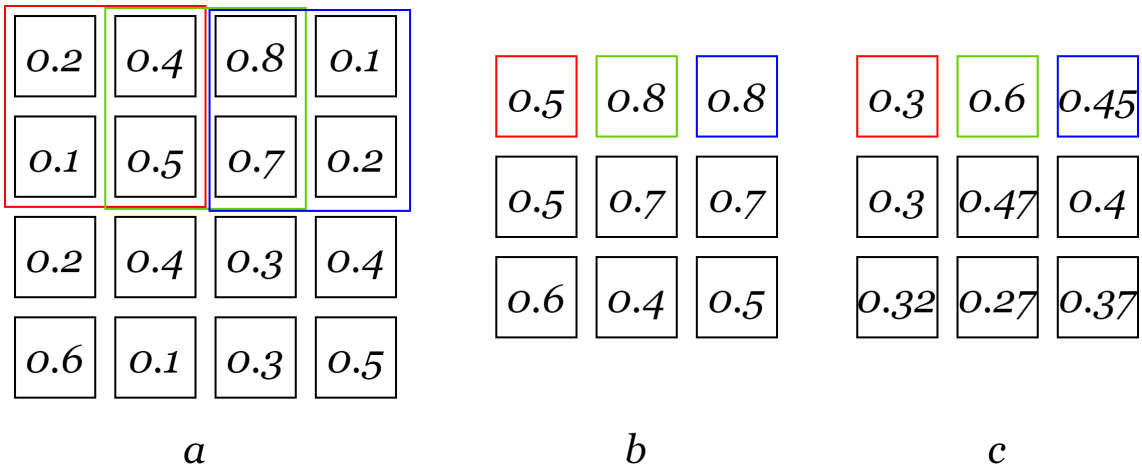


Figure 2.6 An illustration of a pooling operation: (a) is the pooling input; (b) is the output of a max-pooling operation; and (c) is the output of an average pooling operation.

can be utilized during the phase known as backpropagation.

$$y_{i,j,k} = \lambda \max_{(m,n)\in R_{i,j}} \alpha_{m,n,k} + (1-\lambda)\frac{1}{|R_{ij}|}\sum_{(m,n)\in R_{ij}} \alpha_{m,n,k} \tag{2.7}$$

2.1.1.3 Stride and Padding

In some instances, CNN layers may require input and output sizes to be identical. However, due to the nature of the pooling process, the output size is lowered throughout its application. Depending on the size of the kernel, it may also be necessary to resize the input image to produce the desired output size. However, resizing the input may cause hosted features to be lost or enlarged. In this instance, the padding process is applied to the input image. Padding is the process of adding extra pixel frames to an image in order to enlarge it. Similarly, it may be essential to reduce the size of some CNN layers. Convolution (or cross correlation if an unflipped-kernel is applied) can be utilized efficiently with modest adjustments to minimize the image's size while maintaining its features. The stride value can be increased in this instance. The original convolution procedure can be assumed to be performed with a stride value of 1. Figure 2.7 shows how a convolution operation works when the stride value is set to 2 and zero padding is used.

2.1.1.4 Fully Connected Layer

Fully connected layer(s) is/are located in the network over a certain combination of convolutional and max pooling layers. Each neuron in this layer is, like a Multi Layer Perceptron [42], connected to the previous layer via its neurons, with each connection

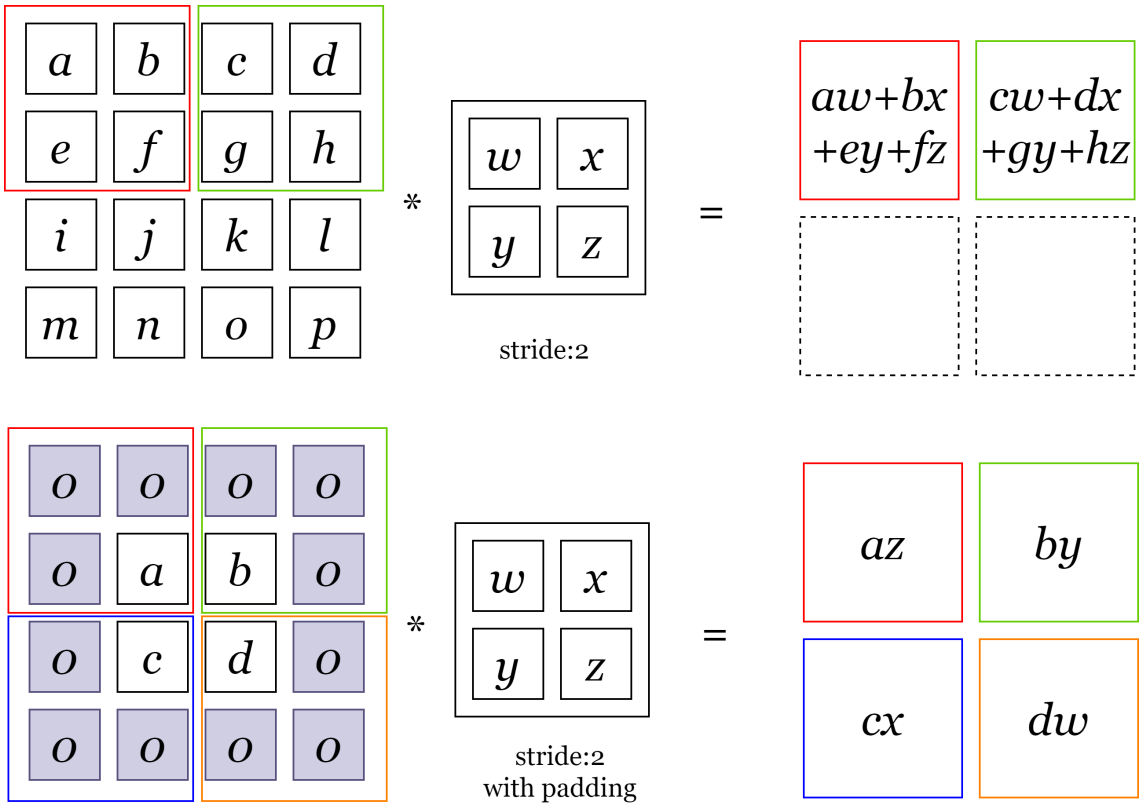


Figure 2.7 Convolution operation with the stride value set to 2 and applied to zero padding.

having its own weight.

2.1.1.5 Activation Function

In the concept of artificial neural networks, the output of a particular node is determined by the activation function of that node given the input or inputs that came from the receptive area. Accordingly, the information from the receptive area is collected and utilized as the activation function's input parameter. Thus, whether the artificial neuron is active or not is decided. Equation (2.8) is a representation of the mechanism that determines the output of a neuron. Here, W stands for the weights, X for the inputs, and b for the bias value of the neurons.

$$Y = ActivationFunction(\sum(W * X) + b) \tag{2.8}$$

CNNs also make use of activation functions in subsequent levels of their networks. The Rectified Linear Unit (ReLU)[43] activation function is a well-known non-saturated activation function used in CNNs. Because of its simple structure and low computational cost, it is the most commonly utilized activation function in CNNs. One of its drawbacks is the discontinuity at the zero point. Leaky ReLU has solved

this problem by using a minor leak function [44]. The activation functions of ReLU and Leaky ReLU are given in equations (2.9) and (2.10), respectively. In (2.10), λ is a predetermined parameter that is chosen from a range of $(0, 1)$.

$$Y = \max(\sum(W * X) + b, 0) \tag{2.9}$$

$$Y = \max(\sum(W * X) + b, 0) + \lambda \min(\sum(W * X) + b, 0) \tag{2.10}$$

2.2 Recurrent Neural Networks

RNNs are a modified form of feed-forward neural networks that employ sequential data [45]. At each time t, just the current input is accepted for classification in CNNs. No consideration is given to the outcomes of earlier or subsequent entries; just the input fed into the current network is assessed. In many real-world applications, however, both inputs and outputs are sequential. Videos, e.g., are sequences of images. To determine the nature of a movement or activity, one must examine the sequence in which the movements occur.

In contrast to CNNs, RNN does not simply consider the current input at each time t; rather, the current entry is created by combining the current entry with the hidden state resulting from the previous entry going through the network. The RNN architecture is utilized to tackle problems that feed-forward networks cannot. Figure 2.8 depicts the structure of a straightforward RNN. X_t denotes the input at time t, while h_t denotes the output at time t. The hidden state of the RNN cell that will be passed to the next cell is represented by y_t.

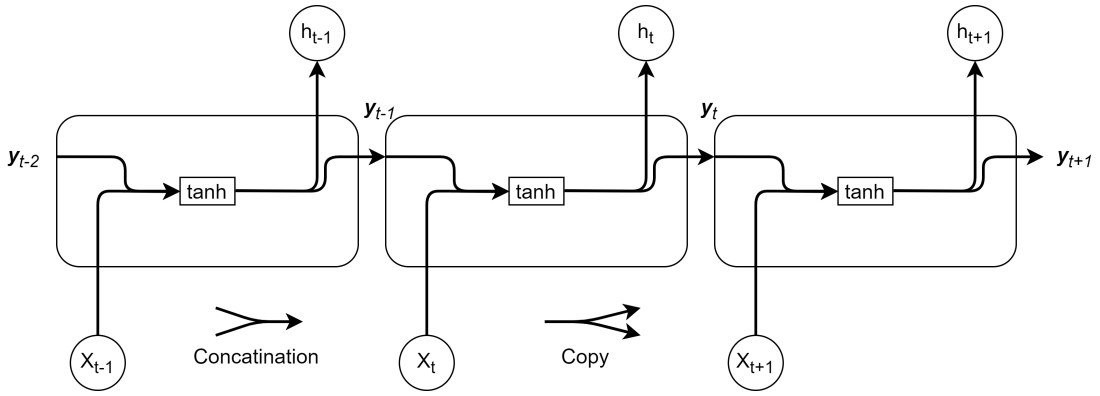


Figure 2.8 A simple RNN cell's structure with sequential connections [46].

2.2.1 Long-Short Term Memory (LSTM) Networks

RNNs have an issue with long-term dependence. In lengthy time series, the initial inputs of a pattern are forgotten as the pattern continues to its conclusion. In 1997, Hochreiter and Schmidhuber created the LSTM Network to combat long-term dependency [47]. Due of the several gates it contains, LSTM is superior to traditional RNN at capturing dependencies between inputs. The LSTM cell has three extra gates than a normal RNN: the input, the forget, and the output gate. In addition, it features a memory structure identical to that of RNN. Depending on the current state of the cell, the input gate regulates which data is entered from the new entry. Controlling the effect of the prior state on the current state, the forget gate determines which information should be forgotten. The output gate finally determines the next hidden state based on the present state of the cell. It can be seen that the LSTM cell structure from Figure 2.9.

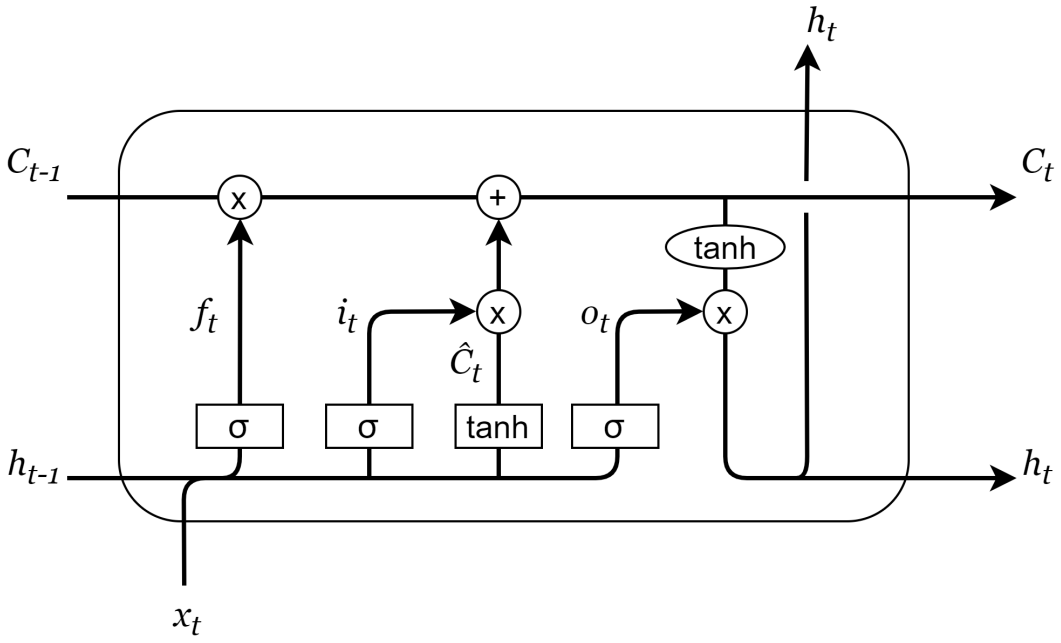


Figure 2.9 Representation of an LSTM cell's structure [46].

The initial stage of LSTM is to decide which information from the cell state will be discarded. The "forget gate layer," a sigmoid layer, makes this judgment. It analyzes the values in h_{t-1} and x_t, and then it returns a number from a range of 0, which indicates "entirely discard this," to 1, which indicates "totally keep this," for the cell state C_{t-1}. The operation of the "forget gate layer" is modeled in (2.11).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.11}$$

The following step is to determine what new data will be stored in the cell state.

This consists of two parts. Initially, a sigmoid layer known as the "input gate layer" determines which values will be updated. A $tanh$ layer then generates a vector of new candidate entries, $\hat{C}_t$, that might be added to the state. This step is represented by (2.12) and (2.13).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.12}$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.13}$$

In the subsequent stage, the previous cell state, denoted by C_{t-1}, is updated to the current cell state, denoted by C_t. By multiplying the previous state by f_t, the data that was decided to be forgotten before is erased. Then the result of the $i_t * \hat{C}_t$ operation is summed with the result of the forget gate layer. Consequently, the current state of the cell, C_t, is determined. These processes are represented by (2.14).

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{2.14}$$

The concluding step decides the output of the cell. This output is a filtered version of the current state of the cell. The sigmoid layer determines which components of the cell state will be output. In parallel, the cell state C_t is passed through the $tanh$ function, which forces the result between -1 and 1, and the sigmoid layer output is multiplied by this result. Thus, temporal information for subsequent cells is generated. The mathematical model of the final step is represented by (2.15) and (2.16).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.15}$$

$$h_t = o_t * \tanh(C_t) \tag{2.16}$$

2.3 The Yolov5 Network

YOLOv5's network structure is divided into three parts: backbone, neck, and output. The initial structure in the backbone section is Focus, which the 640x640x3 input tensors pass through. After the slicing process, the input tensor becomes a 320x320x12 feature map, and then the convolution operations with 32 kernels are applied [48]. After that, it's a 320x320x32 feature map. Conv2D + BatchNormalization + LeakyRELU are represented by the following module CBL, which is a simple convolution module.

The feature map is extracted using the BottleneckCSP module, which extracts a lot of information from photos. The BottleneckCSP structure outperforms other convolutional neural networks by decreasing gradient information in CNN optimization. This structure's intricacy takes up the entire network.

Four distinct models with varying complexities can be obtained by altering the width and depth of the modüle. The YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x models obtained in this technique were published in [48] as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The SPP module is used to increase the network's receptive field and obtain features of various scales.

A bottom-up feature pyramid structure based on FPN can be found in YOLOv5. While strong semantic features are imparted from top to bottom with this combined procedure, feature pyramid caries positional features are imparted from the bottom up. The network enhances its capacity to recognize targets with varying scales by mixing features from multiple feature layers. As a result, the network produces two outputs: categorization results and object coordinates. Figure 2.10 depicts the structure of the YOLOv5 network.

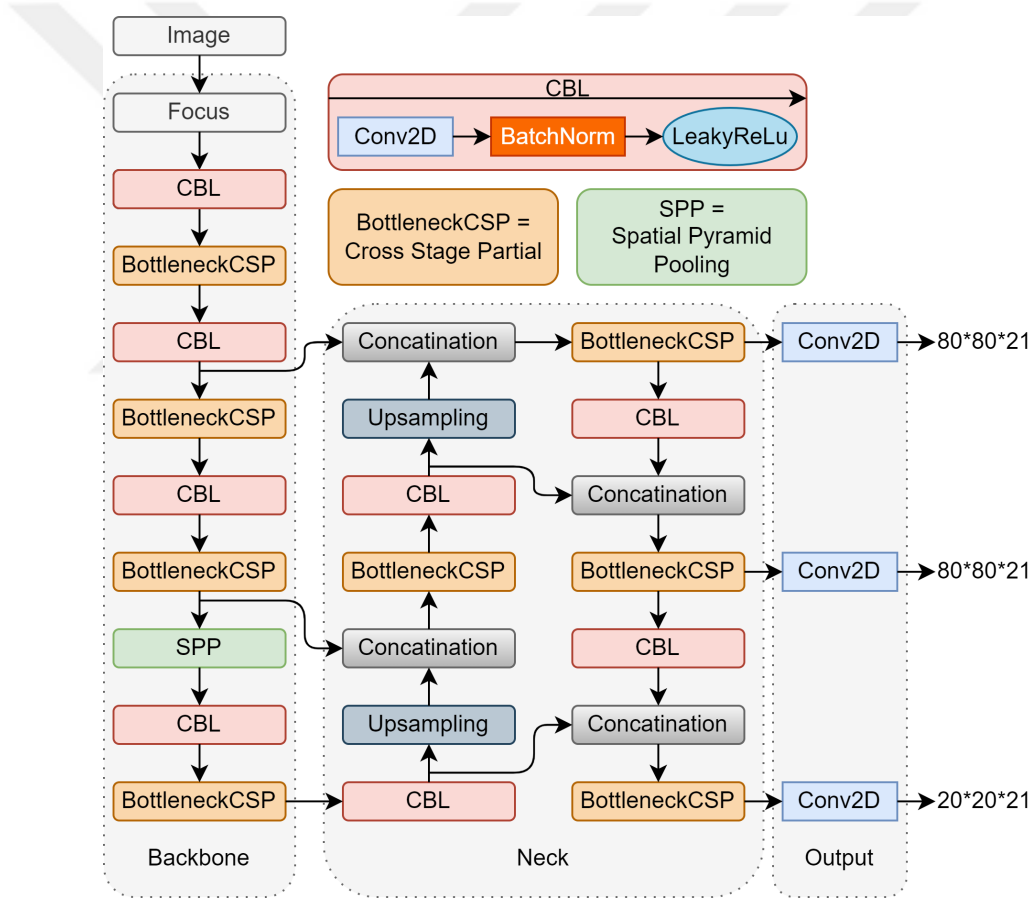


Figure 2.10 The structure of the YOLOv5 network.

2.4 Nvidia Jetson Nano as an Embedded System

An embedded system can be summarized as a computer system with limited features designed to perform a specific task, providing control with a microprocessor or microcontroller, and communicating with other computers with peripherals.

Embedded systems are often preferred to perform a specific task in image processing and deep learning studies. Especially if image processing is going to be done, single-board computers that offer GPU support come to the fore. Within the scope of this thesis, it was deemed appropriate to use the Jetson Nano 4GB model belonging to Nvidia Company. This single-board computer, which has 128 Cuda Cores, can be easily used in image processing and computer vision projects thanks to its 4 GB of RAM memory. The computer specifications are given in Table 2.1 in detail.

Table 2.1 Technical specifications of Nvidia Jetson Nano 4GB

TECHNICAL SPECIFICATIONS	
GPU	NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores
CPU	Quad-core ARM Cortex-A57 MPCore processor
Memory	4 GB 64-bit LPDDR4, 1600MHz 25.6 GB/s
Storage	16 GB eMMC 5.1
Video Encode	250MP/sec 1x 4K @ 30 (HEVC) 2x 1080p @ 60 (HEVC) 4x 1080p @ 30 (HEVC) 4x 720p @ 60 (HEVC) 9x 720p @ 30 (HEVC)
Video Decode	500MP/sec 1x 4K @ 60 (HEVC) 2x 4K @ 30 (HEVC) 4x 1080p @ 60 (HEVC) 8x 1080p @ 30 (HEVC) 9x 720p @ 60 (HEVC)
Camera	12 lanes (3x4 or 4x2) MIPI CSI-2 D-PHY 1.1 (1.5 Gb/s per pair)
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI 2.0 and eDP 1.4

Nvidia Jetson Nano 4GB was used for the installation and operation of the system designed within the scope of the thesis. In order for the computer to be easily mounted, a suitable box was printed from a 3D printer, and a separate adapter and DC-DC converter device was provided to operate it both in the vehicle and on mains voltage. Figure 2.11 shows the device provided.

2.5 Embedded System Peripherals

2.5.1 Power Supply

Since Nvidia Jetson Nano (4GB) needs at least $5V$ and $4A$ of DC power, it is important to limit the energy received from the $12V$ or $24V$ power outlet jack, which can vary according to the vehicle's electrical system, to $5V$. For this reason, the XL4015 $5A$

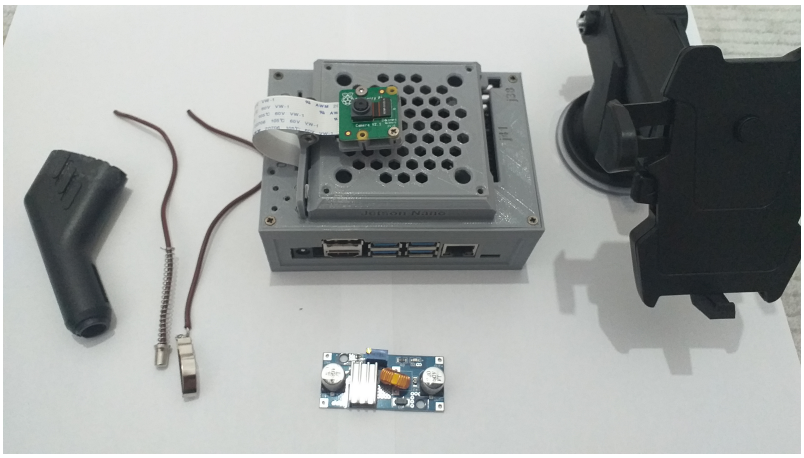


Figure 2.11 The provided Nvidia Jetson Nano 4GB single board computer with accesories for installing to a car.

adjustable DC-DC buck converter circuit produced by Acelex is used. This power supply has been tested under 12V and 24V DC voltages, and its output is adjusted to the 5V required by Jetson Nano via the potentiometer on it. Since most vehicle power outlet connectors on the market are designed for low power consumption, one of them has been modified to provide input power to the DC-DC converter. A 75mm^2 copper cable is used to increase the current carrying capacity. Figure 2.12 shows the schematic of the DC-DC buck converter circuit employed.

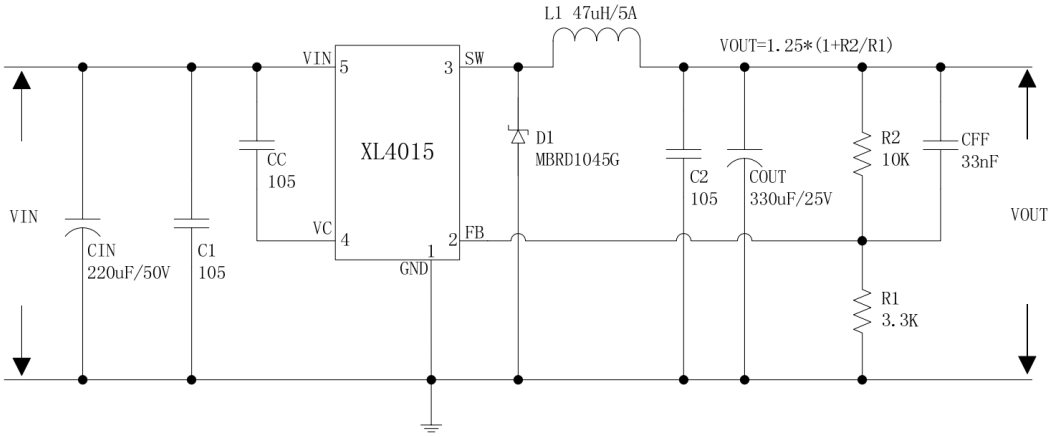


Figure 2.12 The schematic of the DC-DC buck converter circuit.

2.5.2 Local Storage Device

Data read and write speeds affect the operating speed of the embedded system during operation. As a result, using a high-capacity storage unit with fast read and write speeds will ensure that the system runs smoothly. High capacity is required for system files and images to be recorded during system operation. In light of this, a class 10 SD card from PNY with 128 GB of storage and 100 MB/s read-write speed was preferred.

Table 2.2 lists the specifications of the SD card used for data storage, and Figure 2.13 depicts a view of the SD card.

Table 2.2 Technical specifications of Local Storage Device

TECHNICAL SPECIFICATIONS	
Capasity	128 GB
Reading Speed	100 MB/s
Video Speed	C10, UHS-I, U1
Format Factor	microSDXC



Figure 2.13 PNY 128GB SD card was used for the embedded system's local storage.

2.5.3 Camera

The Raspberry Pi Camera v2.1 was chosen due to its CSI interface and accessibility on the market. Raspberry Pi Camera v2.1 is designed as a Raspberry Pi onboard accessory. It features an 8-megapixel Sony IMX219 image sensor and a unique fixed-focus lens. It has applications in CCTV security cameras, motion detection, and time-lapse photography. It can capture images at 3280 x 2464 pixels and films at 1080p30, 720p60, and 640x480p90. Figure 2.14 depicts a view of the camera, and Table 2.3 lists the camera's features.

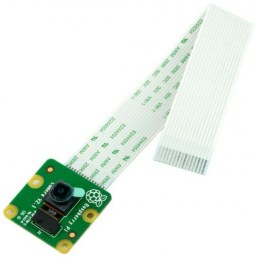


Figure 2.14 A view of Raspberry Pi Camera v2.1

2.5.4 USB Audio Card

Since the Jetson Nano lacks an internal speaker and sound equipment, an easily accessible USB sound card is employed to provide audio alerts. Figure 2.15 displays an image of this device.

Table 2.3 Technical specifications of CSI Raspberry Pi Camera v2.1

TECHNICAL SPECIFICATIONS	
Image Sensor	Sony IMX219PQ CMOS (fixed focus)
Resolution	8-megapixel
Image Resolution	3280 x 2464
Maximum Image Transfer Rate	1080p: 30fps 720p: 60fps
Hardware Connection	15-pin ribbon cable 15-pin MIPI Camera Serial Interface (CSI-2)
Image Control Functions	Automatic exposure control Auto white balance Automatic band filter Automatic 50/60 Hz brightness detection Automatic black level calibration
Lens Size	1/4”
Dimensions	23.86 x 25 x 9mm
Weight	3g



Figure 2.15 USB external sound card used for audible alerts.

2.5.5 HDMI Capture Card

During the development of the embedded system, an HDMI capture card was used to satisfy the requirement for a screen display. Thus, the image captured from the Jetson Nano HDMI connector is sent to the screen of a laptop. A secondary advantage of this card is that it prevents single-board computer fatigue while capturing screen outputs during system test studies. During system tests, screen outputs were captured via this HDMI capture card and recorded on another computer. This saved the Jetson Nano from this effort. Figure 2.16 shows a view of the HDMI capture card used in system testing.

2.5.6 ZTE 3G Internet Modem

A ZTE MF65M type 3g modem was used to connect the system to the internet. A modem offers Internet access by connecting through sim card to cellular base stations. Through its web interface, it may send and receive SMS messages, examine connection status, etc. This device enabled the system to transmit and receive data via MQTT



Figure 2.16 A view of the HDMI capture card used in system testing.

services. Thus, it is possible for remote computers to monitor the situations recognized by the system. Additionally, SMS messages can be delivered by sending get and post request commands to the modem's web interface. In this regard, the structure of the device is open source. Figure 2.17 contains a picture of the device, and Table 2.4 details the various functions and capabilities of the device.



Figure 2.17 A view of the ZTE MF65M 3G modem that is utilized for the system's internet access and send/receive SMS messages.

Table 2.4 Technical specifications of ZTE MF65M 3G modem

TECHNICAL SPECIFICATIONS
HSPA+/HSUPA/HSDPA/UMTS 2100/1900/900 (850)MHz
EDGE/GPRS/GSM 850/900/1800/1900MHz
HSPA+ Speed: DL 21.6Mbps/UL 5.76Mbps
OLED Screen display, 3 LED light indicators
Wi-Fi 802.11b/g/n, 10 Wi-Fi connections
Messaging: SMS
Memory: MicroSD Slot, up to 32GB
Built-in antenna
Support AP/STA, Multi SSID, WPS, IPv6
Dimensions: 102mm × 50mm × 13.7mm
Battery 1500 mAh Li-ion battery. gives up to 100 hours standby
OS Support: Win8, Win7, Windows XP, Vista, Linux, Mac OS

3
APPLIED METHODOLOGY AND USED MATERIALS

There is no pre-created dataset in the literature for the problem of detecting distraction sources such as cigarette smoking and making phone calls by drivers that has been presented in the previous chapters. In this section, the steps of creating a dataset that can be used as training data for the proposed method to solve the problem, the constraints of creating this dataset, and finally the proposed method will be mentioned. Since the created dataset is directly related to the proposed method, it is sometimes necessary to go ahead and look at the proposed method while examining the dataset creation steps.

3.1 Dataset Creation

This section will be broken down into two subsections: collection of images and labeling of the images.

3.1.1 Collection of Images

The environment created for the collection of images consists of a parked vehicle completely independent of traffic. Images were collected by ensuring that a vehicle is parked at different points within the university campus[1] and that there are volunteers willing to participate in the study at these points. In line with the rules set by the ethics commitee[2], volunteers were asked if they would like to take part in the study. The participants were asked to perform predefined scenarios, and the images were recorded during this time. During the video recording, the participants were given instructions at random times and were asked to follow the instructions without giving a detailed explanation of the scenario in order to reflect their own behavioral characteristics in these scenarios. The guidelines do not give instructions on how to hold a mobile phone or how to hold a cigarette while smoking. Exhibiting natural

[1] Yıldız Technical University Davut Paşa Campus
[2] Yildiz Technical University Academic Ethics Committee

behaviors during the image collection phase is important for preventing overfitting and generalizing the network. Figure 3.1 shows a selection of the photographs taken when the subjects engaged in these behaviors. These directives are explained as follows:

- Assume that you are currently driving normally.
- You can sometimes check the side mirrors while driving.
- Suppose you receive a message on your phone right now and try to read the message while driving.
- Suppose you want to smoke right now and light a cigarette (not asked to light it for real, but to simulate this event).
- Pretend that your cell phone is ringing right now and answer the phone.
- You can change hands while using the phone or smoking. Please do these activities as you normally would.



Figure 3.1 A view of collected images from participants

While the majority of participants are young students, four are men in their middle years. At the same time, there are 4 female participants and 17 male participants. Since the images were captured during the winter months, the majority of the images were captured during partly cloudy conditions. However, there are also videos where the sun comes directly from the windshield and casts shadows on the driver. In Table 3.1 is a summary of information regarding collected videos from participants.

Table 3.1 Summerize of collected data from participants. P represents "Participants", and G represents "Gender"

P	G	Date	Time of Data (hh:mm)	Duration (mm:ss)	Weather Condition	Data Size (MB)
#1	M	24 Feb	11:05	05:32	Sunny, Partly cloudy	686
#2	M	24 Feb	11:12	03:05	Sunny, Partly cloudy	381
#3	F	24 Feb	11:19	04:11	Sunny, Partly cloudy	518
#4	F	24 Feb	11:56	02:16	Sunny, Partly cloudy	281
#5	M	24 Feb	12:12	03:04	Sunny, Partly cloudy	380
#6	F	24 Feb	12:15	01:15	Sunny, Partly cloudy	155
#7	M	24 Feb	14:30	02:21	Sunny, Partly cloudy	291
#8	M	24 Feb	14:38	02:44	Overcast	339
#9	M	24 Feb	15:03	02:32	Overcast	314
#10	M	24 Feb	15:13	02:34	Overcast	317
#11	M	24 Feb	15:42	03:59	Overcast	493
#12	M	25 Feb	12:15	01:26	Sunny	179
#13	M	25 Feb	12:19	01:56	Sunny	240
#14	M	25 Feb	13:41	01:37	Sunny	200
#15	M	27 Feb	11:02	01:45	Sunny, Partly cloudy	218
#16	M	27 Feb	11:05	01:43	Sunny, Partly cloudy	214
#17	M	27 Feb	11:07	01:22	Sunny, Partly cloudy	171
#18	M	27 Feb	13:15	01:47	Sunny, Partly cloudy	222
#19	F	27 Feb	13:40	03:02	Sunny, Partly cloudy	377

Images were recorded as mp4 videos in 1920x1080 at 30 FPS via a Samsung Galaxy J6+ (SM-J610F) Android OS mobile phone mounted on the windshield inside the vehicle. Then, these videos were converted to 720x480 size with the Python programming language and well-known Open-cv library for size reduction, and lower dimensional images were obtained by taking 5 samples per second (5 FPS). The pictures contained in these recorded and preprocessed videos were retrieved by decomposing them into sequential frames. These processes are summarized in Figure 3.2.

3.1.2 Labeling of the Images

Image labeling can be summed up as one of the processes of specifying the location of the objects in each image (bounding box labeling for object detection), specifying the class of the image (classifying the image), or specifying which pixels belong to which object (segmentation) in the context of supervised learning. Regarding the proposed framework, there are two image labeling tasks performed. The first is the bounding box labeling for the Object Detector, and the second is the classification labeling for the Driver Gaze Classifier. Within the proposed framework, it is possible to talk about

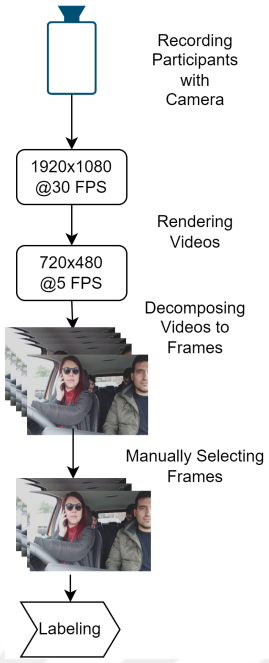


Figure 3.2 The scheme of the data collecting

a third labeling task, but it's not an image labeling task. Instead, it's a data labeling task made up of the normalized coordinates of the centers of objects.

In the context of object detection, image labeling is the act of drawing a rectangle around the desired image object for detection. Different views on how this rectangle should be defined in the literature have led to the formation of labeled data in different formats. For instance, while the label format published for the COCO dataset [49] is in the format that shown in (3.1), Pascal VOC [50] prefers the format shown in (3.2). The format that YOLO [51] uses is designed regardless of the image size. Accordingly, the coordinates of the object's center point are proportional to the horizontal and vertical lengths. Likewise, the width and height of the object are also proportional to the horizontal and vertical length. Equation (3.3) shows the necessary calculations to create this format.

$$B_{Box}(x_{top-left}, y_{top-left}, width, height) \tag{3.1}$$

$$B_{Box}(x_{top-left}, y_{top-left}, x_{bottom-right}, y_{bottom-right}) \tag{3.2}$$

$$B_{Box}(\frac{x_{center}}{width_{image}}, \frac{y_{center}}{height_{image}}, \frac{width_{object}}{width_{image}}, \frac{height_{object}}{height_{image}}) \tag{3.3}$$

YOLO-formatted labeled data were created for the Object Detector in the proposed framework concept. 14 detection classes have been determined. When the preprocessed videos were broken down into images, approximately 15300 images were obtained. 1660 of the most distinctive samples were chosen by manually and prepared for labeling. At the conclusion of the labeling procedure, data with the

distributions presented in Table 3.2 were acquired. This data was initially resized to a 416x416 resolution and then histogram-equalized. As depicted in Figure 3.3, it was subsequently exposed to a number of data augmentation operations. By rotating, sheering, saturating, exposing, and adding noise to the original image about three times, new images were formed. Data augmentation is applied only to train and test sets for data divided into 70% train, 15% test, and 15% validation. The validation set has not been subjected to any data augmentation process. These steps resulted in the creation of training data of around 4000 images from 1660 frames.

Table 3.2 The distribution of labeled Object Detector data

Class	**The number of Annotation**	**Annotation per Image**
PassengerFace	2369	1,43
DriverFace	1656	0,99
DriverHand	1589	0,96
Cellphone	1128	0,68
Cigarette	1058	0,64
PassengerHand	761	0,46
DriverHandWphone	657	0,40
DriverHandWcigarette	600	0,36
DriverSmoking	525	0,32
DriverPhoneCall	460	0,28
PassengerHandWphone	366	0,22
PassengerPhoneCall	211	0,13
PassengerHandWcigarette	187	0,11
PassengerSmoking	92	0,06
Total number of Images	**1660**	

The data set used to train the Driver Gaze Classifier structure is made up of images that were taken from the previous data set by cutting out the "DriverFace" bounding box. For each cropped image, it was manually labeled whether the driver was looking at the road or not. Thus, a two-class training dataset is created, which includes whether the driver is facing forward or in another direction. As in the previous step, this dataset is derived from 1660 original images, but in the training phase, images that have passed through various multiplexing stages, which were produced in the previous step, were used. These images were applied to the classifier by obtaining the information from which original image was produced during the training and examining the class in which the original image was included. Thus, the Driver Gaze Classification training set, which divided into 70% train, 15% test, and 15% validation, was created. The representation of these operations are shown in Figure 3.4.

In order for the LSTM structure to be trained, the time context should not be lost in

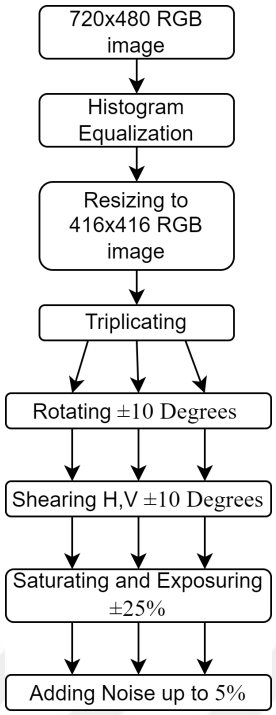


Figure 3.3 The scheme of the preprocessing and data augmentation

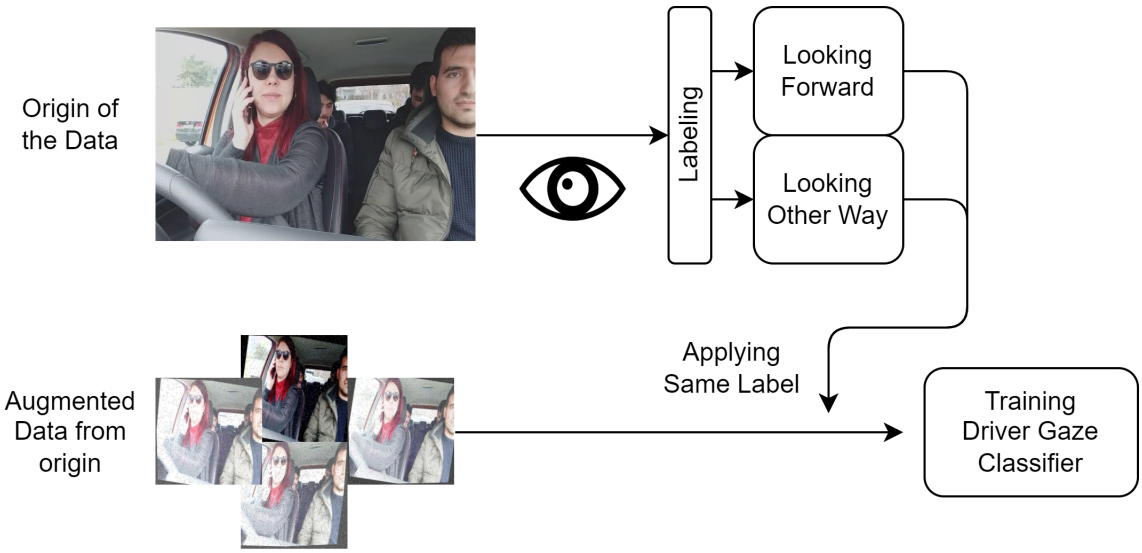


Figure 3.4 Driver Gaze Classifier data preparation for training.

the data to be created. Therefore, while generating the data to be used for training the LSTM network, the 1660 images selected at the beginning were not used, but images decomposed directly from the videos were used. In order to create this data, first of all, the Object Detector structure must be run and it must detect the predefined classes in these images. The resulting center point coordinates will then form the entries of the LSTM network. While the center point data created for each detected class is processed into the relevant column in the LSTM training data, the data of the undetected classes is recorded as zero. After the input data is generated, the images are manually analyzed and the information about whether the driver uses a mobile phone, smokes, or looks at the road is marked in the output column. So, the LSTM training dataset is made, which keeps track of the time context.

3.2 Proposed Framework

Deep learning is useful for determining whether a driver is looking at the road while driving, as well as for detecting mobile phone or cigarette use. In light of the knowledge in the field of computer vision, image classification, object recognition, and image segmentation algorithms have been and continue to be developed. As camera sensor technology advances, both its cost and the cost of searching for answers to issues by processing images with a camera decreases. Today, image processing techniques are utilized in numerous industries, including biomedicine, transportation, agriculture, and astronomy.

The first method that comes to mind for a system design that can be developed with flexible software that works with a single hardware infrastructure that is open to development for the problem for which a solution is sought within the scope of the thesis is the inventive use of image processing techniques and deep learning. However, the computational expense of such systems has also become a disadvantage. Therefore, the system must be optimally constructed in terms of both computing resources and memory use.

Although RNN structures that can process temporal information can also be used for image processing, they are very expensive methods in terms of computational cost. Therefore, designing a system using the capabilities of RNN networks in an optimal way has led to the development of the proposed method. Accordingly, examining only the points moving in a certain pattern in temporal space is a cheaper way to examine the entire image. For this reason, primarily object detection algorithms were developed. If some predefined objects (Driver Face, Driver Hand, Cigarette, Driver Hand with a Cigarette, etc.) are detected by object detection algorithms

and the center coordinates of these objects are rearranged together with temporal information, feeding an RNN structure, both the RNN's capabilities will be utilized and the computational cost is reduced.

YOLOv5 versions are the preferred solutions for object detection problems due to the amount of adaptive pretrained models, variety of their complexity, usability and excellent performance [52–54]. The choice of YOLOv5 as Object Detector was greatly influenced by its low computing complexity and excellent performance in object recognition. Because the YOLOv3 network is sensitive to the initial cluster center, the authors of [52] apply a k-means clustering approach to speed up the network. In comparison to the original structure, this method has improved the YOLOv3 network. It has also been revealed that using the FPGA-based PYNQ Z2 platform and the Movidius NCS accelerator to create the YOLO network results in high FPS and decent detecting performance [53]. On the Nvidia Jetson TX1 embedded system, the authors of [54] developed a YOLO-based network and achieved up to 17.85 FPS with 61.87 percent inference frequency while reducing the number of parameters from 48.2M to 17.1M. The YOLOv5 network has been deemed viable for all of the reasons stated above.

Since it is thought that the LSTM network can be effective in the decision-making process in the estimation of the distraction level and interactions with the mobile phone, it has been decided to add the detection of whether the driver is facing the road or not to the framework with the help of a ResNet classifier. The ResNet structure, as used in YOLOv5, can be used at the point where many classifiers are needed. Thanks to its 18, 34, 50, 101, and 152 layered different structures, it is preferred in many applications due to its flexibility between performance rate and computational cost. In light of these considerations, Figure 3.5 illustrates the proposed framework's structure in detail. According to this framework, the image from a camera that is attached to the windshield is first fed into the object detection structure, which is chosen as the Yolov5 network. Yolov5 network recognizes the driver's face, driver's hands, mobile phone, and cigarette on the image, extracts their locations, and provides the LSTM network their center point coordinates. In addition, after detecting the driver's face, this image frame is extracted and supplied to the Driver Gaze Classifier. After Driver Gaze Classifier detects whether the driver is looking at the road or not, it sends this information in binary to the LSTM network. The LSTM network takes in the center coordinates of these objects plus information about whether or not the driver is looking at the road, and produces three regressions that indicate whether or not the driver is smoking, texting, or chatting on a cell phone. These outputs are transformed to binary outputs by comparing them to the 0.5 threshold value, and they are multiplied by a number of coefficients to determine the distraction level (δ). These coefficients are

0.30 for output associated with mobile phone use (α), 0.20 for output associated with smoking (β), and 0.30 for output associated with texting (λ). In addition, the driver's gaze toward the road is multiplied by 0.20 ($\sigma = 0$ if looking at the road, otherwise 1). Thus, the distraction level (δ) can be calculated as in equation 3.4.

$$\delta = (0.30\alpha + 0.20\beta + 0.30\lambda + 0.20\sigma)x100 \tag{3.4}$$

When the framework shown in Figure 3.5 is closely examined, it becomes clear that the system does more than just identify phenomena that lead to driver distraction; it also alerts the driver audibly, sends this status information to remote computers via an internet connection, and sends notifications to other units via SMS.

The system evaluates the distractions it finds (such as texting, talking on the phone, or smoking), and then decides to either warn the driver and send notifications to authorized centers. The system includes counters to accomplish this. It audibly alerts the driver to any sources of distraction it finds. The interval between two sounds for the same kind of distraction is 30 seconds. During this time, the detected situations are not audibly warned about, but the system reserves each one and sends the data to the web-based monitoring application using the MQTT protocol.

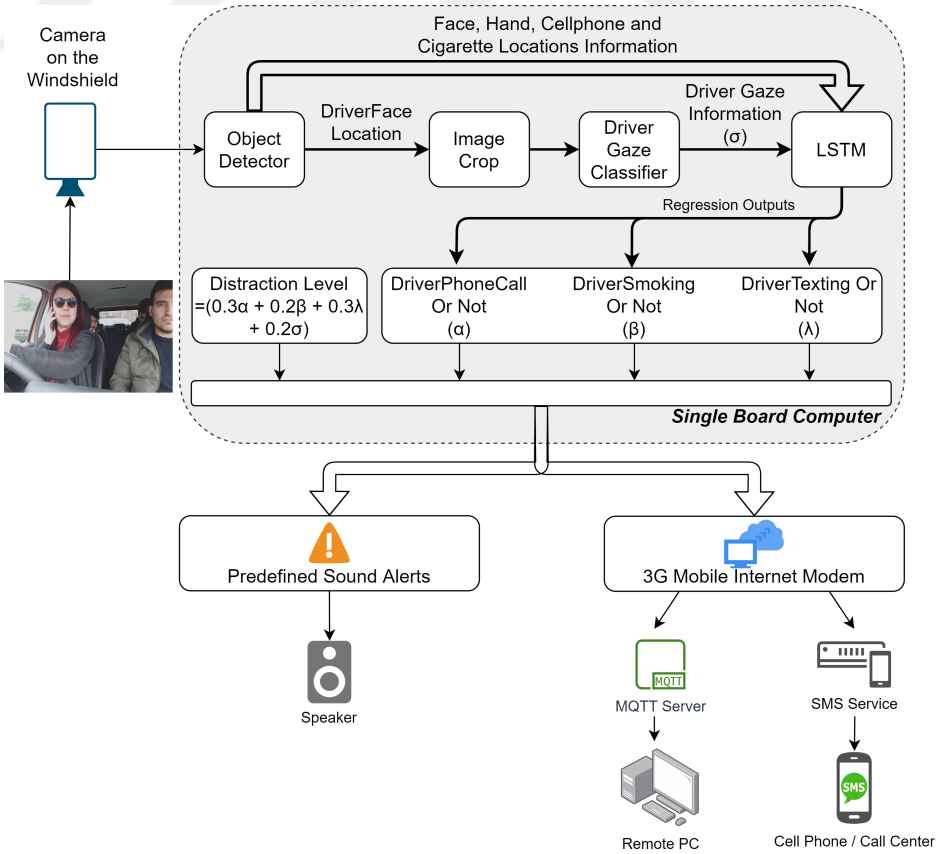


Figure 3.5 The concept of proposed framework.

The web-based monitoring application can be accessed remotely using a standard web browser. As the system is still undergoing testing, no cyber security measures have been implemented, and anyone with a web interface link can connect to the monitoring system. To remotely monitor the system, the service described in [55] was used, which may be used for free for 30 days and supports MQTT and other comparable protocols. Figure 3.6 depicts a dashboard constructed with this web-based remote monitoring software.

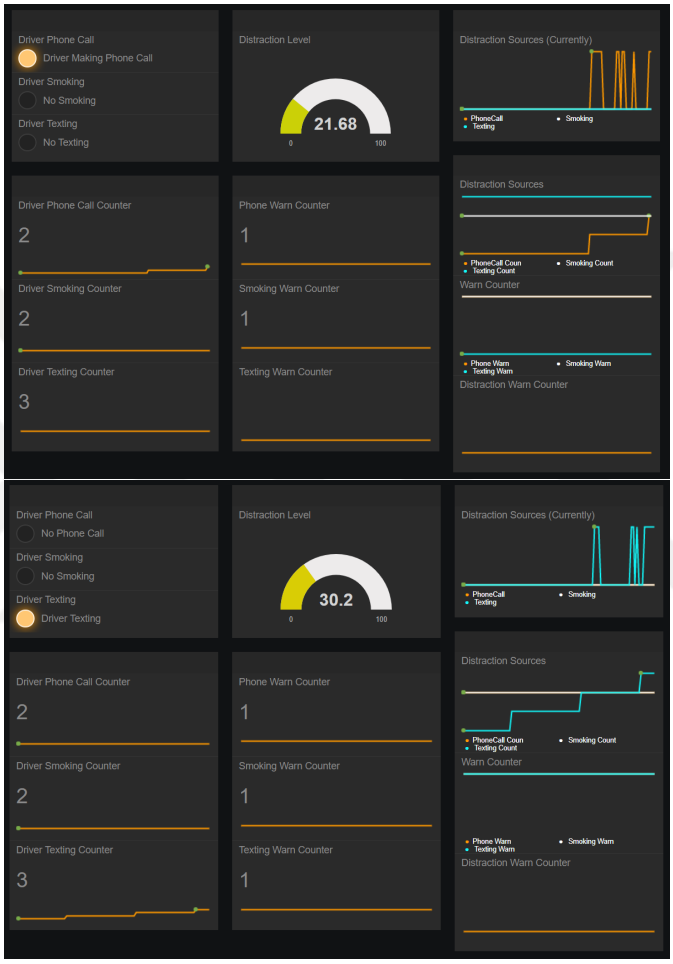


Figure 3.6 The web-based remote monitoring dashboard.

The work of the proposed framework proceeds according to a certain cycle. This cycle begins by taking an image from the camera. Next, the object detector detects the driver's face, driver's hands, mobile phone, cigarette, the driver's talking on the cell phone, and the driver's smoking in this image. After all these objects are detected, the part of the image containing the driver's face is cut and sent to another CNN algorithm (driver gaze classifier). At the same time, the center coordinates of the detected objects are accumulated to be sent to the LSTM network. The LSTM network provides 3 regression outputs. This regression output is multiplied by certain coefficients as shown in equation 3.4 to form the distraction level. At the same time, DriverPhoneCall,

DriverSmoking, and DriverTexting boolean outputs are obtained by comparing them with the 0.5 threshold value.

After obtaining the DriverPhoneCall, DriverSmoking, and DriverTexting boolean values and the distraction level floating point value, the system operates according to the following rules:

- If the driver makes a mobile phone call, the system will issue an audible warning: "Mobile phone call while driving is both dangerous and prohibited."
- After this warning, even if the system detects a mobile phone call again, it does not give an audible warning for 30 seconds, but this information is sent to the web-based monitoring system. Additionally, a counter is incremented by 1 to indicate that the system has issued a warning about a cell phone call.
- When the counter mentioned in the previous article reaches 3, an audible warning is given: "Please do not make a phone call. If you continue, a notification will be sent to the competent authorities." It also increments another counter related to this second level warning by 1.
- When the second level warning's counter reaches 2, the same audible warning is made again, but this time an SMS is sent to a predetermined phone number with the following message: "The driver of the vehicle with license plate number *** is on a mobile phone while driving."

The above system loop is also applied in cases where the driver's smoking is detected and her/his mobile phone texting is detected while driving. In addition to continuing this loop, the system calculates and transmits the distraction level and other parameters to the web-based monitoring system. Moreover, the system issues the following warnings regarding the distraction level:

- If the distraction level exceeds 40%, the system will sound an audible warning: "Your distraction level is too high. Please give your full attention to the road."
- If it makes this warning 2 times, again depending on a counter, the system will send an SMS to a predetermined mobile phone number with the following message: "The level of distraction of the driver of the vehicle with license plate *** is too high."

After making these warnings, the system resets the relevant counters. All parameters detected and calculated in each cycle are sent to the web-based monitoring

application. After the processes are completed, a new image is taken from the camera and the cycle continues. This operation continues until the power of the system is cut off (until the vehicle's engine is stopped and the ignition key is turned to 0 position) or until the "q" key is pressed if there is a keyboard connected to the system. The online text to speech software specified in [56] was used to create all voice alerts used by the system for warning.

4
EXPERIMENTAL RESULTS

4.1 Experiments of Object Detector (YOLOv5)

A data set was constructed in this study by processing photos obtained with the use of a mobile phone mounted on the windshield, and this data set was utilized to train the suggested framework. In the initial stage of the framework, the Yolov5s network is trained as an object detector. The dataset was manually labeled for 14 different classes, and the results of the YOLOv5s network trained on this labeled data were investigated. There has been no class downgrading in order to understand the quality of the created dataset. Thus, the training data of Experiment#1 was created. The model was created using the Pytorch framework and run entirely on the Google Colab platform, which features powerful GPUs and a simple-to-configure environment. Figure 4.1 depicts the precision-recall curve after 100 training epochs.

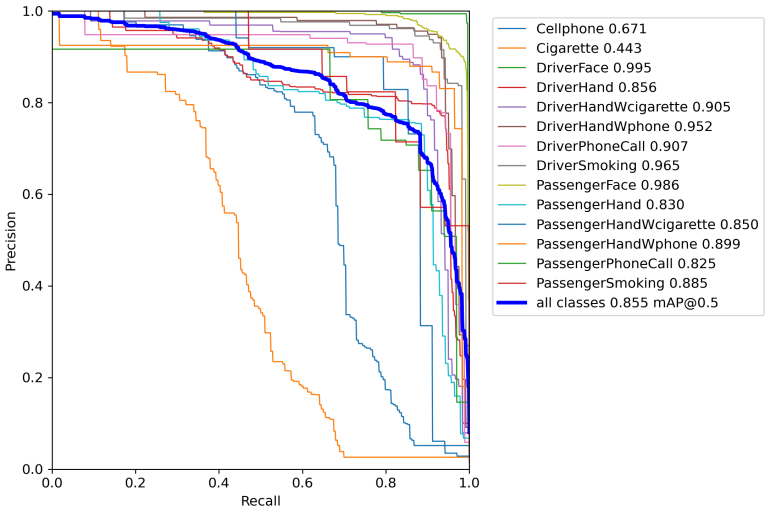


Figure 4.1 Experiment#1, the precision-recall curves of all classes.(The larger the area under the graph, the better the results.)

According to Figure 4.1, it shows that the object detector is working successfully with the overall mAP 0.855. However, due to its small size and often obstructed by other objects, the detection performance of cigarettes is relatively low. The second from

last, the Cell phone class, also had relatively lower performance for almost the same reasons. The relatively better performance of the other classes is promising for the success of the proposed method. By blending LSTM with temporal information, it can establish relationships between these classes and eliminate the influence of some ill-defined classes.

On the other side, because the system is solely concerned with driver distraction, the data set was reorganized, the passenger classifications were deleted. Thus, the training data of Experiment#2 was created and the object detector structure was retested. As a result, it was discovered that, while categorization performance improved slightly, there was no meaningful improvement (Figure 4.2). The confusion matrix for this experiment may be found in Figure 4.3.

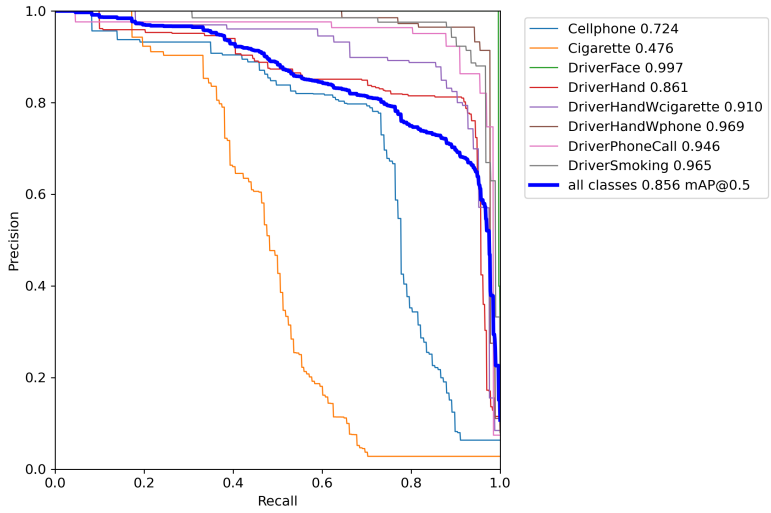


Figure 4.2 Experiment#2, the precision-recall curves of reduced classes (Focused on driver, the larger the area under the graph, the better the results).

According to the results obtained in Experiment#1 and Experiment#2, it is seen that the performance of mobile phone and cigarette detection is lower than other classes. In order to improve this, in addition to the previously created training data, images obtained by searching for mobile phone and cigarette images on the internet were added. By labeling these images and applying the data augmentation techniques applied in Figure 3.3, 2190 cigarette data from 730 cigarette samples and 597 mobile phone data from 199 mobile phone samples were produced, and these data were added to the training data. Thus, data for Experiment#3 and Experiment#4 were created. Experiment#3 was carried out with training data, including all classes as in Experiment#1. Experiment#4, on the other hand, was carried out with the training data created by removing the classes related to passengers, as in Experiment#2. Figure 4.4 and Figure 4.5 show the precision-recall curve obtained in Experiment#3 and Experiment#4, respectively. As a result, later additions of mobile phone and

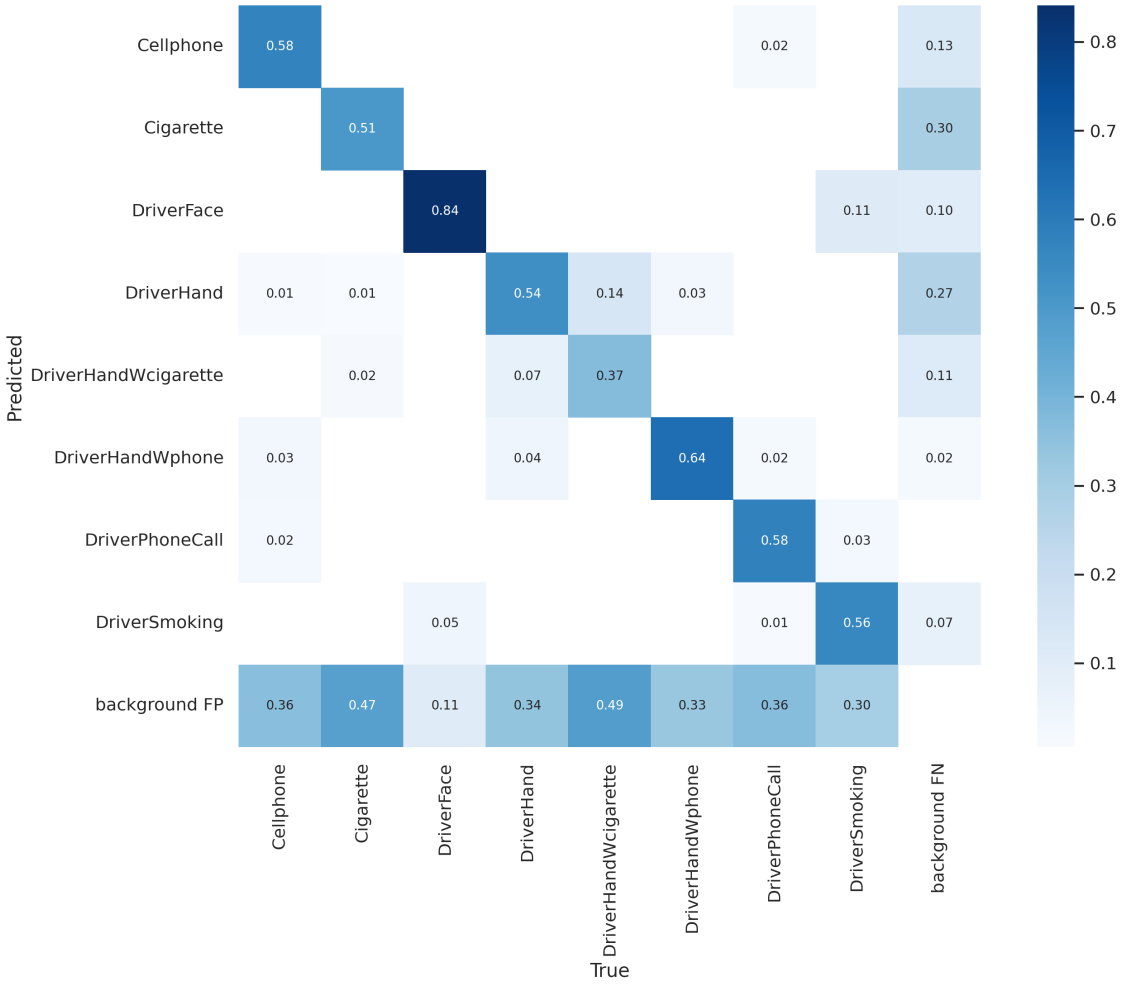


Figure 4.3 The confusion matrix of the Experiment#2 in YOLOv5 (High values on the diagonal indicate high classification success).

cigarette data improved the performance of these classes. Also, confusion matrix showing the performance of Experiment#4 is given in Figure 4.6.

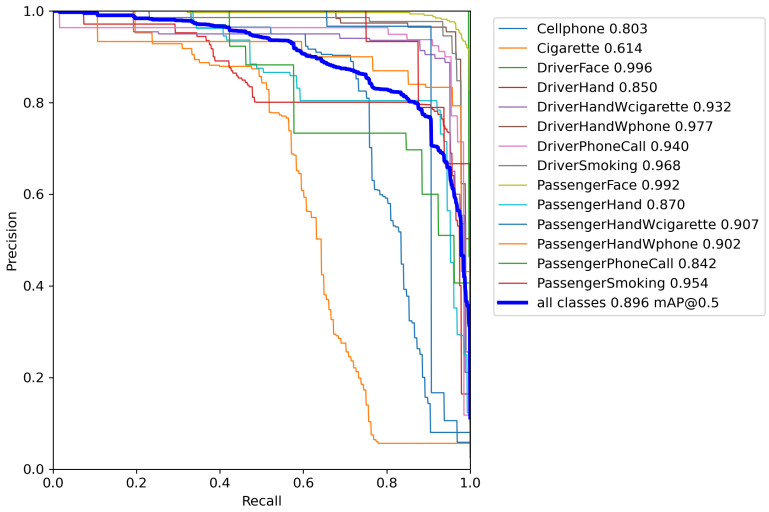


Figure 4.4 Experiment#3, the precision-recall curves of all classes.(The larger the area under the graph, the better the results.)

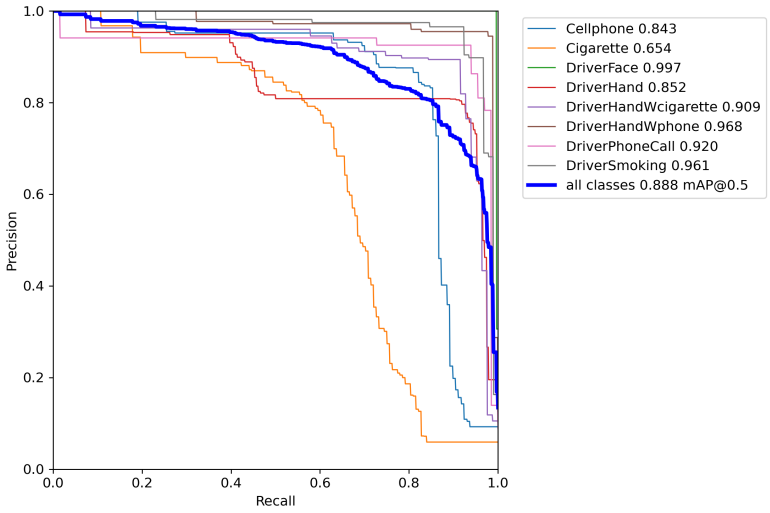


Figure 4.5 Experiment#4, the precision-recall curves of reduced classes (Focused on driver, the larger the area under the graph, the better the results).

The performance metric used in Figures 4.1, 4.2, 4.4, and 4.5 is mAP@0.5. This metric is the mean value of the average precision calculated using the intersection over union (IoU) threshold of 0.5. Precision is a measure of prediction accuracy, or the proportion of correct predictions. Precision is defined as the number of true positives over the number of true positives plus the number of false positives as in equation 4.1. Recall, on the other hand, is defined as the number of true positives over the number of true positives and the number of false negatives, as in equation 4.2. Intersection over Union (IoU) is a metric that shows how accurately an object detector predicts the location of

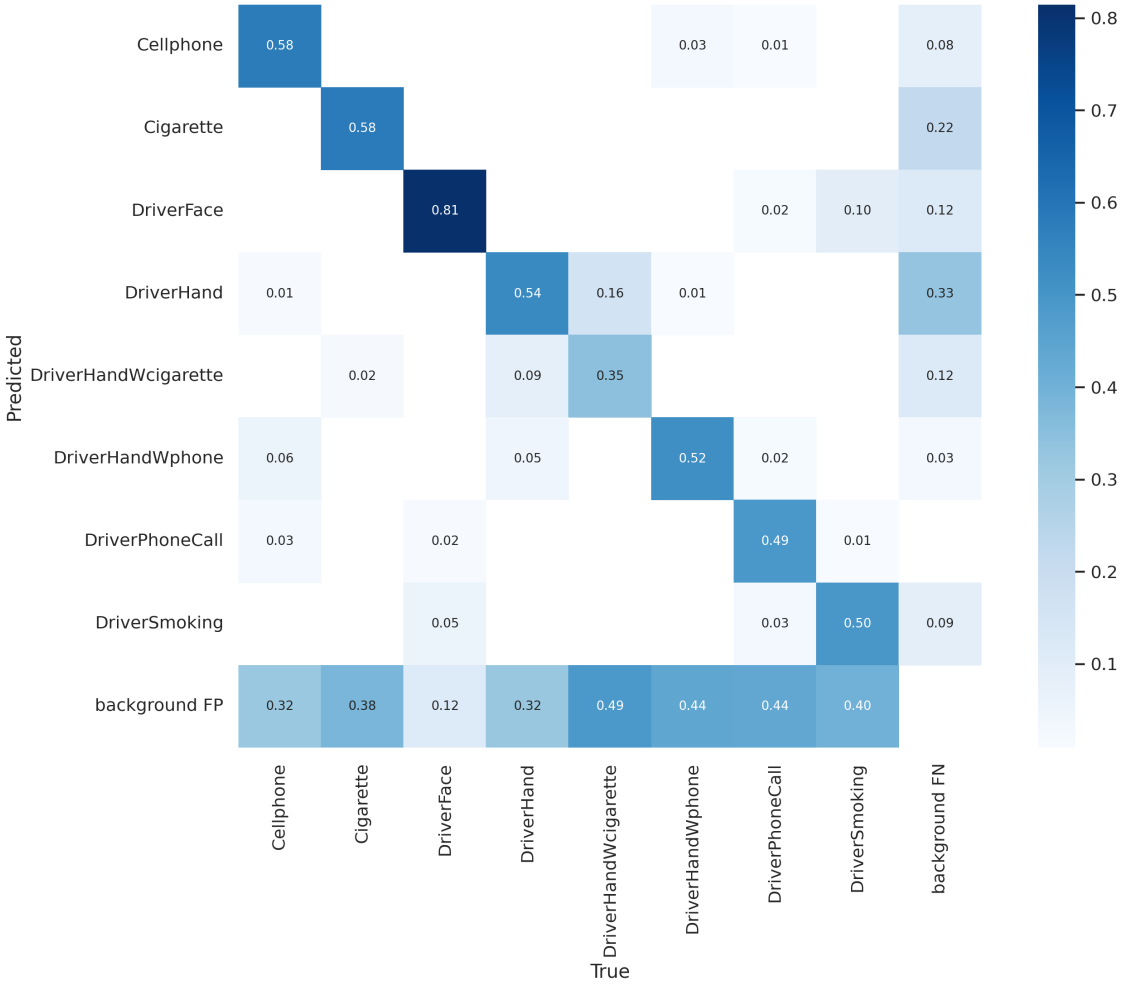


Figure 4.6 The confusion matrix of the Experiment#4 in YOLOv5 (High values on the diagonal indicate high classification success).

the object. It is the ratio of the intersection area of the estimated bounding box and the actual bounding box to the junction area. The IoU is used to calculate the true positive and false positive numbers in this calculation. If IoU threshold 0.5 is selected (which is 0.5 in mAP@0.5), if an object's IoU score exceeds 0.5 on location detection, it will be counted as true positive, otherwise it will be counted as false positive. The area under the precision-recall curve is known as the AUC (area under curve) and forms the mean average precision (mAP).

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{4.1}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.2}$$

In the confusion matrix of Figures 4.3 and 4.6, higher values on the diagonal indicate a high classification success. Values outside the diagonal indicate misclassification results. When both results are compared, it is seen that Experiment 4 is successful in the classification of cigarettes. However, it is seen that the performance of other classes is higher in Experiment2. In addition, it was observed that the results in Experiment 2 were better when the misclassified values were examined. For this reason, the model trained in Experiment2 was used for the object detector stage.

The high values on the Diagonal in Figure 4.3 indicate a high classification success. However, as can be seen, there are cases where the predicted class and the ground truth class are different. For example, ground truth is Cell Phone, while 58 percent of the estimates are Cellphone, 36 percent are background, 3 percent are DriverHand with a Phone, and the other 2 percent and 1 percent are Driver Phonecall and Driver Hand. It is thought that such classification errors can be eliminated with the LSTM network. The two of the predicted classes are shown in Figure 4.7.

4.2 Experiments of Driver Gaze Classifier

The Driver Gaze Dataset was created from the Driver Face location information from the object detector by cutting and manually tagging these bounding boxes. As a result, a two-class classifier structure was established and training was carried out on the ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152 networks. The training was repeated 5 times by emptying the RAM memory and initialization weights choosed randomly each time, and the average results produced by each network were shown in Table 4.1. According to the results in Table 4.1, Resnet152 network gave the best accuracy and sensitivity score. However, the complexity of the Resnet152 network is very high. When considering running on an embedded system, it is important that the



Figure 4.7 The detection examples of YOLOv5 Object Detector.

Table 4.1 Training results of ResNet networks trained for Driver Gaze Classifier implementation.

Network	Trainable Parameters	Accuracy	Sensitivity	Specificity
Resnet18	11,177,538	0.86344	0.84818	0.879
Resnet34	21,285,698	0.89022	0.88192	0.89876
Resnet50	23,512,130	0.88778	0.86264	**0.91358**
Resnet101	42,504,258	0.88172	0.88434	0.879
Resnet152	58,147,906	**0.8951**	**0.89396**	0.8963

computational cost is low and accordingly it gives a good test result. For this reason, it was decided to use the Resnet50 network. Figure 4.8 shows the training graph of the Resnet50 network, and Figure 4.9 shows the confusion matrix results. A 3-fold cross validation has been applied to better test the performance of the Resnet50 network. Table 4.2 shows the results obtained when cross validation is applied. In Figure 4.10, the operation of the Driver Gaze Classification stage after the detection process in the original videos taken by the participants is shown. The software automatically adds an estimate of the direction the driver is facing to the image.

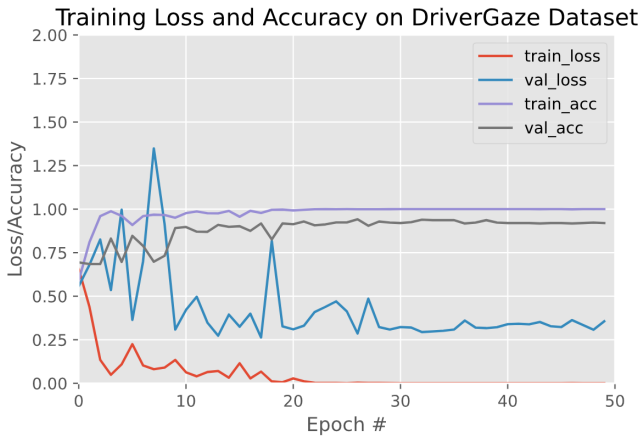


Figure 4.8 The train performance for 50 Epochs of Driver Gaze Classifier.

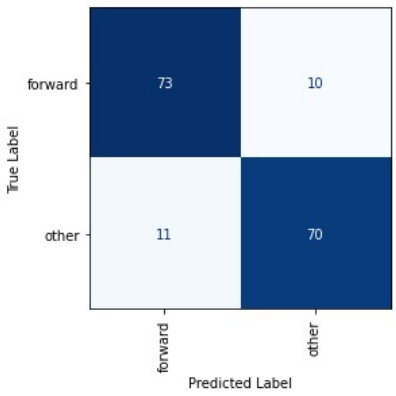


Figure 4.9 The confusion matrix of Resnet50 network for Driver Gaze Classifier.

Table 4.2 Training results of ResNet50 3-fold cross validation for Driver Gaze Classifier.

Fold	Accuracy	Sensitivity	Specificity
Fold#1	0.86234	0.85313	0.87945
Fold#2	0.89582	0.85152	0.88776
Fold#3	0.87898	0.86354	0.85438
Avg	0.87904	0.85606	0.87386

Looking at Table 4.1 and Table 4.2, it is seen that accuracy, sensitivity, and specificity are used as performance metrics. Of these, accuracy represents the proportion of correctly guessed classes. Sensitivity is known as true positive rate and is calculated as in equation 4.3. Specificity is known as true negative rate and is calculated as in equation 4.4. The sensitivity of a test is defined as its ability to detect real positives, while its specificity is defined as its ability to detect false negatives.

$$Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.3}$$

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives} \tag{4.4}$$



Figure 4.10 Prediction of driver gaze classification after detection of driver face. The estimation of the direction the driver is facing is automatically added to the image by the software.

4.3 Experiments of LSTM network

In the first stage, the center coordinates of the DriverFace, DriverHand (two can be detected, right and left), DriverHandWithCigarette, DriverHandWithPhone, Cellphone, Cigarette, DriverSmoking, and DriverPhoneCall classes detected by the object detector are x (horizontal) and y (vertical) respectively. It creates 18 floating point numbers in the range of 0 to 1. If the driver's gaze classification result is added to this, 19 floating point numbers are created to provide input to the LSTM network. Although the Driver Gaze Classification result is binary (0 and 1), these values have been updated to 0.0 (if the driver is facing the other way) and 1.0 (if the driver is facing the road).

Since the object detector is not trained to distinguish between left and right hands, it can only detect the DriverHand. However, it may be important whether the driver's hand to be sent to the LSTM network is right or left-handed. In order to make this distinction, the left side, which is the side of the driver (left side of the image, the regions where the horizontal value is close to zero in the coordinates), is defined as the driver's right hand, and the remaining hand on the other side is defined as the left hand. In cases where only one hand of the driver is detected, it is calculated whether this hand is close to the middle or the leftmost part of the image, and if it is close to the middle zone, it is decided that it is the driver's left hand and information is sent to the LSTM network in this way.

The object detector may not always detect all defined classes. Therefore, the center coordinate will be created for the detected objects, but no center coordinate will be formed for the undetected objects. However, since an input must be sent to the LSTM network, the location information of these objects is sent as predefined values when the object detector cannot detect them (or when that object is not present in the scene). These values are indicated in Table 4.3.

Table 4.3 Default center coordinate values of the objects that are not detected in scene.

Object	x	y	Region on Image
Cellphone	0.0	1.0	Bottom Left
Cigarette	0.0	1.0	Bottom Left
DriverFace	0.0	0.0	Top Left
DriverHand (Left)	0.0	1.0	Bottom Left
DriverHand (Right)	0.0	1.0	Bottom Left
DriverHandWcigarette	0.0	0.0	Top Left
DriverHandWphone	0.0	0.0	Top Left
DriverPhoneCall	0.0	0.0	Top Left
DriverSmoking	0.0	0.0	Top Left

After the LSTM data was preprocessed, it was applied to a 3-layer many-to-many structure with 19 inputs and 128 hidden layers, yielding 3 regression outputs. The gathered photos from the 19 participants, a summary of which is provided in Table 3.1, were applied to the object detector structure trained with these images, and a total of 13056 rows and 19 columns were obtained by extracting the center coordinates and preprocessing. With this data, the LSTM network is trained for 100 epochs. Figure 4.11 shows the train loss graph.

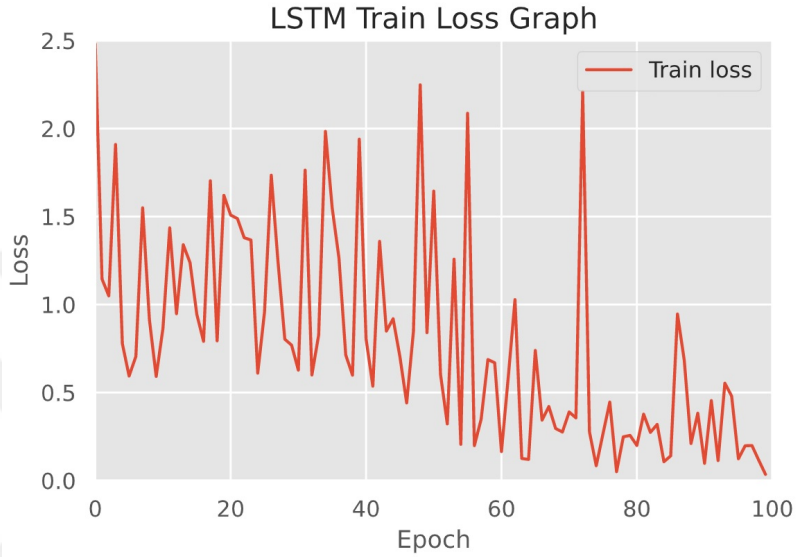


Figure 4.11 RMSE loss graph of the results of the LSTM network trained for 100 epochs

4.4 Real-time Experiments

The Jetson Nano device, with the software loaded for real-time experiments, is placed on the windshield of a vehicle. The power requirement of the system is provided by the power adapter shown in Figure 2.11 using the vehicle's 12v power outlet (cigarette lighter outlet). The first test performed was initiated by the system in such a way that it did not save any images. The recorded images were recorded over the OBS software running on another computer using the HDMI capture card. During the tests, the driver was asked to simulate a cell phone conversation, simulate cell phone texting, and simulate smoking.

When the operation of the system was observed, it was observed that the driver's movements were captured by the camera and reflected in the system with a delay of 6-7 seconds. However, it has been observed that this delay does not affect the detect performance of the system. The driver was asked to make phone calls, smoke cigarettes, and text on a mobile phone until he received a secondary level warning and the system was enabled to send SMS. Although it has been observed that the object

detector does not detect some objects and/or sometimes makes false detections, it has been observed that it can detect the situations of smoking, making mobile phone calls, and texting on the mobile phone and give the necessary warnings (see Figure 4.15). While these scenarios were being implemented, a video was recorded from a mobile phone, and this video was shared on the internet so that only those with the url could see it [57]. The system operating speed is up to 15 FPS before entering any detect loop. However, when it enters the detection cycle, the operation of the system slows down due to many sub-programs it runs in the background and drops to 1-2 FPS levels. Images of the first real-time test performed are shown in Figures 4.12, 4.13, 4.14, and 4.15. The image quality is poor because it is recorded from another computer with an HDMI capture card.

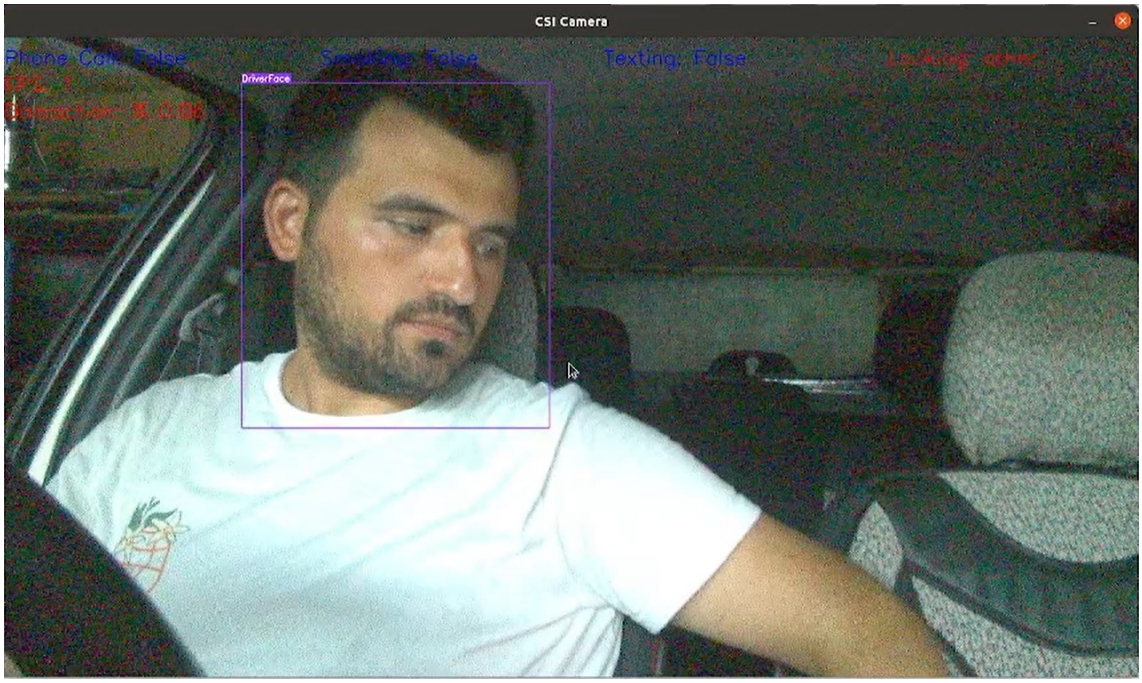


Figure 4.12 An image from the first real-time test.

In the second real-time evaluation, the system was modified to save screen output. In addition, the system's internet connection has been disconnected and the SMS sending function has been deactivated. A second vehicle was assembled, and testing began at noon (see Figure 4.16). There were no video recordings taken during these tests. The drivers were tasked with recreating the scenarios posed in earlier examinations. During the second test, the system's operating speed climbed up to 15 FPS before starting a detection cycle, then fell to 1-2 FPS during the cycle. In contrast to the first test, however, a clearer image was obtained because the screen output was captured directly by the system rather than utilizing an HDMI capture card. Figures 4.17, 4.18, 4.19, and 4.20 depicts second real-time examination images.

During real-time tests, it was observed that the Driver Gaze Classifier could not make

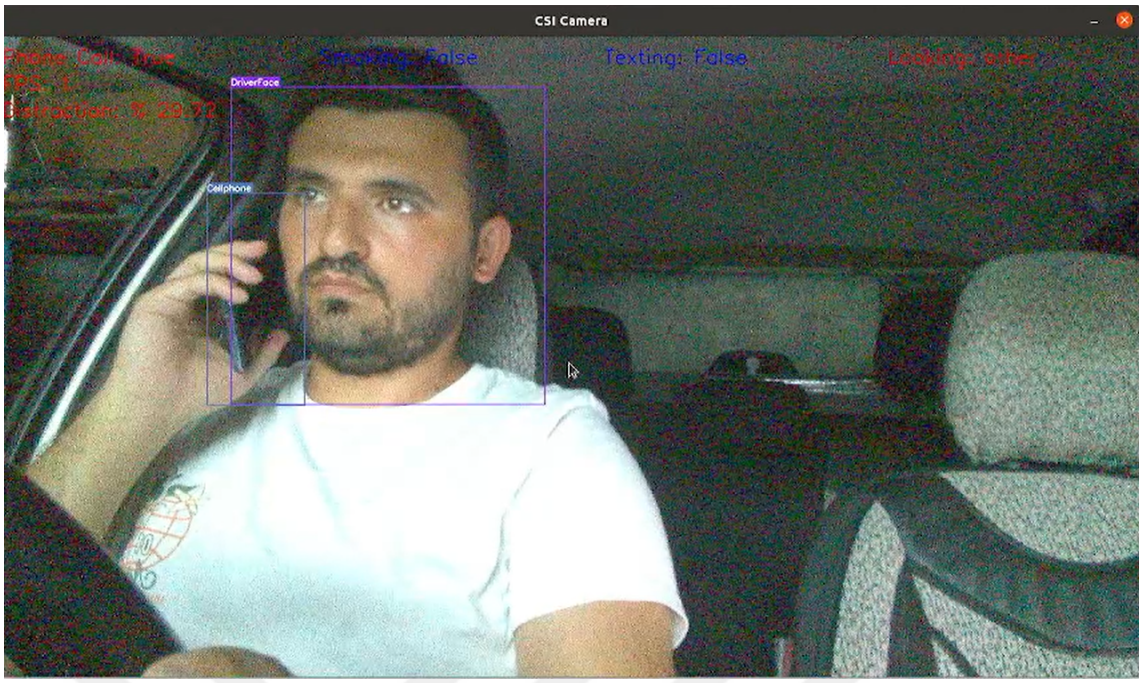


Figure 4.13 An image from the first real-time test (Phone Call: True).

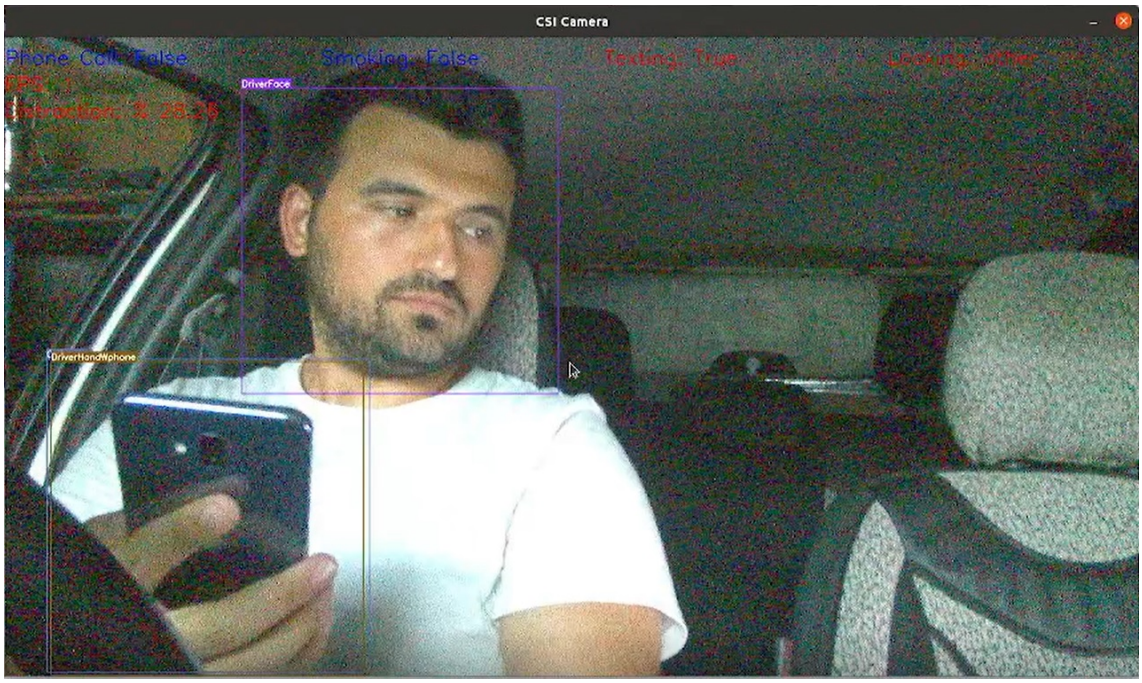


Figure 4.14 An image from the first real-time test (Texting: True).

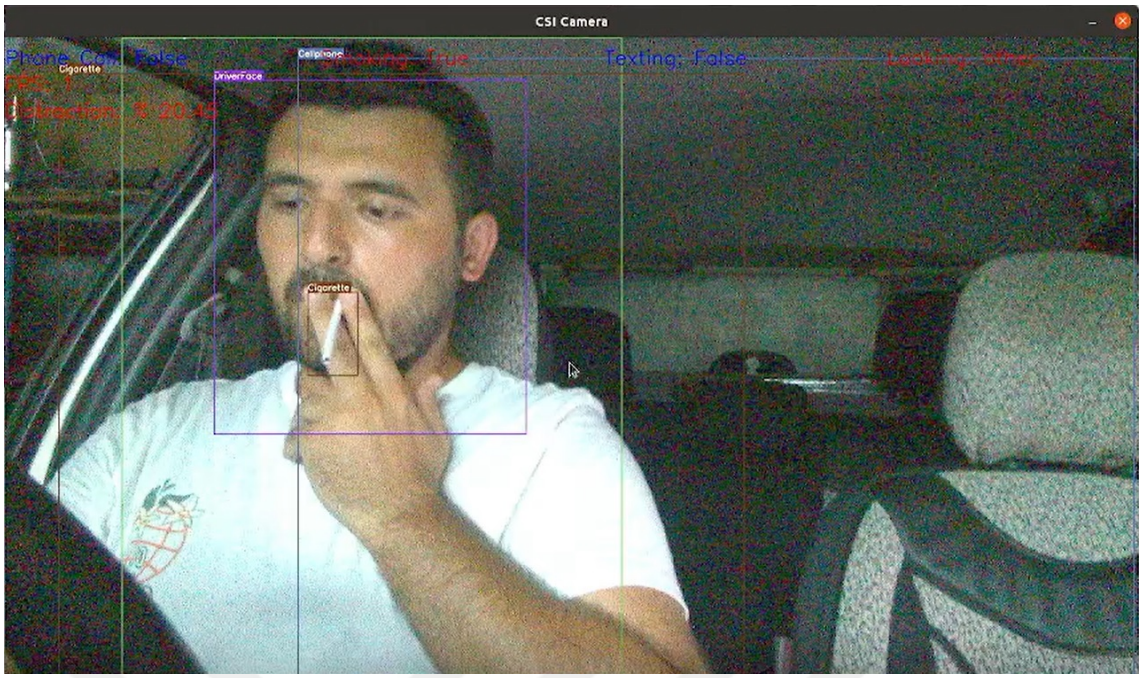


Figure 4.15 An image from the first real-time test (Smoking: True).



Figure 4.16 Docking to the windshield the system for the second real-time test.

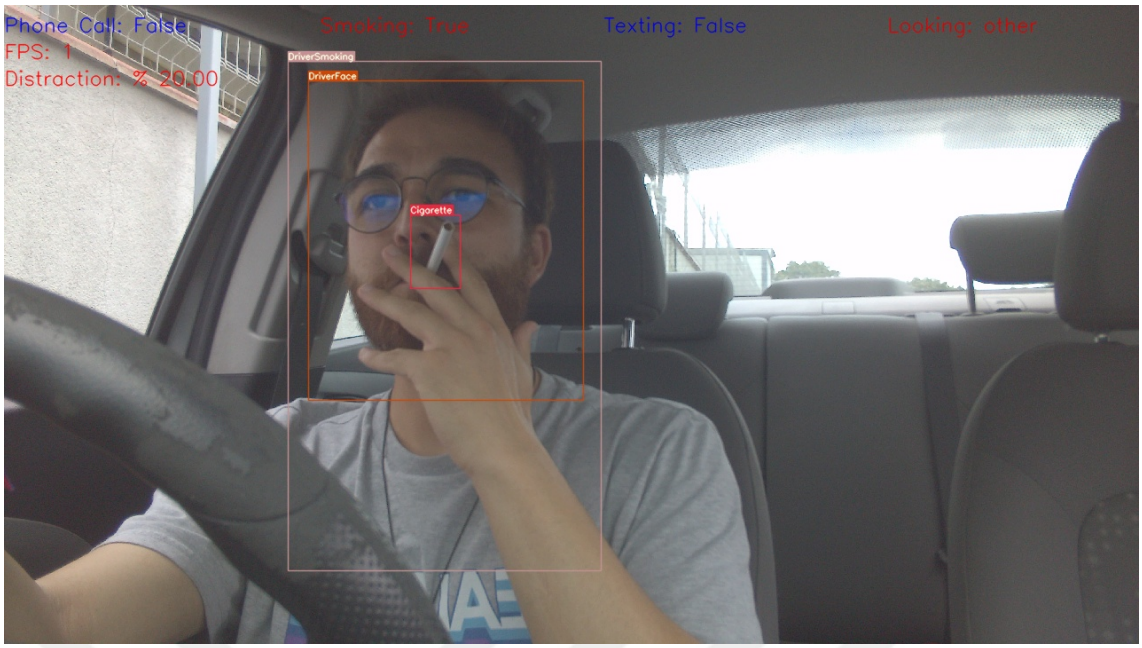


Figure 4.17 An image from the second real-time test.

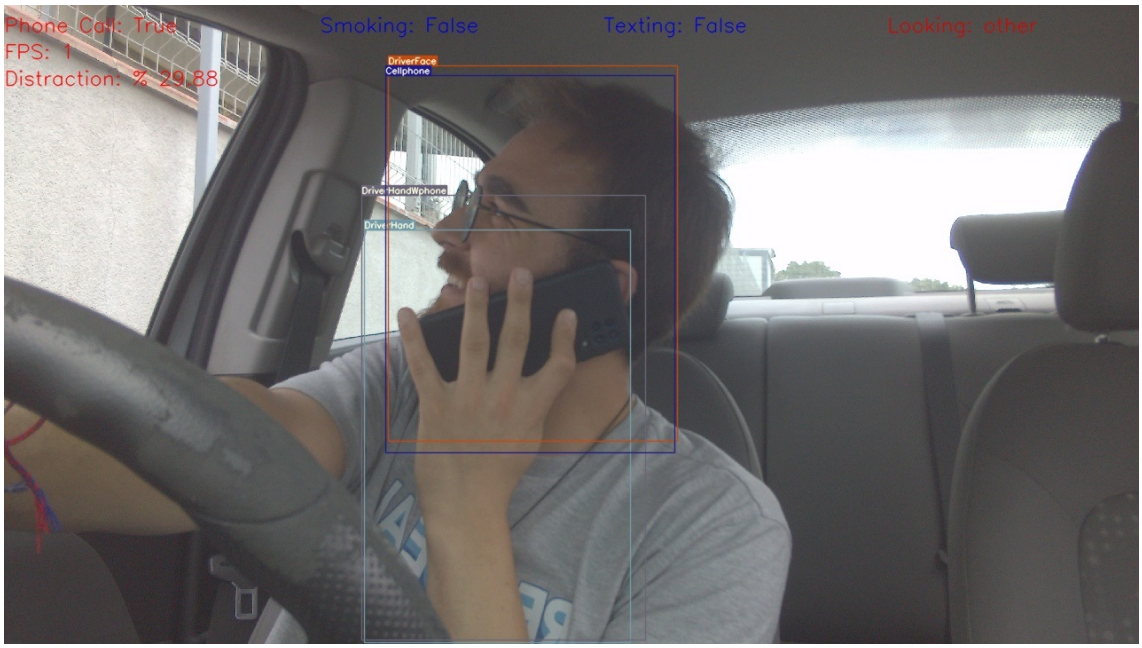


Figure 4.18 An image from the second real-time test.

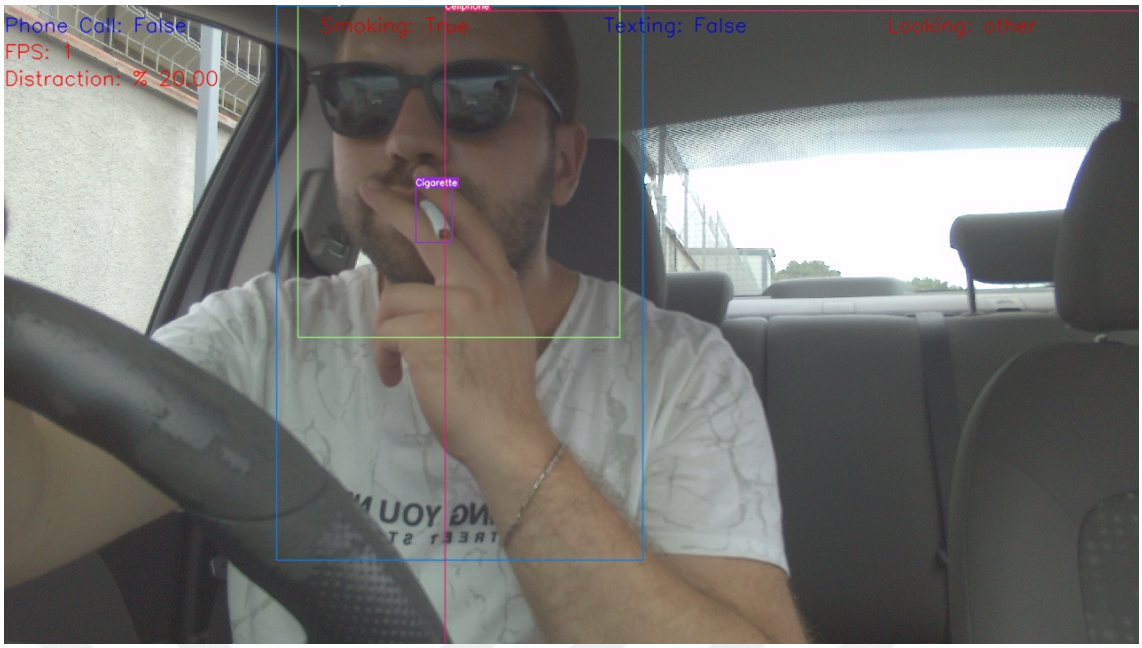


Figure 4.19 An image from the second real-time test.

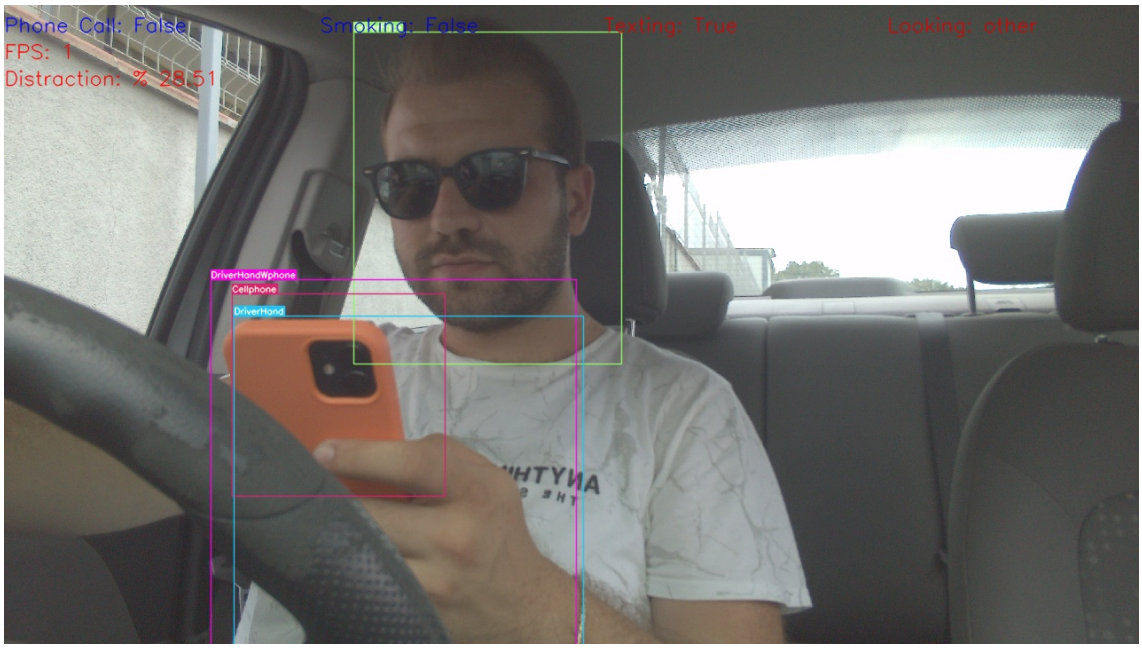


Figure 4.20 An image from the second real-time test.

accurate predictions. Whether the driver is looking at the road or not, after the driver's face is detected, it is printed out that the road is not looked at. The reason for this situation is that there is a high probability that there will be differences between the location of the camera that records the videos used in the training phase of the system and the location of the camera that records the images taken during the real-time test phase of the system. However, incorrect operation of the Driver Gaze Classifier did not result in the overall system making very wrong decisions.

5
RESULTS AND DISCUSSION

Within the scope of the thesis, it aims to contribute to the literature by presenting a system that can enable drivers to behave more carefully in traffic, although there are still many problems related to traffic accidents that are still waiting to be solved. In the concept of this line, a computer vision-based system has been proposed to detect mobile phone use and smoking, which are sources of driver distraction, by analyzing only camera images.

A well-prepared data set is needed in order for the system proposed in the thesis to work robustly. Therefore, a dataset based on the object detection approach has been prepared. For this dataset, participants of various appearances (beards, mustaches, hair, etc.), various genders, and various ages were studied as much as possible. With permission from the Academic Ethics Committee, data were collected in a parked car in pedestrian areas and then labeled by hand. Another data set derived from this data set was used to determine whether the driver was looking at the road or not. Finally, the training data was prepared manually so that the class coordinate information produced by the object detector and the driver gaze information produced by the Driver Gaze Classifier would form the inputs for the training of an LSTM network.

In the experiments, although the object detector has achieved relatively lower performance in detecting cigarettes and mobile phones, the performance of the classes that catch the driver's mobile phone call (DriverPhoneCall) and catch the smoking (DriverSmoking) is high. For this reason, the fact that these classes, which will constitute the input of the LSTM network, are detected with high performance gives a preliminary signal that the performance of the LSTM network will also be high. On the other hand, it gave very successful results in estimating the driver's looking direction in the Driver Gaze Classifier. Taking all of these results into account, it is likely that the performance of the system as a whole will be good enough.

REFERENCES

[1] W. H. Organization *et al.*, "World health statistics 2021," 2021.

[2] D. R. Mayhew, R. D. Robertson, S. Brown, W. Vanlaar, *Driver distraction and hands-free texting while driving*. Traffic Injury Research Foundation Ottawa, ON, 2013.

[3] "Distracted driving 2020," National Highway Traffic Safety Administration, National Center for Statistics and Analysis, Tech. Rep. DOT HS 813 309, May 2022.

[4] J. Stutts, D. Reinfurt, E. Rodgman, "The role of driver distraction in crashes: An analysis of 1995-1999 crashworthiness data system data.," in *Annual proceedings. Association for the Advancement of Automotive Medicine*, vol. 45, 2001, pp. 287–301.

[5] K. Hatoyama, M. Nishioka, M. Kitajima, K. Nakahira, K. Sano, "Perception of time in traffic congestion and drivers' stress," in *International Conference on Transportation and Development 2019: Smarter and Safer Mobility and Cities*, American Society of Civil Engineers Reston, VA, 2019, pp. 165–174.

[6] N. H. T. S. Administration *et al.*, "Preliminary statement of policy concerning automated vehicles," *Washington, DC*, vol. 1, p. 14, 2013.

[7] J. D. Lee, K. L. Young, M. A. Regan, "Defining driver distraction," *Driver distraction: Theory, effects, and mitigation*, vol. 13, no. 4, pp. 31–40, 2008.

[8] M. Née, B. Contrand, L. Orriols, C. Gil-Jardiné, C. Galéra, E. Lagarde, "Road safety and distraction, results from a responsibility case-control study among a sample of road users interviewed at the emergency room," *Accident Analysis & Prevention*, vol. 122, pp. 19–24, 2019.

[9] J. P. Foley, R. Young, L. Angell, J. E. Domeyer, "Towards operationalizing driver distraction," in *Proceedings of the 7th International Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, 2013, pp. 57–63.

[10] D. L. Strayer, J. Turrill, J. M. Cooper, J. R. Coleman, N. Medeiros-Ward, F. Biondi, "Assessing cognitive distraction in the automobile," *Human factors*, vol. 57, no. 8, pp. 1300–1324, 2015.

[11] C. Blaschke, F. Breyer, B. Färber, J. Freyer, R. Limbacher, "Driver distraction based lane-keeping assistance," *Transportation research part F: traffic psychology and behaviour*, vol. 12, no. 4, pp. 288–299, 2009.

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

[13] T. D'Orazio, M. Leo, C. Guaragnella, A. Distante, "A visual approach for driver inattention detection," *Pattern recognition*, vol. 40, no. 8, pp. 2341–2355, 2007.

[14] H. Zhang, M. R. Smith, G. J. Witt, "Identification of real-time diagnostic measures of visual distraction with an automatic eye-tracking system," *Human factors*, vol. 48, no. 4, pp. 805–821, 2006.

[15] Y. Artan, O. Bulan, R. P. Loce, P. Paul, "Driver cell phone usage detection from hov/hot nir images," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 225–230.

[16] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, C. P. Thor, "Driver cell phone usage detection on strategic highway research program (shrp2) face view videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 35–43.

[17] T. Hoang Ngan Le, Y. Zheng, C. Zhu, K. Luu, M. Savvides, "Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 46–53.

[18] N. J. Pedersen, S. Noureldin, "The second strategic highway research program," 2013.

[19] N. Das, E. Ohn-Bar, M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *2015 IEEE 18th international conference on intelligent transportation systems*, IEEE, 2015, pp. 2953–2958.

[20] Y. Artan, B. Balcı, A. Elihoş, B. Alkan, "Vision based driver smoking behavior detection using surveillance camera images," in *International Conference on Image Analysis and Processing*, Springer, 2019, pp. 468–476.

[21] D. H. Hubel, T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[22] K. Fukushima, S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, Springer, 1982, pp. 267–285.

[23] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] W. Zhang, K. Itoh, J. Tanida, Y. Ichioka, "Parallel distributed processing model with local space-invariant interconnections and its optical architecture," *Applied optics*, vol. 29, no. 32, pp. 4790–4797, 1990.

[25] Y. LeCun, C. Cortes, C. Burges, *The mnist database*, Available at `http://yann.lecun.com/exdb/mnist/` (2022/05/17).

[26] Y. LeCun *et al.*, "Comparison of learning algorithms for handwritten digit recognition," in *International conference on artificial neural networks*, Perth, Australia, vol. 60, 1995, pp. 53–60.

[27] S. Belongie, J. Malik, J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[28] L.-N. Teow, K.-F. Loe, "Robust vision-based features and classification schemes for off-line handwritten digit recognition," *Pattern Recognition*, vol. 35, no. 11, pp. 2355–2364, 2002.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[30] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[31] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[32] M. D. Zeiler, R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, Springer, 2014, pp. 818–833.

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

[34] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[35] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, PMLR, 2015, pp. 448–456.

[38] K. Weiss, T. M. Khoshgoftaar, D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[39] D. Erhan, A. Courville, Y. Bengio, P. Vincent, "Why does unsupervised pre-training help deep learning?" In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[40] Y. Zhou, R. Chellappa, "Ieee 1988 international conference on neural networks," 1988.

[41] D. Yu, H. Wang, P. Chen, Z. Wei, "Mixed pooling for convolutional neural networks," in *Rough Sets and Knowledge Technology*, D. Miao, W. Pedrycz, D. Ślzak, G. Peters, Q. Hu, R. Wang, Eds., Cham: Springer International Publishing, 2014, pp. 364–375, ISBN: 978-3-319-11740-9.

[42] F. Rosenbaltt, "The perceptron–a perciving and recognizing automation," *Cornell Aeronautical Laboratory*, 1957.

[43] V. Nair, G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[44] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, Citeseer, vol. 30, 2013, p. 3.

[45] Z. C. Lipton, J. Berkowitz, C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[46] C. Olah, *Understanding LSTM Networks – colah's blog*, [Online; accessed 2022-06-12], Aug. 2015.

[47] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] G. Jocher, A. Stoken, J. Borovec, C. A. NanoCode012, *et al.*, *Ultralytics/yolov5: V5. 0-yolov5-p6 1280 models, aws, supervise. ly and youtube integrations (2021)*.

[49] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[50] M. Everingham, L. Gool, C. K. Williams, J. Winn, A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010, ISSN: 0920-5691. DOI: 10.1007/s11263-009-0275-4. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4.

[51] ultralytics, *Github - ultralytics/yolov5: Yolov5 in pytorch > onnx > coreml > tflite*, Jun. 2022. [Online]. Available: https://github.com/ultralytics/yolov5.

[52] L. Zhao, S. Li, "Object detection algorithm based on improved yolov3," *Electronics*, vol. 9, no. 3, p. 537, 2020.

[53] S. Kaarmukilan, S. Poddar, *et al.*, "Fpga based deep learning models for object detection and recognition comparison of object detection comparison of object detection models using fpga," in *2020 Fourth international conference on computing methodologies and communication (ICCMC)*, IEEE, 2020, pp. 471–474.

[54] M. J. Shafiee, B. Chywl, F. Li, A. Wong, "Fast yolo: A fast you only look once system for real-time embedded object detection in video," *arXiv preprint arXiv:1709.05943*, 2017.

[55] *Visualize the internet of things*. https://freeboard.io/, Accessed: 2022-06-12.

[56] *Text to speech turkish*. https://www.narakeet.com/languages/text-to-speech-turkish/, Accessed: 2022-06-12.

[57] *Drive securely - video presentation of msc project - nvidia jetson nano*. https://youtu.be/oBDd8hGDOBY, Accessed: 2022-08-25.

A
ETHICS COMMITTEE APPROVAL

Yıldız Technical University Ethics Committee, with its decision on 06.05.2020, approved the data collection within the scope of the thesis titled Embedded System Design for Drive Security.

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
Araştırma ve Planlama Rektör Yardımcılığı

Sayı : 73613421-604.01.02-E.2005060052 **Tarih**: 06.05.2020
Konu : Etik Kurul Kararı 2020/2

Sn. Kubilay ATAŞ

YTÜ Etik Kurulu, Üniversitemiz yüksek lisans öğrencisi olarak hazırlamış olduğunuz **"Sürüş Güvenliği İçin Gömülü Sistem Tasarımı"** başlıklı projenizi etik açıdan incelemiştir. Sunulan dosya ve bu dosyaya göre yapılacak olan veri toplama araç ve yöntemlerine konu olan bilgiler hakkında etiğe aykırı herhangi bir bulguya rastlanmamıştır.

Bilgilerinize rica ederim.

e-imzalıdır
Prof. Dr. Yücel ŞAHİN
Rektör a.
Rektör Yardımcısı

Eki:Etik Kurul Kararı (1 Sayfa) (Elektronik Ek)

Adres : İstanbul
Tel / Fax : 0 (212) 383 20 58 / 02122585140
Kep Adresi : yildizteknikuniversitesi@hs01.kep.tr
İrtibat : Serpil Düzenli
Web : http://www.apry.yildiz.edu.tr/
e-Posta : sduzenli@yildiz.edu.tr

Bu belge 5070 sayılı elektronik imza kanununa göre güvenli elektronik imza ile imzalanmıştır.
http://www.ebys.yildiz.edu.tr/Dogrulama/Index?EvrakNo=E.2005060052&ErisimKodu=2566aa52

YILDIZ TEKNİK ÜNİVERSİTESİ
Akademik Etik Kurul

Toplantı Tarihi: 06.05.2020 **Toplantı No:2020/02**

AKADEMİK ETİK KURUL TOPLANTI KARARI

Yürütücülüğünü Üniversitemiz Elektrik-Elektronik Fakültesi öğretim üyelerinden Dr. Öğretim Üyesi Revna Acar Vural'ın danışmanlığında, yüksek lisans öğrencisi Kubilay Ataş'ın yapacağı **"Sürüş Güvenliği İçin Gömülü Sistem Tasarımı"** adlı çalışma ve bu çalışmada kullanılacak veri toplama araçları ve yöntemlerine ilişkin bilgilerde etiğe aykırı herhangi bir bulguya rastlanmamıştır.

Akademik Kurul Üyeleri

Prof. Dr. Abdürrezzak E. BOZDOĞAN
Başkan

Prof. Dr. Nihan ÇETİN DEMİREL
Başkan Yardımcısı

Prof. Dr. Kenan AYDIN
Üye

Prof. Dr. Adem BAKKALOĞLU
Üye

Prof. Dr. Erhan ALTAN
Üye

Yıldız Kampüsü 34349 Yıldız /İstanbul **Tel:** ((0212) 260 21 99 **Faks:** (0212) 258 51 40 , Tlx:26837 IYU
E-posta: apry@yildiz.edu.tr **Web:** http://www.yildiz.edu.tr
Bu belge 5070 sayılı elektronik imza kanununa göre güvenli elektronik imza ile imzalanmıştır.
http://www.ebys.yildiz.edu.tr/Dogrulama/Index?EvrakNo=E.2005060052&ErisimKodu=2566aa52

PUBLICATIONS FROM THE THESIS

Conference Papers

1. K. Ataş and R. A. Vural, "Detection of Driver Distraction using YOLOv5 Network," 2021 2nd Global Conference for Advancement in Technology (GCAT), 2021, pp. 1-6, doi: 10.1109/GCAT52182.2021.9587626.