FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

TRAFFIC SIGN RECOGNITION

Botan Hamza Hasan

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

DECEMBER 2020

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Master's Thesis

TRAFFIC SIGN RECOGNITION

Author
Botan Hamza Hasan

Supervisor
Assist Prof. Dr. Mehmet KAYA

DECEMBER 2020
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

TÜRKİYE

Department of Software Engineering

Master's Thesis

Title:	TRAFFIC SIGN RECOGNITION
Author:	Botan Hamza Hasan
Submission Date:	20 November 2020
Defense Date:	25 December 2020

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Signature

Supervisor:	Assist Prof. Dr. Mehmet KAYA Syracuse University, Faculty of Engineering and Computer	Approved
Chair:	Prof. Dr. Asaf VAROL Maltepe University, College of Engineering and Natural Science	Approved
Member:	Assoc. Dr. Murat KARABATAK Firat University, Faculty of Technology	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Assoc. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled " TRAFFIC SIGN RECOGNITION " in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

25 December 2020

Botan Hamza Hasan

PREFACE

This thesis discusses the traffic signs recognition by Android Studio and Java language with OpenCV library. This is the first project implemented in this way with clarifying the steps scientifically explained.

It has a great impact in our daily life which gives notification about our speed and the traffic signs during driving car and makes us less accidents and avoid drivers from over speeding and doesn't miss the traffic signs by the drivers and provide the drivers with pieces of information which help in driving safe and conveniently.

During implementing I faced difficulties of lack resources and combining the technologies such as combining OpenCV library and methods to be implemented in reality and gives us result! So, by efforts and supporting my supervisor Dr. Mehmet Kaya able to finalize and complete the thesis with practice part. In addition, all contain of the thesis is unique could be used as a patent to be a standard for other thesis and projects to be get benefit from it.

Botan Hamza Hasan
ELAZIG, 2020

TABLE OF CONTENTS

Preface iv
Abstract vii
Özet viii
List of Figures ix
Lıst of Tables xi
Abbreviations xii

1. INTRODUCTION 1
1.1. Aim of the Research Project 2
1.2. Problem Statement 2
1.3. Scope of the Research 3
1.4. Contributions 3
1.5. Thesis Layout 4

2. LITERATURE REVIEW & BACKGROUND THEORY 5
2.1. General object detection framework 6
2.2. Detection Methods: 7
2.2.1. Viola Jones 7
2.2.2. Scale-invariant feature transform (SIFT) 7
2.2.3. Histogram of oriented gradients (HOG) 8
2.2.4. RGB Color Space 8
2.2.5. HSV Color Space 9
2.3. Deep Learning approaches 9
2.3.1. CNN (Convoluted Neural Networks) 9
2.3.2. Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN) 10

3. MATERIAL AND METHODOLOGIES USED 12
3.1. Feature Extraction Methods 14
3.2. Region of Interest Method 14
3.3. Mat Matrix 16
3.3.1. Creating Mat Matrixes 16
3.4. SURF (Speeded-Up Robust Features) 18
3.5. Color Segmentation Methods 20
3.5.1 Color Information based detection 20
3.5.2 RGB Color Space 20
3.5.3 HSV Color Space 22
3.5.4 HSV Color Model usage 22
3.5.5 Value (or Brightness) 23
3.5.6 HSV model usage 23
3.6. OpenCV and Computer Vision 25
3.7. Fundamentals of Computer Vision 26
3.7.1. Camera Calibration 3D 26
3.7.2. Image Processing 28
3.7.3. Difference between Image Classification and Segmentation 29
3.7.4 Image Classification Working Principles 29
3.7.5 Image Segmentation in Image Processing Processes 30

3.3.6. Noise Filtering 32
3.3.7. Sources of Noise in Images 32
3.7.8. Different types of Noise 32
3.8 Filter Images and Videos 33
3.9 Smoothening/Blurring of Images 33
3.10 Image De-Noising 33
3.11 General Object Detection Framework 35
3.12 Object Tracking by Kalman Filter 35
3.12 CNN (Convolutional Neural Networks) and Machine Learning 35
3.13 Training Process of TSDR 39
3.14 . German Traffic Sign Dataset (GTSRB) Description 40
4 RESULTS AND DISCUSSION 41
4.1 Requirements for Installation on Windows Machine 41
4.1.1 Installation steps for Windows Machine 41
4.2 Adding Custom SDK Tool 42
4.3 Adding OpenCV Library 300 42
4.4 GRADLE 43
4.4.1. Necessity of Gradle 43
4.4.2 Top-level build.gradle 44
4.4.3 Module-level build.gradle 45
4.5 Android Manifest 46
4.5.1 The importance of Android manifest file 47
4.5.2 Elements of AndroidManifest.xml 47
4.5.3 The Elements of components of the Application 48
4.5.4 Structure of AndroidManifest.xml 49
4.6 Android Activity 49
4.7 Kalman Filter 51
4.8 Running the System 52
5. CONCLUSION AND FUTURE WORK 58
REFERENCES 59
Currıculum Vıtae 63

ABSTRACT

TRAFFIC SIGN RECOGNITION

Botan Hamza Hasan

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

December 2020, Page: ix + 62

Effective and accurate object detection has been a matter of great significance in the progression of computer vision systems. With passage of time, there have been many effective and accurate object detection has been a matter of great significance in the progression of computer vision systems. With passage of time, there have been many significant enhancements in deep learning techniques, and in turn it has led to the accuracy for object detection being risen drastically.

The aim of this thesis is to amalgamate state of the art technique for Road Sign Detection and Recognition with the goal of conquering greater accuracy with a real-time performance. An extra level of driver assistance is furnished by the Road Sign Detection and Recognition (RSDR) system which helps in enhanced safety for road users, passengers, passengers, vehicles. Its usage benefits the drivers with driving disabilities as it alerts them about the presence of road signs to decrease risks in critical situations of driving distractions, poor sight and weather conditions. There have been several attempts in the field of Road Sign Detection and Recognition have been proposed in literature Review; the design of a plush algorithm still an ongoing research prospective with room for improvement. The primary objective of the thesis is to solve some of the major challenges in RSDR. This is achieved keeping in mind the variations in colour illumination, scale, rotation, translation, occlusion, computational complexity and functional limitations.

The pipeline can be sub-categorized further into; Colour Segmentation, Shape Classification and Content Recognition. The methodologies used in this project are discussed in details in a chapter whereas the results and outcomes are talked about in another. In this project, we use a Convolutional Neural Network Approach with OpenCV to solve the problem of frame extraction, colour segmentation, signs detections and recognition in an end-to-end fashion.

The model was trained on lots of publicly available road sign images as well as it recorded video real time. The resulting system was then very fast and accurate.

Keywords: Traffic Sign Detection & Recognition, Convolutional Neural Network, Speed Up Robust Feature, Hue Saturation Value HSV, Colour Segmentation, OpenCV, Shape Classification.

ÖZET

Botan Hamza Hasan

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Aralık 2020, Sayfa: ix + 62

Etkili ve doğru nesne tespiti, bilgisayarlı görme sistemlerinin ilerlemesinde büyük öneme sahip bir konu olmuştur. Zaman geçtikçe, birçok etkili ve doğru nesne tespiti, bilgisayar görüntü sistemlerinin ilerlemesinde önemli bir konu haline geldi. Zaman geçtikçe, derin öğrenme tekniklerinde birçok önemli gelişme meydana gelmiş ve buda nesne algılama doğruluğunun büyük ölçüde artmasına neden olmuştur.

Bu araştırmanın amacı, gerçek zamanlı bir performans ile daha fazla doğruluk elde etmek amacı ile Yol İşareti Algılama ve Tanıma için en son teknoloji tekniğini birleştirmektir. Yol kullanıcıları, yolcular, yolcular ve araçlar için gelişmiş güvenlik sağlamaya yardımcı olan Yol İşareti Algılama ve Tanıma (RSDR) sistemi tarafından ekstra bir sürücü desteği seviyesi sağlanır. Kullanım faydları, engelli sürücülere sürüş dikkat dağınıklığı, kötü görüş ve hava koşulları gibi kritik durumlarda riskleri azaltmak ve yol işaretlerinin varlığı konusunda yardımcı olmasıdır.

Yol İşareti Algılama ve Tanıma alanında birkaç girişimde bulunulmuştur literatür taramasında önerilmiştir; Bir peluş algoritmanın tasarımı, iyileştirme alanı olan devam eden bir araştırma beklentisidir. Tezin temel amacı, RSDR'deki bazı önemli zorlukları çözmektir. Bu, renk aydınlatma, ölçek, döndürme, çevirme, kapatma, hesaplama karmaşıklığı ve işlevsel sınırlamalardaki farklılıklar akılda tutularak elde edilir. Boru hattı aşağıdaki alt kategorilere ayrılabilir; Renk Bölümleme, Şekil Sınıflandırma ve İçerik Tanıma. Bu projede kullanılan metodolojiler bir bölümde ayrıntılı olarak tartışılırken, sonuçlar başka bir bölümde anlatılmıştır. Bu projede, çerçeve çıkarma, renk bölümleme, işaret algılama ve tanıma problemlerini uçtan uca bir şekilde çözmek için OpenCV ile Evrişimli Sinir Ağı Yaklaşımı kullanılmıştır. Model, halka açık bir çok yol işareti görüntüsü üzerinde eğitildi ve gerçek zamanlı video kaydedildi. Ortaya çıkan sistem daha sonra çok hızlı ve doğruydu.

Anahtar Kelimeler: Trafik İşareti Algılama ve Tanıma, Evrişimli Sinir Ağı, Sağlam Özelliği Hızlandırma, Renk Doygunluğu Değeri HSV, Renk Segmentasyonu, OpenCV, Şekil Sınıflandırma.

LIST OF FIGURES

Page

Figure 2.1 - The detection algorithm bounding boxes. 6

Figure 3.1 - Project Architecture Design 12

Figure 3.2 – System Process Flowchart 13

Figure 3.3 - Region of interest detection 15

Figure 3.4 - RBG Dimensional coordinates axis 21

Figure 3.5 - Additive color mixing 21

Figure 3.6 - HSV model components (Hue, Saturation and Value) 24

Figure 3.7 - Solving Distortion 27

Figure 3.8 - Removing Distortion 28

Figure 3.9 - Traffic Sign Classification Process 30

Figure 3.10 - Traffic Sign Segmentation 31

Figure 3.11 - Applying Image DE-Noising 34

Figure 3.12 - CNN Classifier Procedure 36

Figure 3.13 - CNN Classifier Layers 37

Figure 3.14 - Feature Extraction and Classifier in CNN 37

Figure 3.15 - Traffic Sign Recognition Process 39

Figure 3.16 - German Traffic Sign Recognition Benchmark (GTSRB) Dataset..... 40

Figure 4.1 - Project Architecture 42

Figure 4.2 - OpenCV files 43

Figure 4.3 - Build.gradle 44

Figure 4.4 - App Files & Architecture 45

Figure 4.5 - Launching the Application 46

Figure 4.6 - Android Manifest.xml file 47

Figure 4.7 - Android Activity Framework 50

Figure 4.8 - 4.10 Kalman Filter File 52

Figure 4.9 - The system Application Icon 52

Figure 4.10 - When recognize a traffic sign gives us notification alarm of recognizing 53

Figure 4.11 - Showing real time traffic sign recognition during driving 53

Figure 4.12 - Day & Night Modes for detecting traffic signs 54

Figure 4.13 - HSV Settings 55

Figure 4.14 - Traffic sign recognition during Daytime 55

Figure 4.15 - Traffic sign recognition during nighttime 56
Figure 4.16 - Recognizing types of traffic signs 56
Figure 4.17 - Giving alarm when driving over speed 57

LIST OF TABLES

Page

Table 2.1 - Represents a comparison among the mentioned studies. 10

ABBREVIATIONS

Abbreviations

API	: Application Programming Interface
CMYK	: Cyan, Magenta, Yellow, Key
CNN	: Convolutional Neural Networks
CUDA	: Compute Unified Device Architecture
GTSRB	: German Traffic Sign Recognition Benchmark
HOG	: Histogram of oriented gradients
HSV	: Hue, Saturation and Value
INRIA	: French National Institute for Research in Computer Science and Automation
OpenCL	: Open Computing Language
OpenCV	: Open Computing Visual
PRN	: Region Proposal Network
R-CNN	: Region Convolutional Neural Networks
RGB	: Red, Green and Blue
RoI	: Region of Interest
RSDR	: Road Sign Detection and Recognition
SIFT	: Scale-invariant feature transform
SURG	: Speeded-Up Robust Features
TSDR	: Traffic Sign Detection and Recognition

1. INTRODUCTION

The universe is governed by a combination of several laws which are environmental, physical and many more. Likewise, mankind has created a set of traffics rules, to guide the people travelling and to regulate the traffic flow. Road and traffic signs are basically symbols/visual stimulus which give information about the roads ahead that can be understood by drivers [1]. They provide the drivers with pieces of information which help in driving safe and conveniently. Such information is encoded in an accepted format in which the traffic signs possess certain visual traits: Shape, color and pictogram, failing to notice/interpret or not complying with these traffic signs may directly lead or indirectly cause a traffic accident. A driver has to do multiple tasks while driving including things apart from paying attention to the road such as looking at the vehicles around, constantly monitoring blind spots, indicating the lane changes and abiding by traffic rules leading to him/her getting overburdened [2]. Moreover, in adverse conditions like bad weather (foggy, rainy, harsh winter), the driver also may accidentally not notice the road signs. It can just be imagined how difficult it would be for drivers with disabilities in these conditions. However, if in such scenario, there exists an application/method which enables automatic detection and recognition of road signs, it could counterbalance for a driver's possible inattention. Such a method would decrease a driver's stress by helping him follow the road traffic signs and hence make driving safer, smoother and more convenient. This thesis's main proposition is based on the development of a RSDR (real-time road sign detection and recognition) system [3]. It is based upon four main steps cited below:

- Acquiring image: A digital camera was used and placed in front of the car used to collect and acquire data which was in the form of a live video stream which consisted of several frames.
- Pre-processing Image & Traffic Sign Detection:
  This part comprises of the following steps: -
    1. Frames: Frames were taken one at a time and converted from the Red, Green and Blue (RGB), into the HSV model i.e., Hue, Saturation and Value for segregation.
    2. Noise filter: It was applied to the frames to remove noise from the scene.
    3. Connected Components Labelling was applied to the objects left in the frame.
    4. Detection and machine learning in order to remove objects which are not of appropriate size, the Size filter was applied.
- **Traffic Sign Tracking**: Blob tracking have been used to predict the search region for the sign in the next frame to improve the traffic sign detection time.
- **Traffic Sign Recognition Stage (feature extraction**): A new traffic sign recognition technique was applied named as "Pattern Matching", which was found to be much faster, and efficient according to the real time requirements.

1.1. Aim of the Research Project

Since the research has been done in the real time environment for traffic sign recognition, there are three most important aspects considered, to be achieved by the real-time traffic sign recognition system:

- **Speed:** To enhance the processing time in traffic sign recognition.
- **Efficiency:** To check the overall system's response time and take measures accordingly to improve the efficiency.
- **Reliability:** To make this system more reliable, the project is to be designed in such a way that the predictions are to be very accurate.

The basic area of this project is the shape-based detection and recognition of the traffic signs in Real-Time, while the pictogram information has been used to aid in the correct shape detection.

1.2. Problem Statement

Robust and well-arranged system is important to cover and find out the attention of drivers during driving over the streets and roads safety, the Traffic Sign Detection and Recognition (TSDR) system has been initiated, especially the traffic signs are not in fine corner or it is faced damaged by external factors. After opening the camera of the mobile in real-time, the video sequences under different conditions (lighting, weather and climate, velocity and quickness, etc.), below factors are predictable:

- Determine a method to extract the traffic signs through driving car in real time by video sequencing the frames to be segmented.
- Coming up with a method to clean the extracted the traffic sign from noise and distortion.
- Coming up with a method to find out the region of interest and identify the traffic sign with alarming notification.
- Coming up with a method to find out the speed meter with alarming notification when the car drives over speed.

1.3. Scope of the Research

This approach is with OpenCV to solve the problem of frame extraction, colour segmentation, signs detections and recognition in an end-to-end fashion and speed meter as well, to excute this system all the operations for detection and recognition of traffic signs CNN with the library of OpenCV is used, as Open CV comes with a lot of pre-defined functions. to get frame extraction and fetch the velocity of the vehicle, Region of Interest and Robust Feature methods have been used in this application.

Additionally, to achieve better accuracy for color segmentation, in daytime setting the RGB model is used whereas HSV (Hue, Saturation and Value) model is utilized in night time conditions. A very advanced library developed by Intel, OpenCV, is very robust artificially intelligent library. Its capability in performance includes image processing tasks, facial recognition, advanced video surveillance.

Kalman filter is one of the algorithms used for traking objects such as traffic signs. The model is trained on lots of publicly available road sign images as well as it recorded video real time. The resulting system is very fast and accurate.

1.4. Contributions

The major contribution of this work is to design and implement a new approach in the field of detection and recognition traffic signs from the existing methods and algorithms which is not done already in a scientific way to explain all the steps. The new approach can reduce the car accidents and aware the drivers from all traffic signs which passes through driving by getting notification alarm from the system with speed meter when drives over speed by exceeding 120 KMPH. In building this approach accurate of detecting and recognizing the traffic signs are approved and implementing in realistic by using Android studio tools and java language is one of the characteristic of this system.

This work comes in this narrative approach, and the content of this approach can provide high accurate (which means accuracy improvement which is not implemented by Android Studio such as a mobile application) in traffic signs detection and recognition.

1.5. Thesis Layout

The Outline of the Thesis There are 6 chapters in the thesis. Apart from the current chapter, there are another five chapters which deal with the theoretical and pragmatic aspects of the contents of the thesis. Each chapter is self-contained and independent in terms of topics discussed. Inspite of that, there are certain dependencies amongst the different chapters which are mentioned as follows:

Chapter 1: This chapter describes the background as well as the aims and objectives of the thesis.

Chapter 2: Here the previous approaches to general visual object detection and recognition are reviewed. Evolved traffic sign recognition systems are separately discussed. There limitations are pointed out and the space for improvement is identified.

Chapter 3: This chapter presents out the methodologies involved in the whole project. Firstly, Region of Interest method and Speed Up Robust Feature (SURF) technique is introduced for frame extraction. In addition to frame extraction, it also helps determine the speed of the vehicle. The color segmentation is carried out by two methods; RGB (Red, Green, Blue) model for daylight and HSV (Hue, Saturation, Value) model for nighttime. Finally, all the aspects such as image pre-processing, detection, tracking and recognition are discussed in detail which are done via OpenCV library. The international traffic database is used over here to train the model.

Chapter 4: It is focused on the results and experimentation of the thesis. This includes deliberation of real-time tests carried out for the application as well as their results.

Chapter 5: This is the concluding chapter which recapitulates the work presented in this thesis and provides a general discussion of its contributions and possible further research applications.

2. LITERATURE REVIEW & BACKGROUND THEORY

We, as human beings, face the issue of recognition in everyday lives. In the computer vision and pattern recognition communities, object detection and recognition has been discussed a lot over the past few years. When we speak of problems that we can solve with a computer, one of the most pertinent issues that we come across is image analytics and recognizing the objects [3]. It is indeed very difficult to find a universal solution to perceivable issues such as face or object recognition. Because what may work for stationary objects, would not work for moving objects. The problem of image recognition can be divided into further smaller bits. For example, if the object we are looking for is known to us, then it's an object detection sub problem [4].

This involves taking the input images and scanning them to determine the locations where the target object will be present. Another important aspect of image recognition is of category recognition. Simply put, the objective is to detect and determine which category the objects belong to such as Furniture, Human, vehicle etc. So, in this phase the focus is on object detection, the entity is known to us in the image that we are looking for and we need to recognize the target objects as fast as possible with maximum efficiency [5].

To this day, in spite of all modern facilities that are present in the cameras and other detection devices humans are better at visual recognition in comparison with computers. The human visual system information including the eye which is behind generating and interpreting sensory data. It consists of an advanced structure and the activity of natural biological optical sensors [6].

Let's say, we possess an object class of interest and a reference illustrated picture for scrutiny, with the aim to produce a system which can recognize and ascertain instances of the object in hand in the given picture to then produce an output with their locations. If the object to be identified in our case is a car, the outcome off the detector should produce something like illustrated can be in the Figure 2.1. The detection algorithm bounding boxes, some typical issues enlisted with object detection are identified. The biggest and crucial problem with the creation of a strong object detector is that the degree of fluctuation in images and videos [7].

Figure 2.1 - The detection algorithm bounding boxes.

The detection algorithm should detect the traffic signs and be identified in bounding boxes discussed earlier. One of the most essential things that lead to fluctuation include:

- Position of object in hand
- Separation /position with respect to the camera
- Large variance inside class in object classes
- Derangement in background
- Contrast in color
- Difference in brightness

The objective of detection algorithms is to recognize and identify objects in all circumstances and be a working model under all conditions [7].

2.1. General object detection framework

To summarize we can say that mainly there are three steps in an object detection and identification structure/model [8]:

- For producing regions of interest or region proposals, a particular algorithm or model is utilized. These region proposals can be defined as basically a set of bounding boxes covering the full image that is, an object localization factor.

- Post application of algorithm to get region proposals, visual characteristics are obtained for each of the bounding boxes. These are assessed, and it is if objects are present in the proposals based on visual features. That is basically identifying an object classification component.
- Finally, all overlapping boxes as shown in Figure 2.1. are collated into a solitary bounding box also called as non-maximum suppression.

Over the course of time, many drastic improvements have been added to detection quality. A major factor contributing to the detection quality is the development of feature representations with the design of better representations which can be extracted efficiently, it turns out to be very beneficial than modifying machine learning algorithms that are used for disposition.

2.2. Detection Methods:

2.2.1. Viola Jones

Viola-Jones was one of the first detection framework to be developed which provided real time competitive detection rates. It was developed by Paul Viola and Michael Jones in the year 2001. In the context of object detection, it was primarily motivated to focus on the issue of face detection; however, apart from faces it could also be used to detect and perceive a range of object classes [9].

The primary problem to be fixed was the detection of pictures, humans can do this task easily however a computer needs to be equipped with precise instructions and constraints. So, this model of framework required full view frontal upright faces. Hence, the entire face had to face the towards the camera and shouldn't have tilts to any side. So, even though the restrictions don't right away seem to diminish the utility of the algorithm, as a recognition step usually follows the detection step, in reality the limits on pose are quite acceptable [10].

2.2.2. Scale-invariant feature transform (SIFT)

Scale-invariant feature transform (SIFT) was a feature detection algorithm using the technology of Computer Vision, which will be discussed separately, to distinguish, identify and describe the local features in images. It was patented by David Lowe in University of British Columbia in 1999 [11]. Several of its application included object recognition, robotic mapping and exploration, 3D modelling among other things. Using a bunch of reference pictures, SIFT key points of objects are first derived and then saved in database. The method of an object being detected in a new image is by pitting it and analyzing every feature of it with the ones in database and then finding candidates based on their Euclidian distance of vectors. After taking the set of

equivalent twins, subset of key points which match and possess similarity on the object and its location, scale, and sense of direction in the new image are marked to figure out good matches. A productive and useful hash table along with a Hough Transform is used in combination to figure out consistent clusters. From these, each of 3 or more features which have similarity with object and its orientation are then made to undergo another comprehensive model validation. The ones that conform from this step are then removed [11]. In the end, the probability that a set of features indicates the presence of an object is calculated, taking into account the accuracy of the fit and number of false probable matches. Hence, the object that passes all the steps can be accurately be declared as correct with high precision.

2.2.3. Histogram of oriented gradients (HOG)

This method first arrived on the scene in 1986, when Robert K. McConnell from Wayland Research Inc came up with the concepts behind HOG. It is a feature descriptor, finding its place in Computer Vision and Image processing for Object Detection. The occurrences of gradient orientation present in the localized parts of the image are primarily used in this technique [12].

One can see parallels in this technique to the one that can be seen in Edge Oriented Histograms, scale invariant transform descriptors and shape contexts as the similarities are clear. However, on the difference side, HOG is calculated on a compact grid of evenly spaced cells. In order to improve on accuracy, they utilize overlaying local contrast normalization [13].

These concepts and the method gained mainstream usage popularity in 2005 when Navneet Dalal and Bill Triggs, researchers for the French National Institute for Research in Computer Science and Automation (INRIA), presented findings on Histogram of Oriented Gradients. In their work, the primary focus was on still images. In addition to this work, they extended their test range to include human detection in videos and range of common animals and vehicles, in still images [14].

2.2.4. RGB Color Space

There is color susceptible receptor for red, green and blue in the human eye. The perception that every visible color can be decomposed into combinations of these three “primary colors” is only possible hypothetically and theoretically. A color display screen can display millions of colors by combining various severities of red, green and blue. Therefore, it would be trivial to value the range of depth for each color, in a range of 0 to 255 (size of one byte). The term “color depth” is otherwise known as ‘range of intensity’ [15].

To represent the possibilities of combining three primary colors, a 3-dimensional coordinate plane can be used. It can be done by representing R(Red), G(Green) and B(Blue) with numbers on each axis. This coordinate plane obtained in the shape of a cube is called the RGB color space.

2.2.5. HSV Color Space

The most convenient way to combine and generate colors is the RGB model discussed above. The color combination CMYK (Cyan, Magenta, Yellow, Key) is widely used in commercial printers. It's also observed that HSV (hue, saturation, value) and is so popular and convenient that it is commonly used as the color selector for graphics software. Therefore, the spectrum perceived by human eyes which explains the process of color combination is done by these methods [16].

In comparison to RGB and CMYK which are mostly used as primary colors, humans' perception of colors is closer to HSV. It consists of these:

- Hue
- Saturation
- value

The scope of this color explains hue or tint in terms of their shade (saturation or amount of grey) and their illumination value. Some software like Adobe Photoshop, use the abbreviation HSB, cause the color combination is Hue, Saturation, Brightness (where value is substituted by brightness). But this shouldn't be confused as in essence, HSV and HSB are the same models.

2.3.Deep Learning approaches

2.3.1. CNN (Convoluted Neural Networks)

CNN or Convoluted Neural networks is one of the most known algorithms in the field of image classification and consists of Convolution layers, Activation function layers, max pooling of layers to decrease the dimensionality by not letting go of a lot of features. A feature map is generated by the ultimate layer of CNN [17]. If we input a Car or a Forklift image, the algorithm m will come back by telling whether it's a car or a forklift. Its use is not just restricted to that; it is laced up with some remarkable computational abilities which produce great advancements.

2.3.2. Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN)

In the field of Object detection, using R-CNN, RPN has proven to be the most consistent and an efficient method till date. Its aimed to propose multiple identifiable objects within a given image. It was developed by Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun in a very popular paper, and ever since it has held the imagination of Data Scientists and AI engineers. Its application is varied from helping people with disabilities, to detection of objects on the road in a self-driven car [18]. Table 2.1. deals with the collection of some represents a comparison among the mentioned studies.

Table 2.1 - Represents a comparison among the mentioned studies.

Author	Used Algorithm(s)	Usage(methodology)	Conclusion
Promlainak, S., Woraratpanya, K., Kuengwong, J., & Kuroki, Y. [10].	convolution neural network CNN used	TensorFlow library with multithreaded programming Compute Unified Device Architecture (CUDA)	It is used as classifier not traffic sign detection and it is computer API not Mobile application.
Romdhane, N. B., Mliki, H., & Hammami, Romdhane, N. B., Mliki, H., & Hammami [19]	classified using HOG features and a linear SVM classifier	A color-based segmentation method is applied to generate traffic sign candidate regions	It is theoretical and supposed the mentioned algorithms and methods to be used in order to get high accurate rate in detection.
Credi, J. [20]	Convolutional Neural Networks (CNNs, and ConvNets),	Open Computing Language (OpenCL) framework with C# with German Traffic Sign Recognition Benchmark (GTSRB)	It is a desktop application used just for recognizing German traffic data sets by input data to the application.
Sharma, A. [21]	Richardson-Lucy algorithm and Gaussian blur	Using Matlab and TensorFlow library with Python	It is a desktop applicant lo classify and recognize traffic signs

Bilgin, E., & Robila, S. [22]	A Raspberry Pi board and K-Nearest algorithms are used.	OpenCV libraries, color-based segmentation methods with IDE by Android studio with Python language are used	It is a hardware project to recognize the road signs on real time.
Htet Wai Kyu, Lu Mau, Hla Mio Tun [23]	Hough Circle Transform and SRUF algorithm are used	Grey-Scale Conversion and Sobel edge detection methods are used.	A specific of traffic sign for limits of dataset are used to detect the limit of speed signs then compare with its dataset and gives the result. Also, it is a desktop application.
Villalonga, G., Van de Weijer, J., & López, A. M. [24]	Taylor et al and Convolutional Neural Networks (CNN) are used	synthetic dataset and ResNet-based classifier methods are used	It is a theoretical research and used to recognize traffic signs which they predefined by the dataset.

3. MATERIAL AND METHODOLOGIES USED

To perform all the operations for detection and recognition of traffic signs, CNN with the library of OpenCV is used, as Open CV comes with a lot of pre-defined functions. In order to get frame extraction and fetch the velocity of the vehicle, Region of Interest and Robust Feature methods have been used in this application. Additionally, to achieve better accuracy for color segmentation, in daytime setting the RGB model is used whereas HSV (Hue, Saturation and Value) model is utilized in nighttime conditions. The flow chart is can be seen in Figure 3. A very advanced library developed by Intel, OpenCV, is very robust artificially intelligent library. Its capability in performance includes image processing tasks, facial recognition, advanced video surveillance, analysis of emotions, sentiment analysis, human computer interaction among many applications [22]. Hence it comes as no surprise that to get most of critical tasks done, tasks including frame extraction, color segmentation, object detection, tracking and recognition, OpenCV is used. It's used because all of the functionalities of most traditional methods mentioned before are imbibed and utilized in Computer Vision's library. This allows and empowers scientists/engineers to execute image processing/segmentation/classification activities with utmost precision and accuracy and enhance its output than would have been achieved if traditional methods were being used. The architecture diagram of the project and it is flowchart is mentioned below for better understanding [23]. Figure 3.1. Project Architecture Design. It is from the executed image detection process. In addition, Figure 3.2 System Process Flowchart given to understand the detection process easily.

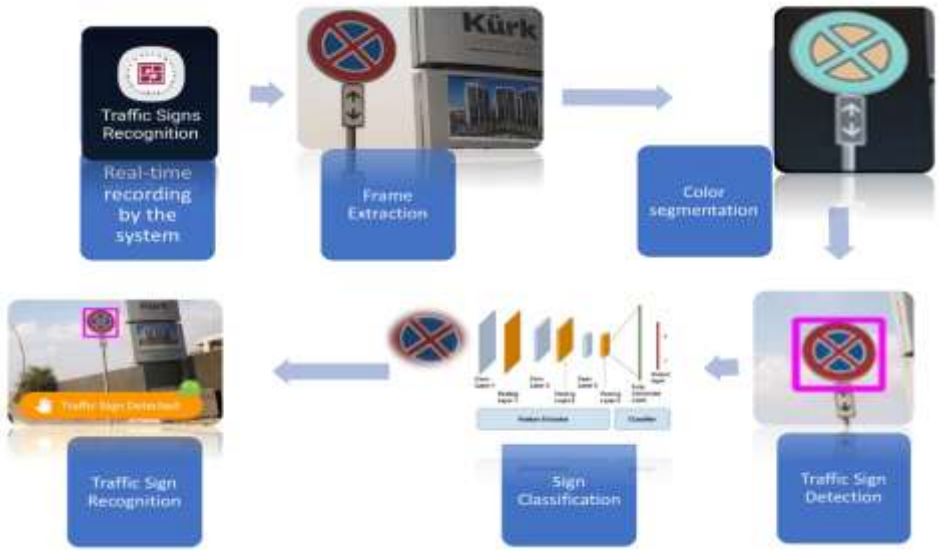

Figure 3.1 - Project Architecture Design

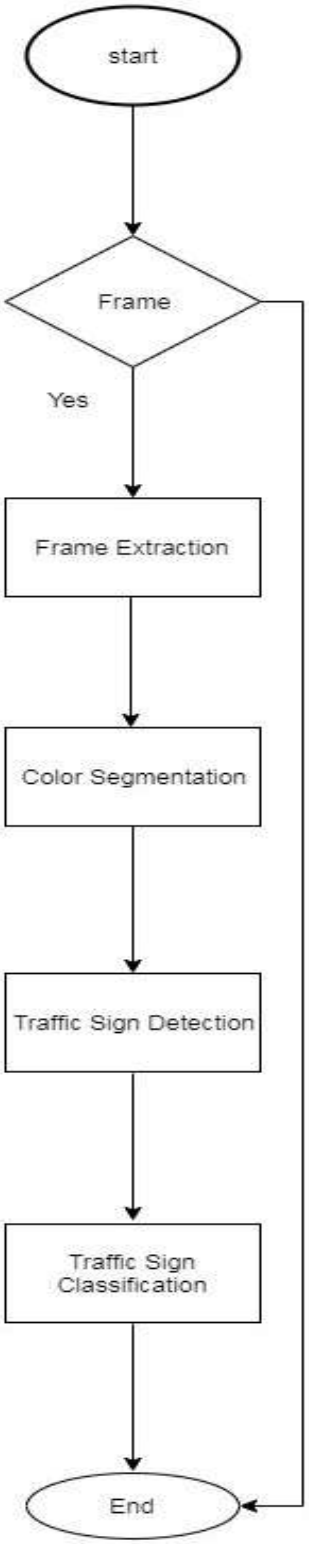

Figure 3.2 – System Process Flowchart

3.1. Feature Extraction Methods

Image Processing has become vital in many spheres of life for instance medicine, disease diagnosis, tumor/cancer detection, satellite imagery analysis etc. But analyzing the whole image captured to find answers for a particular pain area becomes cumbersome, hence sorting out right features play a very vital role in designing solutions for the problem in hand. It is also because demeanor of the images is defined by features only which showcase the time consumed, memory occupied and classification efficiency. Thus, it can only be achieved with the application of proper feature extraction techniques to smoothen image classification, detection and other following processes as well [24].

3.2. Region of Interest Method

To learn and understand more about object detections methods, we need to take and capture images, digital in nature. A lot of methods and equipment are used of which Digital cameras, computed tomography, scanners, and magnetic resonance imaging, which get widely used. What we (humans) observe are images in every case. However, while we transform the images that are observed to the equipment mentioned earlier, we come across are numerical measurable values for every point of captured picture, for more clarifying see Figure 3.3. Region of interest detection.

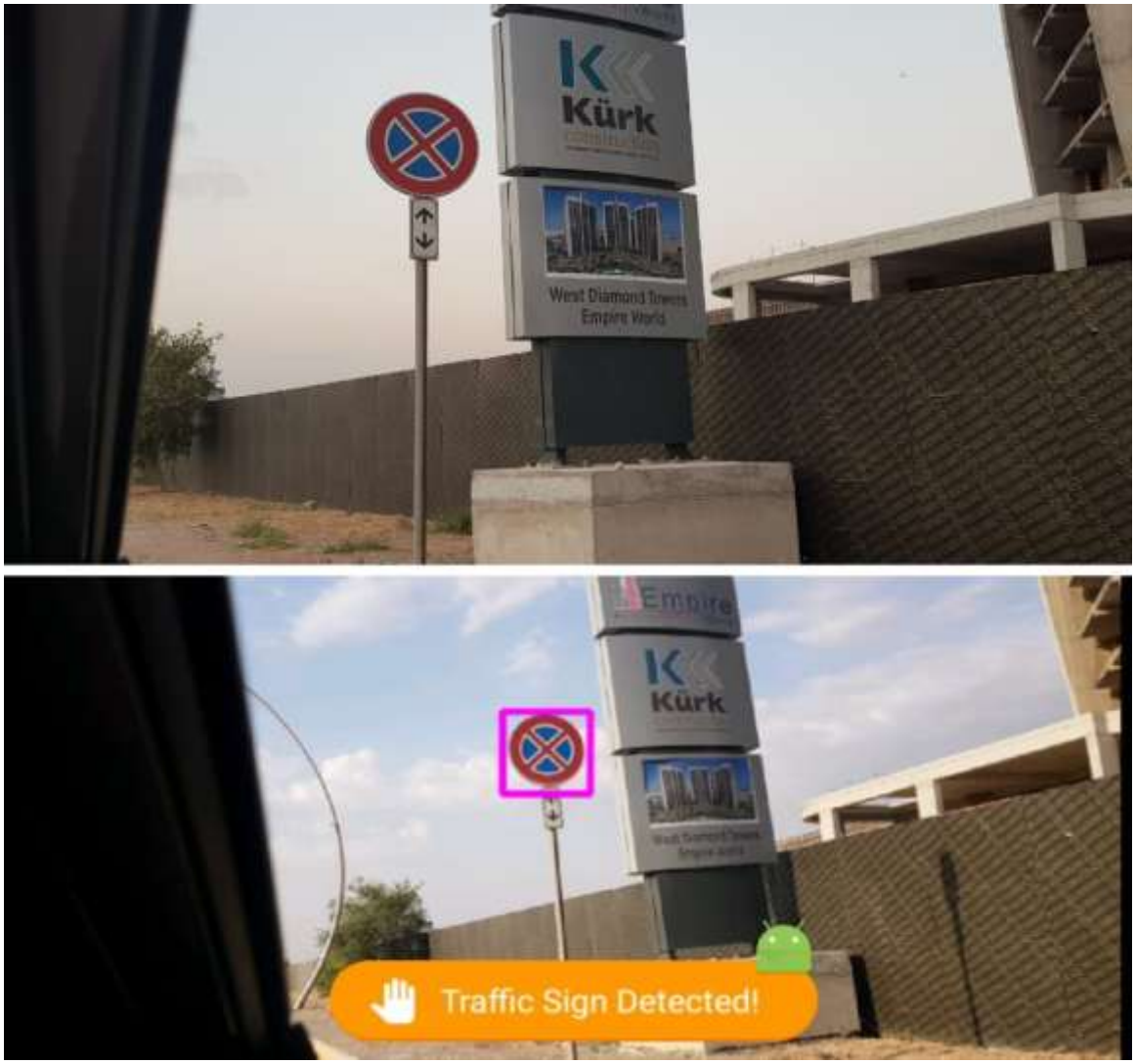

Figure 3.3 - Region of interest detection

While we see an image of the mirror of the vehicle, the input for the devices and equipment are quite different. While being converted into its digital form is actually a matrix containing all the concentration number of the pixel points. The method and process of obtaining and storing the pixels values may differ depending on what is needed. Overall, any input that comes in the form of a picture can be broken down into a set of numerical values and other data that can give description of the matrix [25]. As discussed earlier, OpenCV is a revolutionary robust computer vision library developed primarily to convert and modify this data. So, one of the first things to be taken care of in the process of image processing is to understand how OpenCV accumulates and examines pictorial information.

3.3. Mat Matrix

The OpenCV library has been around in use since 2001. In its earlier day, the library was built around a C interface. To store the image in the memory a C structure called Ipl image was used and utilized. One of the advantages is that it eliminates all the disadvantages of C language but one of the major issues is the necessity of manual management of memory. The basis of this is the assumption that it is the user's responsibility to allocate and free memory. It may be understood this may not be an obstacle for smaller programs, however once the code base grows, one will have to deal with all of memory management issues and not focus on solving the development work. Fortunately for all developers around, advancement in technology and coding resulted in development of C ++ and with it came the concept of classes [26].

This development now allowed the user automatically (more or less) manage memory and not do it by oneself manually. One of the major advantages of C ++ is that it is completely adaptable and congruent with C. So, a major issue of compatibility is eliminated and solved when there are changes being made in the code. Therefore, to ease the overall process, a new C ++ interface was introduced in OpenCV 2.0, which provided a new way of performing, meaning that one didn't have to worry about memory management, which helps in cr3eatinf more accurate and concise codes code. In short, one had to type less to get more output. In spite of all the advantages and efficiency it offered, many IDE (Integrated Development Environment) systems could only worked with C and didn't support C++. Over the course time, OpenCV came up with a Java interface, which was even more efficient than C / C ++. This was so, because Java is an object-oriented language [27]. So, it is faster and platform independent, whereas C ++ is platform based and C is a procedural language and hence came with some of their own lacunas in terms of operations.

An important part of image processing is fetching frames from live video / broadcast, which can be achieved using this "mat" method integrated into Computer Vision. So, it was no longer needed by Mat to allocate its memory manually and it could be released as soon as the memory was not actively necessary anymore. While it can be achieved anyway, a high percentage of OpenCV functions' output data gets computed automatically. And on top of this important feature as a reward, if a currently existing Mat object is passed, it can be reused as the matrix already contains the necessary memory space. In short, only that amount of memory that is critical/necessary for the task to be performed is used [26].

3.3.1. Creating Mat Matrixes

Mat can be described in crude terms as a class which consists of two data parts. The first one is the matrix header containing data like size of the matrix, storage method, address of matrix and

so on and secondly a pointer as reference to the matrix which includes pixel values. This could take any dimensionality and it depends on the method utilized for storage. The matrix header doesn't change, but the size of the matrix is dynamic and can change depending on the image. It generally is greater by orders of magnitude [26].

As discussed in the earlier paragraphs, OpenCV is a very advanced library used in image processing comprising of a massive compilation of image processing functions. In order to solve a computational problem, a lot of functions of the library will be used for a significant amount of time. As a result, the passage of images to functions is pretty common. Image processing algorithms can be computationally heavy. So, it would be absolutely undesirable and the last thing on anybody's mind to further decrease the speed of the program and reduce its efficiency by making copies of big sized images which are not necessary and obsolete.

Hence, to fix the issue a reference counting system is utilized by OpenCV as each *Mat* object possess its own header and not inside a separate object however thee same matrix could be common to two *Mat* objects. It can be done when the matrix pointers of the objects refer to the same address. Additionally, copy operators will replicate the headers and the pointer to the large matrix but not the information contained in them.

Mat A, C; // creates just the header parts

A = imread(argv[1], IMREAD_COLOR); // tells about the method used (matrix allocation)

Mat B(A); // copy constructor used

C = A; // Assignment operator

The objects mentioned eventually point to the same data matrix and eliciting changes using one of them can effect a change on the other ones as well. In practice, same data can be referred to and be accessed while using OpenCV. However, the parts contained in their header are unique. One of the most interesting aspects is that headers can be created which don't refer to the entire data but rather a subsection of it. For example, when a region of interest (*ROI*) is created in an image, a new header has also to be generated with new limitations or restricting boundaries:

Mat D (A, Rect(10, 10, 100, 100)); // using a rectangle

Mat E = A(Range::all(), Range(1,3)); // using row and column boundaries

So the question here is – if the matrix can belong to multiple Mat objects, then how is it ensured that the matrix assumes responsibility for cleaning it up when it's not necessary/relevant anymore. Concise answer would be the object that among other objects which used it last for the matrix. This is managed and dealt with by the service of a reference counting mechanism. Whenever a *Mat* object's header is copied, the matrix's counter is incremented and on the other hand If it is cleaned, the counter decreases. If it reaches and records a null or zero, the matrix

memory is freed and reset. At certain times, duplicating the entire matrix itself seems like a viable option, and to facilitate that OpenCV comes preloaded with '***cv::Mat::clone() and cv::Mat::copyTo() ' functions.***

Mat F = B.clone();

Mat G;

B.copyTo(G);

Once we have reached this level, implementing changes on *F* or *G* will not cause any changes in the matrix pointed by the *B*'s header. From all the information presented above, it is critical to point out that:

- The output allocation of image happens by itself (automatically) unless it is specified otherwise
- There is no need of allocating memory manually, since with OpenCV's C++ or Java interface it is taken care of automatically.
- The assignment operator replicates the header. Same is followed by the copy constructor.
- OpenCV comes with important pre-loaded functions such as, to copy contents of underlying matrix of an image by using the functions:

cv::Mat::clone() and cv::Mat::copyTo()

3.4. SURF (Speeded-Up Robust Features)

The development of SURF (Speeded-Up Robust Features) back in 2006, was a result of speeding up the process of key point detection and description. Back in the day, SIFT or the Scale-Invariant Feature Transform was earlier used for key point detection but because of its slowness SURF was created. In SIFT, the Laplacian of Gaussian was approximated with Difference of Gaussian for figuring scale-space. But SURF being slightly advanced, digs in a little further and approximates LoG with the help of a Box Filter [28]. A demonstration of such an approximation is orchestrated in the image below. A big bonus and a plus point of this is that, convolution with box filter can be derived easily. It can be done so with the help of integral images as well as parallel for scales varied in nature. SURF relies on determinant of Hessian matrix for both calibration and position.

With regards to assigning of orientation, wavelet responses in horizontal and vertical direction for a neighborhood of size 6s are used by SURF with application of sufficient gaussian weights. They are then mapped out and drawn in a space as mentioned afterwards in the image. Estimation of the prevalent orientation is achieved by calculating the sum of all responses inside a sliding orientation window angled at 60 degrees. One of the interesting aspects is that, wavelet response can be calculated by putting to use integral images without any hassle at any scale.

Rotation invariance is not necessary for a lot of applications. Hence there is no necessity of finding this orientation in turn speeding up the overall procedure [28]. One can find such functionality called Upright-SURF or U-SURF. Depending upon the flag, OpenCV is compatible and it supports both. It drastically enhances the pace and is robust up to -15 degrees to +15 degrees. If the degree value shows 0, orientation is found out. If it is 1, orientation is not necessary to be calculated and the process is sped up.

To describe feature, SURF uses Wavelet responses in horizontal and vertical direction. As figured earlier, the using integral images is very convenient. A neighborhood of size 20sX20s is taken around the key point, where 's' represents the size. The neighborhood frame is further broken down into 4x4 subregions. After this, horizontal and vertical wavelet responses are recorded for each of the sub regions and an equation like this is generated:

$$v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|) \quad \textbf{(3.1)}$$

When this is represented in vector form, SURF feature descriptor with a total of 64 dimensions is generated. The speed of computation and matching is inversely proportionally to the dimension value but furnish better diversity in characteristics [29].

For better distinctiveness, SURF feature descriptor has an extended 128-dimension version. The additive resultant of d_x and $|d_x|$ are calculated separately for the condition $d_y < 0$ and $d_y \geq 0$. Likewise, the sums of d_y and $|d_y|$ are taken apart depending to the sign of d_x, resulting in doubling the number of features which doesn't further much to computation complexity. OpenCV comes with a platform that supports both the 64 dimensions and 128 dimensions version by setting the value of flag extended with 0 and 1 for 64-dimensions and 128-dimensions respectively. The default is set at 128 dimensions.

One of the major enhancements in this case is the use of sign of Laplacian (trace of Hessian Matrix) for underlying interest point. This does not burden and anything to the computation cost as we know it is calculated during detection phase. Sign of the Laplacian separates and tells us the illuminated blobs on dark backdrop or environment from the opposite condition While doing comparison and finding matches the characteristics are taken to consideration which have same type of contrast as demonstrated in the image below. Even though this information while comparison is pretty basic in nature, it helps in faster and efficient matching, without affecting the output of descriptor [28].

Basically, the SURF improves a lot of features at all stages to fasten up the process. While performance is similar to SIFT, analysis reveals that it is thrice as swift in generating output than SIFT, at the same time. SURF fares better at dealing with images which are not straightforward and come with some degree of blurring and rotation, but it doesn't maintain the same level of performance when it comes to handling change in orientation and brightness.

3.5. Color Segmentation Methods

A low-level feature map of the input image is selected, from which image is computed. The actualization of the proposed interest operator presented in this thesis includes obtaining of the areas with high cluster of the color-specific gradients.

3.5.1 Color Information based detection

Hybrid Methods which include color-based and shape-based methods have their own set of pros and cons. As a result of which, Modern systems in current era are incorporating the use of RGB or HSV models. This is done in order to achieve greater efficiency of the TSDR system. In these hybrid methods, when we consider shape of an object, in color-based approaches it is taken into account after having looked at color, or it is used in shape detection as the main method but integrate some color aspects as well. Coming to color-based approaches, a two-stage strategy is usually used in these cases. Firstly, in the initial stage segmentation is done to narrow the search space [30]. It is then followed by implementation of shape detection to be applied only to the segmented regions. The combination of color and shape features are taken into account for Traffic Sign Detection.

3.5.2 RGB Color Space

The human eye has receptors which are susceptible to color red, blue and green. The perception that every visible color can be decomposed into combinations of these three "primary color", is only possible hypothetically and theoretically. There can be millions of colors by amalgamating and blending different combinations of intensities of red, green and blue on a screen meant for display. Therefore, it would be trivial put across a range of depth for each color, inside the range of 0 to 255 (size of a byte). The term "color depth" is otherwise known as 'range of intensity' [30].

To represent the possibilities of combination of the three primary color together, a 3-dimensional coordinate plane can be used. It can be done with values for R(Red), G(Green) and

B(Blue) on each axis. This coordinate plane after plotting them on those axis produces a cube called the RGB color space [30] Figure 3.4. RGB dimensional coordinates axis:

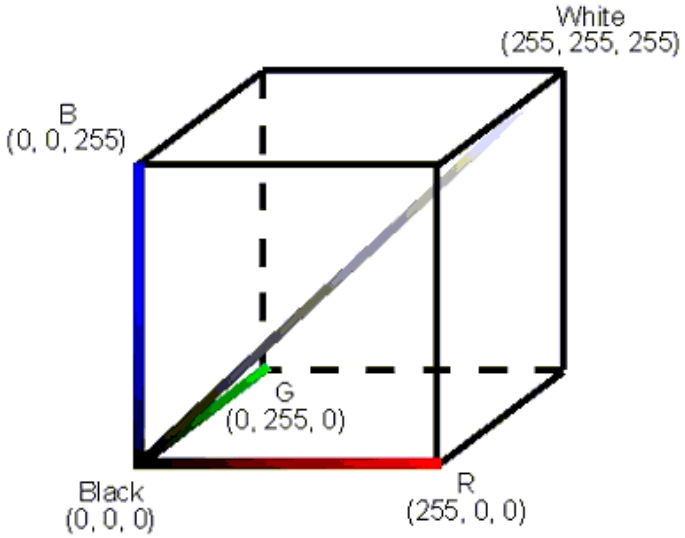

Figure 3.4 - RBG Dimensional coordinates axis

If the values of the 2 channels are Null (0), then it basically implies there is no light being transmitted. Hence the color generated is Black. For example, if the color on a Display monitor which produces no light is the darkest, there can't be anything darker than that. To obtain the resultant color white, we need to set the three-color channels to their highest values i.e. 255 at one-byte color depth. As shown in Figure 3.5. Additive color mixing.

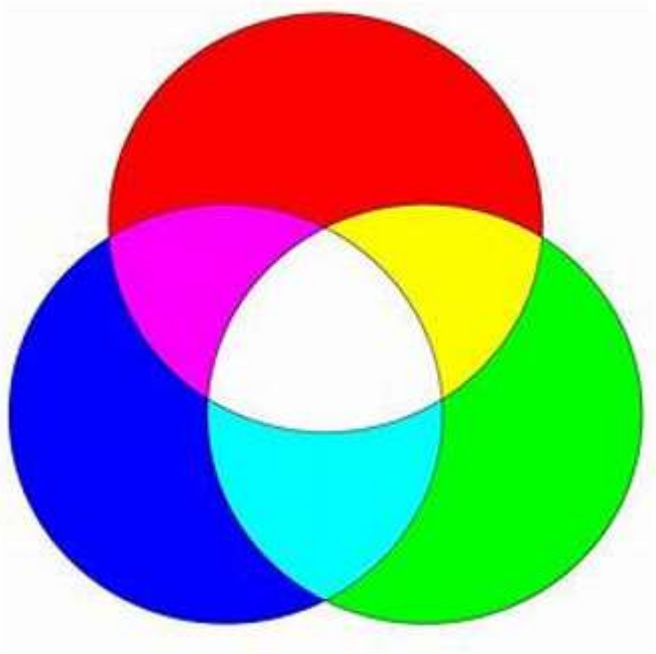

Figure 3.5 - Additive color mixing

Considering the 3-D cube, if a diagonal emanating from black's (0, 0, 0) origin point of the color cube is plotted to the white (255, 255, 255) point, a line is obtained whose every single point possess the same RGB values. The observation here is that if value of R, G, B channels are same, Grey color is obtained. As we travel along this diagonal from originating from lack to white, the depth of shade of grey decreases. It increases, if we travel from white to black [15].

Color is a very essential medium of information. Color can be utilized in numerous applications to handle inspection tasks during the process of image processing. However, as complex it is the human brain has incredible abilities to process colored images. This often results in underestimation of difficulty of a task, when it comes to correctly concurring image processing of color pictures. Fortunately, without overloading the processor there are several applications which just require a color camera which come with useful functions that help handling and processing color images [30].

3.5.3 HSV Color Space

The popular way to amalgamate and produce color is the RGB model, also known as Red, Green, Blue model. The color combination CMYK (cyan, magenta, yellow, key) is widely used in commercial printers. Moreover, the HSV (hue, saturation, value) is the color picker of the graphics software. So, the spectrum perceived by human eyes which explains the process of color combination is done by these methods [16].

In comparison to RGB and CMYK which are mostly used as primary colors, humans' perception of colors is closer to HSV. It consists of these:

- Hue
- Saturation
- value

The scope of this color explains hue or complexion in terms of their shade (saturation or degree of greyness) and their illumination value. Some software like Adobe Photoshop, use the abbreviation HSB, cause the color combination is Hue, Saturation, Brightness (where value is substituted by brightness). However, this should not be confused as in essence, HSV and HSB are the same models [31].

3.5.4 HSV Color Model usage

For better understanding, this color scope is sometimes represented with help of a cone/cylinder. In spite of this fact, the Model is described by these basic components [30]:

A. HUE: The color area of the model given a numerical value within range of 0 to 360 degrees is known as Hue. The angular composition of the other colors is listed below:

- **Value of Red** ranges from 0 to 60 degrees.
- **Value of Yellow** ranges from 61 to 120 degrees.
- **Value of Green** ranges from 121 to 180 degrees.
- **Value of Cyan** ranges from 181 to 240 degrees.
- **Value of Blue** ranges from 241 to 300 degrees.
- **Value of Magenta** ranges from 301 to 360 degrees.

B. Saturation: Saturation describes the greyness in a particular color, with a range from 0 to 100 in terms of percentage. The more the value of the component moves towards zero, more is the grayer and faded effect. At times, saturation appears as a range in between Zero and One, where 0 is gray, and 1 is a primary color [15].

3.5.5 Value (or Brightness)

Value or brightness is always a confluence with Saturation. It describes luminance of the color ranging 0-100 percent, where 0 implies complete blackness and 100 being the brightest, revealing the most color [30].

3.5.6 HSV model usage

Often the color combination of paint or ink is HSV color scope which is chosen by software engineers/ designers. The reason behind this is that the HSV model represents how people perceive colors in a better way than the RGB color model does. High-quality graphics is designed with the help of HSV color wheel. Nonetheless, not very much popular in comparison to its kin; RGB and CMYK. Most of the high-end image editing software use the HSV methodology. The selection procedure begins with by picking up of one of the available hues in HSV color. Then it is followed by adjustment of values of shade and brightness [16].

The shades contain red, green and blue color always. This implies that the components of the color vector used always map to the RGB components of the color. It makes the work easier if color rendering or tint. This results in difficulties while adjusting the hue or saturation. For those kinds of operations, we can use the HSV color space. In addition to the HSV color space there are also other similar color spaces, like the HSL or CIE color models. Some of them are very similar to the HSL model. In addition, the others get way closer to the visible spectrum at their expensive

computation. However, we are concerned about HSV model because this was used to detect traffic signs during nighttime [32] Figure 3.6. HSV model components (Hue, Saturation and Value).

When using the HSV model we also have 3 components which define our color, but in this case, they map to the hue, saturation and value of the color. We can view it as a circle. Only because of the mapping of max and min value of hue to same value (red). To perceive the color space as a cylinder with hue as the rotation around the center, the saturation is the closeness to the center and the value is represented by comparative height of the point in the cylinder.

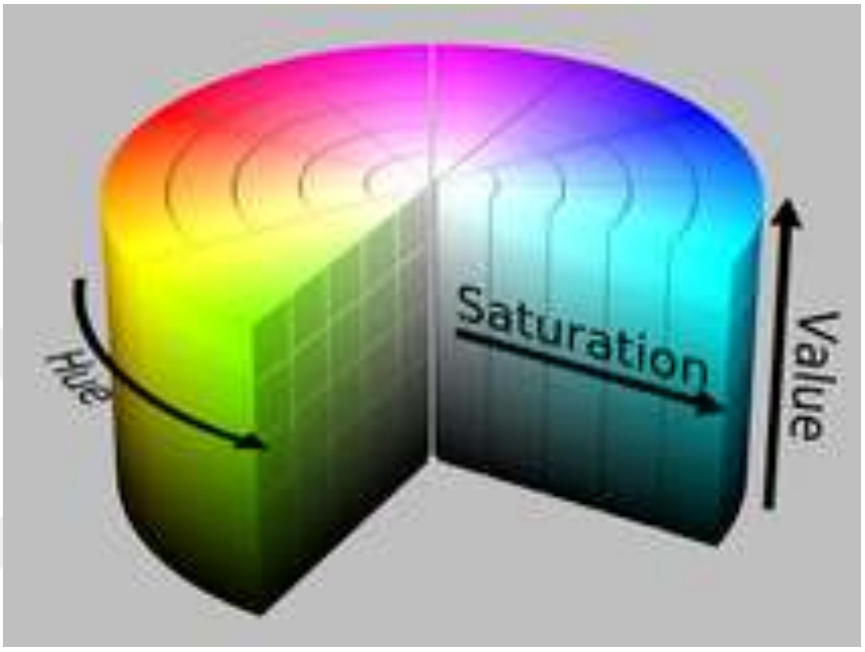

Figure 3.6 - HSV model components (Hue, Saturation and Value)

- The type of color defined by the HSV model space is similar to what is found in advanced RGB and CMYK models where color is defined in terms of their shade (saturation or degree of greyness) and their luminance value. It can be described in the shape of a cone/cylinder. The additive primary and secondary color; red, yellow, green, cyan, blue and magenta along with a linear combination between adjacent pairs of those 3 components [30].

- In the model described above, hue is an angle ranging between 0 and 360 degrees. As discussed, the component color has ranges angle:

    - Red: 0-60 meant for Red
    - Yellow :60-120
    - Green: 120-180
    - Cyan: 180-240
    - Blue: 240-300 meant for blue and 300-360 meant for magenta.

- Saturation indicates the level or the amount of greyness in color space having a range of 0 to 100% or sometimes the range is set between 0-1. For instance, if the value is 0, the color is grey and if the value is 1, it means it is a primary color. Lower saturation level indicates faded color which means the content of grey is more.

- Value: It works in confluence with saturation describing the degree of brightness or intensity of color from 0 to 100 percent. The color space will totally be black when the value is 0 and it brightens up with increase in value showing various color [30].

3.6. OpenCV and Computer Vision

OpenCV or Open Source Computer Vision is vast open source library was developed by Intel for usage in the fields of Computer Vision, Image Processing and many more applications. It comes with the support of an array of programming languages like C, C++, Java, Python, Julia. Computer Vision deals understanding how computers can gather and mimic or enhance high-level understanding from images, videos by processing images and videos to identify objects, signs, faces, digits, even the handwriting of a human. With the integration of various other libraries or frameworks such as CNN, it becomes easier to automate which human visual systems do easier enabling computers' ability to recognize objects such as faces, digits or shapes/signs at great accuracy [33].

Prior to development of such technology, traditional methods were used to extract feature by Hough transform/SIFT, Viola Jones framework for object detection, Kalman Filter for tracking and so on. However, these methods were highly inefficient on their own. So, post adoption and application of OpenCV, which comes with more than 2500 optimized algorithms including both the comprehensive and novel and modern state-of-the-art computer vision and machine learning algorithms, there has been a dramatic change in the landscape of Image Processing. In other words, OpenCV has made the tasks of image processing, segmentation and classification more convenient and efficient. Moreover, if CNN (Convolutional Neural Networks; Deep Learning algorithm) is used along with Open Source Computer Vision library then scientists/engineers are able to predict/classify/segment/process images with more precision and with better accuracy. It's even better than more than the combined use of OpenCV with machine learning techniques that were developed previously. OpenCV and CNN will be discussed in detail with which this application has been developed combined with proper results [34].

3.7. Fundamentals of Computer Vision

There has always been a curiosity to understand how exactly our brain works and function. Which hasn't been completely understood by today's Neuroscientists and Machine Learning experts in spite of possessing state of the art technologies at disposal. However, thus far we have a very limited understanding of the and there are very few working and absolute all-inclusive theories of how exactly the rain works. That's why we can concur that in spite of designing modern Neural Nets aimed to mimic or function in the way the human brain does, we are not quite sure if that's actually the way brain functions and works [33].

This holds true for Computer Vision as we are not sure and since it's not verified how exactly the brain and the eye process information and work. We still can't tell how well the algorithms used in production emulate and replicate our own internal mental processes. To illustrate and explain this lets take some studies which have shown that some functions that take place in the eyes of frogs , were earlier thought to be taking place in the brain. We're very different in functioning from amphibians, but the same doubts exist in human cognition. The way Machines perceive images is pretty basic and rudimentary: as a set of pixels, each associated with its own set of color values [35].

3.7.1. Camera Calibration 3D

To understand the scope of Image Processing better, this chapter will discuss and elaborate in detail about distortions in camera. There are mainly 2 types of distortions; Intrinsic and Extrinsic.

To get a better depth of the subject we need to understand the parameters of camera and how to find these parameters, undistorted images etc. The cheap pinhole cameras that we have today produces a lot of distortion to images which can be problematic [36]. The major distortions encountered are:

- Radial distortion
- Tangential distortion

A. Radial distortion: It results in linear lines to appear distorted. Further the deviation from the center of image, the radial distortion effect is more enhanced and prominent. To illustrate this example better, let us take the case of image shown below. It can be observed there are two edges of a chess board which are marked with red lines distinctly However, it can be observed that the border is not linear. To perceive this better, we can observe this with the tracing along the red line. Other than this, it can be seen that the expected linear lines are lumped outwards [37] .

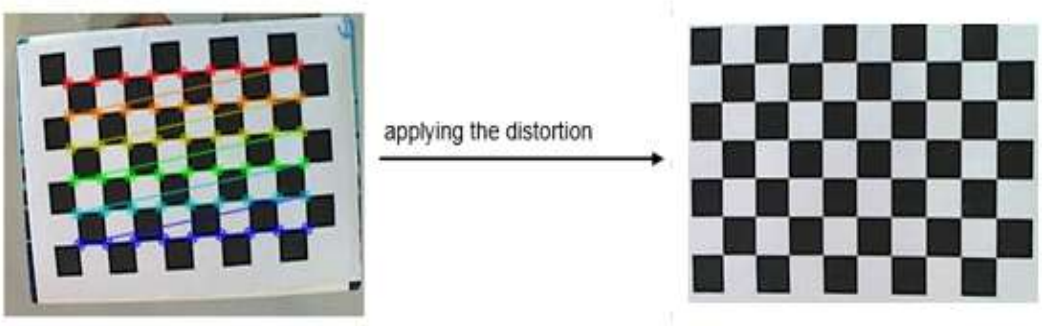

Figure 3.7 - Solving Distortion

To solve this distortion encountered above, we solve it by using these:

Similar to radial distortion, an additional type of distortion is the tangential distortion. Can be seen in Figure 3.7. Solving Distortion. It takes place and can arise because lenses that capture images are not coordinated perfectly parallel to the plane containing the image. So, there can certain regions in image which may appear closer than it is in real. So, all in all it can be concluded that basically five parameters are needed to be found. These are also known as distortion coefficients given by:

$$\text{Distortion Coefficients} = K1\ K2\ P1\ P2\ K3 \quad \textbf{(3.2)}$$

Apart from figuring the distortion coefficients, some more additional information is needed to be found, such as intrinsic and extrinsic parameters of a camera. Intrinsic parameters are particular to image capturing device and the type of camera used as it contains data like focal length, (Fx, Fy), optical centers (Cx, Cy) etc. illustrated in Figure 3.8. Removing Distortion. This is generally known as the camera matrix. As mentioned here, it depends on the camera only. So, once it is known, it can be stored and utilized for future activities. It is expressed as a 3x3 matrix:

$$\text{Camera Matrix} = \begin{matrix} Fx & 0 & Cx \\ 0 & Fy & Cy \\ 0 & 0 & 1 \end{matrix} \quad \textbf{(3.3)}$$

Figure *3.8* - Removing Distortion

B. Extrinsic parameters: They are as critical as the internal ones and they correspond to rotation and translation vectors. This translates a coordinate of a 3D point and maps it to a coordinate system.

In order to be able to use it for stereo applications, these distortions need to be corrected first. To find and figure out all these parameters, some sample images of a well-defined pattern for example chess board need to be provided and processed beforehand. Some specific points in the square corners in chess board we have taken for example are found. The real-world coordinates space and its coordinates in image are known and with these data, some mathematical problem is solved in background to achieve and get hold of the distortion coefficients. To arrive at better results, at least 10 test patterns are needed as more empirical information results in better accuracy [37] .

3.7.2. Image Processing

Image processing is done to analyze, enhance or optimize the features of an image by operating and applying some techniques on an image, in order deduce some useful information from it. It can also put as a type of signal processing in which an image is received by the Image processing method and an image or characteristics/features associated with that image is generated

from it. In today's day and age image processing is among quickly growing technologies so much so that it has become one of the critical areas for research within engineering and computer science disciplines too.

Image processing broadly comprises of these steps:

- Obtaining image by the help image acquisition tools.
- Analyzing and modifying the image to make it suitable for next step.
- Deriving output in which can result in modified image or report that is based on image analysis.

Primarily there are two types of methods used as they are concise for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and pictures. A lot of the foundation work of deciphering and understanding is utilized by Image Analysts to determine the methods while using the visual techniques. Computers help in modifying and manipulating digital images with the aid of Advanced Digital image processing. The three general phases that all types of data have to undergo while using digital technique are [38]:

- Pre-processing
- Enhancement
- Display, information extraction

3.7.3. Difference between Image Classification and Segmentation

From an objective viewpoint, Object classification and segmentation — both are part of machine learning based image processing which are used as training sets for AI algorithms through computer vision. Both of them are equally important and necessary for object recognition precisely in the field of machine learning and AI development [39].

However, Object Classification as name suggests is a kind of more precise categorization of objects in an image of a single class. Whereas Image Segmentation is used to simply classifies the two different objects in a single image. Image annotation techniques are used for classifying such objects. On the other hand, in semantic segmentation, the objects are detected, classified and segmented for computer vision processing [40].

3.7.4 Image Classification Working Principles

To understand what Image Classification is. It can be described as the task of obtaining the information classes from a multiband raster image. It analyzes the numerical characteristics of different image features and performs data organization by classifying objects into different types.

Overall, one can say image classification is like image categorization. In fact, data classification algorithms typically employ two stages of processing; training and testing. First, characteristic properties of image features are separated based on which a unique depiction of each classification division is created as shown in Figure 3.9. Traffic Sign Classification Process.

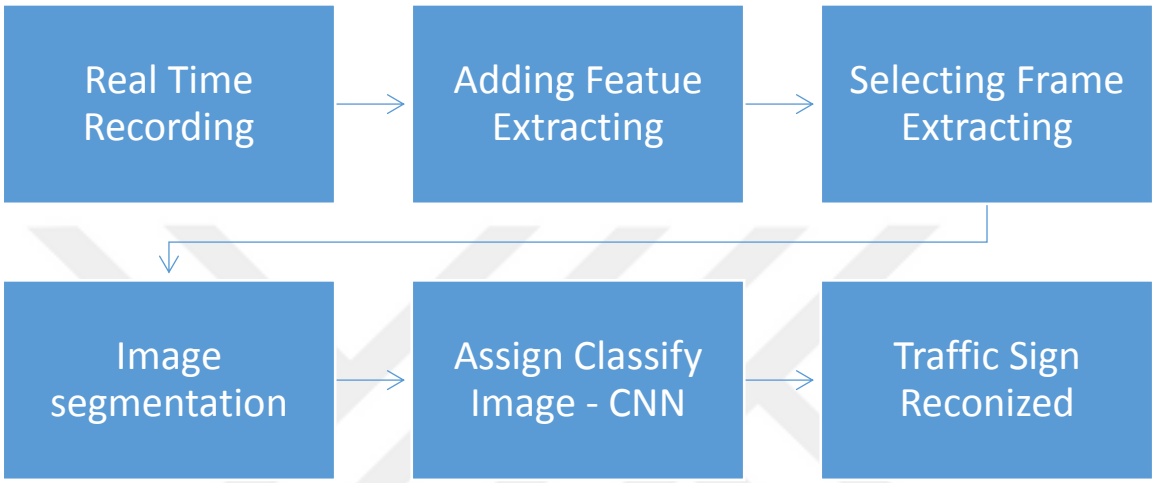

Figure 3.9 - Traffic Sign Classification Process

And after the first stage is done with, these character space partitions are then used to allocate the images properties can be identified uniquely from each other. In the field of machine learning, image classification is used for both; supervised learning and unsupervised learning.

In practice both supervised and unsupervised classification are pixel-based classification process which creates square pixels where each pixel has a class. On the other hand, object-based image classification clubs' pixels into representative shapes and sizes [41].

3.7.5 Image Segmentation in Image Processing Processes

Segmentation in the context of Image Segmentation can be described as the set of methods followed in converting the image i.e. the breaking down the digital image into more segments. In other words, the digital image is broken down into the set of different pixels. The aim of breaking down is to simplify or change the way an image is presented into an easier format or a more convenient format by making to more meaning for machines to analyze and understand better. It can also be described as a method of putting a designation to each pixel present in the image in a way that pixels with the same designation possess certain features, it can be shown in Figure 3.10.

traffic Sign Segmentation. Its primary objective is to locate objects and boundaries such as lines and curves present inside the images [40].

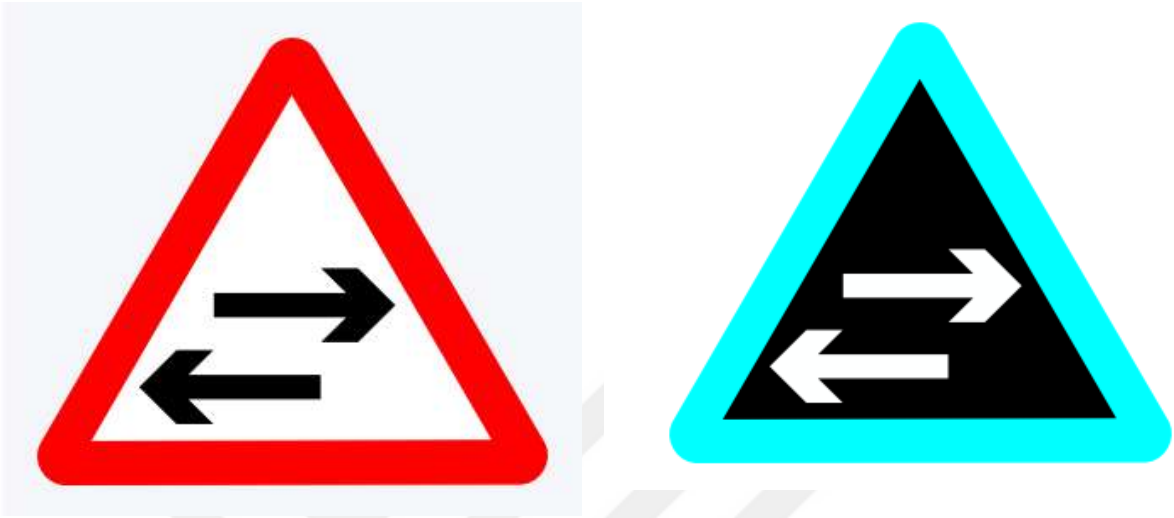

Figure 3.10 - Traffic Sign Segmentation

To achieve a more accurate view of the image Semantic Segmentation is used. It can recognize and understand what exactly is in the image at basic level view in a single class to provide certainty for computer vision view to the machines and also useful in detecting and classifying the object in an image when there is more than one class in the image. Hence, there are two popular techniques are used — Semantic segmentation and instance-based Segmentation is used for objects nested classification create objects having separate regions.

The difference between segmentation and classification is clear at some extend. And there is one difference between both of them. The part of allotment to categories is easier than distribution and segregation, in classification all objects in a single image is are clubbed under same label or allocated into a single class. While in distribution part, each object of a single class in an image is accentuated with different color to make them discernible to computer vision [40].

Image segmentation is one of the key topics in the aspect of image processing and computer vision as it can have several uses in different fields such as scene understanding, medical image analysis, robotic perception, video surveillance, augmented reality, and image compression, among others. To achieve these applications, several algorithms for image segmentation have been developed in recent past. Owing to the success of deep learning models in a wide range of vision applications, a considerable amount of work dedicated to developing image segmentation approaches using deep learning models have been developed [42].

3.3.6. Noise Filtering

Any arbitrary deviation of image Intensity and visibility generally represented as grains in the image can be described as Noise. It can come across in images as effects of photon nature of light or heat emanating from the image sensors which are few of the common reasons for causing it. It can be generated at the time of capturing or image transmission. Because of it, the pixels of the image show different intensity values than the original real theoretical pixel values that are generated from the image. Therefore, to remove it, Noise removal algorithm is used which is the process of taking out or lessening of the noise from the image. The noise removal algorithms reduce or remove the visibility of noise by smoothing the entire image leaving areas near contrast boundaries. As a result of which these methods can cause cloudiness and lowering of contrast details. Few of the types of noise that are encountered are [43]:

- Impulse noise
- Additive noise
- Multiplicative noise

3.3.7. Sources of Noise in Images

Noise is introduced in the image during the process of gathering or transmission stage. There can be several factors which can be responsible for generation of noise. Quantity of pixels affected decides the magnitude of the noise. Some of the common sources of noise in the digital image are:

- Imaging sensor of the capturing device could be affected by environmental conditions while image acquisition
- Lower Light levels and sensor temperature
- Obstruction of signal in the transmission channel
- Lightweight dust particles present on the scanner screen that obstruct while capturing of input image [43].

3.7.8. Different types of Noise

Noise or any deterioration in quality of the image signal can occur due to external disturbances as well. During transmission or sharing of an image from one place to another electronically by satellites or wirelessly, glitches may appear in the image signal. These glitches

can pop up on the image output in different avatars, it depends on the cause of interference. As we are aware of common type of glitches and the kind of noise on the image; we can identify some of the regular noise for eliminating noise in colour image. Image Noise can generally be classified under as [43]:

- Amplifier noise (Gaussian noise)
- Salt-and-pepper noise (Impulse noise)
- Shot noise
- Quantization noise (uniform noise),
- Film grain
- On-isotropic noise

3.8 Filter Images and Videos

This is a neighborhood operation during which the value of any given pixel in the output image is found out by using a series of steps to the pixel values in closeness of corresponding input pixel. The objective of using such operation is to smoothen, sharpen and detect edges of images and videos [44].

3.9 Smoothening/Blurring of Images

During the image acquisition and capture by a camera or any other input method, the image may get deteriorated by random dots and noises. This is one of image processing techniques implemented to get rid of such glitches and enhance the overall quality of the image. It is applicable for all the frames of a video to get rid of blemishes and glitches to better the quality of the video. The blurring and smoothening techniques are listed below [45]:

- Homogeneous Blur
- Gaussian Blur
- Median Smoothing
- Bilateral Smoothing

3.10 Image De-Noising

Techniques like Gaussian Blurring, Median Blurring etc. and can work well to a certain degree in getting rid of small amount of noise for the enhancement of the quality of output image. In these mentioned techniques, a limited region around a pixel is and operations like Gaussian

weighted average, median of the values etc. are performed on it to put in place of the central element. Briefly, noise elimination at a pixel is local to its neighborhood.

Noise is generally considered to be a random variable with zero mean. Let us take the case of a pixel with high quantity of noise, p=p0+n where p0 is the actual value of pixel and n = noise in that pixel. So, one can take high magnitude of same pixels (say N) from multiple images and compute their mean. Ideally, we should get p=p0, as discussed mean of noise is null.

This can be proved and hence verified by a basic setup including a static camera pointed to a particular area for a short period of time which will result in multiple frames, or a lot of pictures of the same scene. After deploying, some amount of code to find the mean of all the frames in the video the final result is compared to the original and first frame noise reduction can be seen. However, this easy way is not applicable or powerful enough for camera and scene motions [46].

The basic thought behind is that we require a lot of similar pictures to average out the noise from the other ones. Let us imagine a small frame (say 5x5) in the image. There is a strong possibility that the same frame may also be present at a different location in the image and in certain cases in a small neighborhood around it. To solve this, we could use these similar frames in combination and find their mean as shown in Figure 3.11. Applying Image De-Noising. It could work well for that definite window in case. Let us illustrate this case with the image below:

Figure 3.11 - Applying Image DE-Noising

Blue patches in the image all look identical to each other. Similar in the case of the blue ones, Green patches look similar. Therefore, a pixel is taken with a small window around it searching for windows matching the one in the picture, averaging all the windows and replacing the pixel with the output obtained. It is called as Non-Local Means DE-Noising which occurs a lengthier period of time in contrast to smoothening and blurring techniques, but the resulting output is satisfying. For color images, image is converted to CIELAB color space and then it independently eliminates noise in L and AB components [46].

3.11 General Object Detection Framework

- Typically, there are three steps in an object detection framework. For Producing regions of interest or region proposals, a particular Algorithm or Model is utilized. These region proposals can be defined as basically a set of bounding boxes covering the full image that is, an object localization factor.
- Post application of algorithm to get region proposals, visual characteristics are obtained for each of the bounding boxes. These are then assessed, and it is if objects are present in the proposals based on visual features. That is basically identifying an object classification component.
- Finally, all overlapping boxes are collated into a solitary bounding box

3.12 Object Tracking by Kalman Filter

Kalman Filter is a type of linear filter, which is discrete in nature. It has a very high computationally potent as well as recursive which is highly useful in tracking objects. It is useful in cases where the capability of model is unknown and not accurate, but in cases, we need to approximate past state of system along with the estimate of the present and future state of it. Taking the case of Road Sign Detection where the object to be traced is tracked until it arises on the scene, Kalman Filter comes very handy. Speaking of the Algorithm, there are two primary components: System and Noise Models.

$$Xt = (Xt * Vt)2 \quad \textbf{(3.4)}$$

Regarding above equation Xt and Vt are the real position of the targeted objection and the speed of the targeted moving object, respectively, and T indicates the overrun. The given dynamic model is a Computer Vision (CV) model [47].

3.12 CNN (Convolutional Neural Networks) and Machine Learning

CNN or Convoluted Neural networks is one of the most known algorithms in the field of image classification and consists of Convolution layers, Activation function layers, max pooling of layers to decrease the dimensionality by not letting go of a lot of features. A feature map is generated by the ultimate layer of CNN. Let's say, if we input a Cat or a rabbit image, the algorithm

m will come back by telling whether it's a cat or a rabbit. Its use is not just restricted to that, it is laced up with some remarkable computational abilities which produce great advancements [49].

Several models, which are trained earlier, are furnished so that one does not have to go through the trouble of educating the system arising cause of restrictions in computation. Many models got popular as well like VGG-16, ResNet 50, DeepNet, AlexNet by ImageNet. Faster R-CNN is used most of the time to recognizing the objects. However, the objective of creating this algorithm was to particularly deal with the logic and maths of getting the bounding box around the recognized objects. The people who came up with this algorithm named it Region Proposal Networks, short for RPN. For generating these "proposals" for the region where the object resides, a small network is slid over a convolutional feature map, which is the output of the previous convolutional layer [17] Figure 3.12. CNN Classifier Procedure.

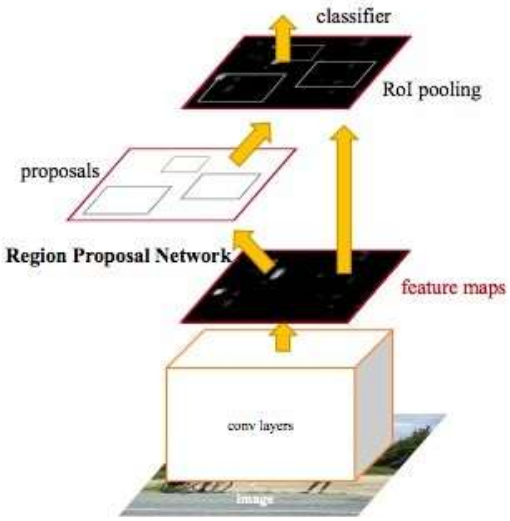

Figure 3.12 - CNN Classifier Procedure

The illustration of the CNN is shown above in the picture. RPN generates the proposal for the objects. RPN has a specific and exclusive architecture in itself. RPN architecture is further broken down as shown below Figure 3.13. CNN Classifier Layer.

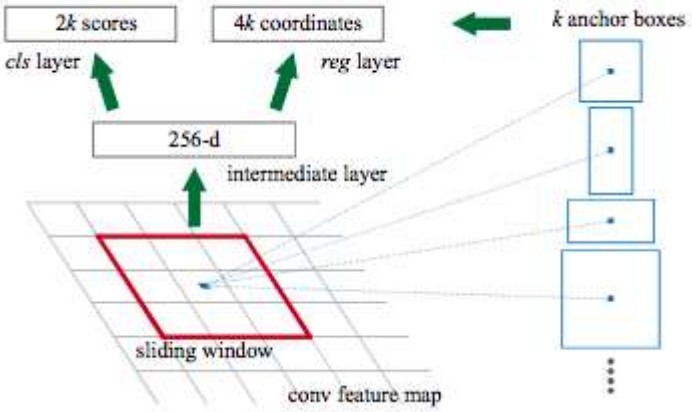

Figure 3.13 - CNN Classifier Layers

RPN consists of a classifier and a regression. The makers of the advanced algorithm came up with the concept of Anchors, which is the focal point of the sliding window [17] Figure 3.14 Feature Extraction and Classification in CNN. For ZF Model, which was an extension of AlexNet, the dimensions are 256-d and for VGG-16, it was 512-d.

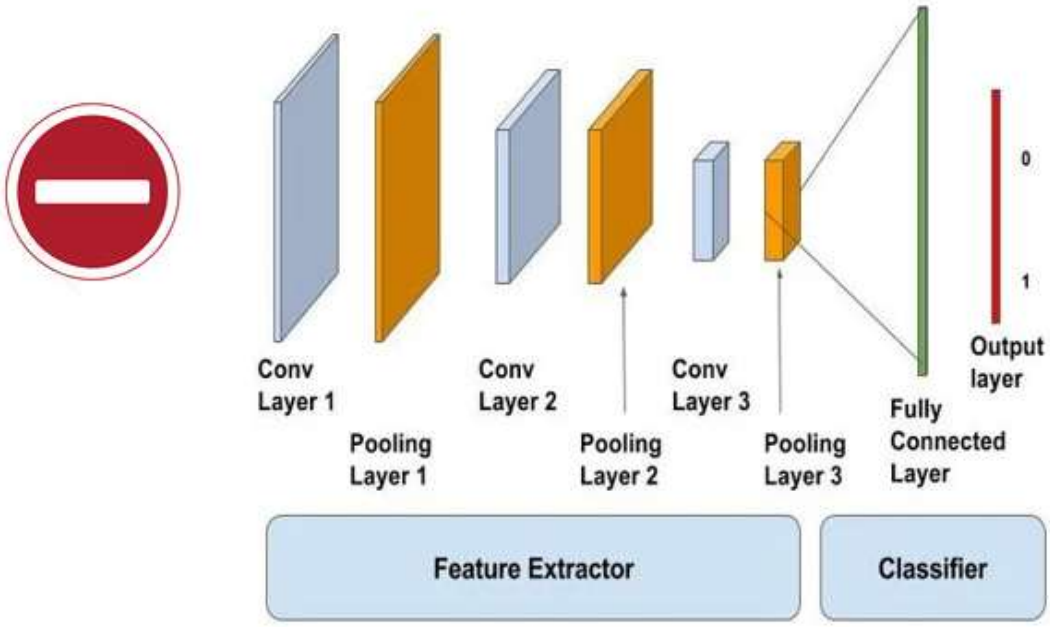

Figure 3.14 - Feature Extraction and Classifier in CNN

Classifier regulated the possibility of a proposal having the desired object. Regression regresses the coordinates of the proposals. For any image, scale and aspect-ratio are two important parameters, the aspect ratio can be defined as the width of image divide by the height of image

where scale is the extent/size of the image. 3 scale and 3 aspect-ratio were chosen in this model. So, total of 9 proposals can be obtained for any given pixel, K=9 for this case, k can be defined as the quantity/magnitude of anchors. So, for the entire picture, the total magnitude of anchors is given as the product of width, height and K; W*H*K. This algorithm is robust against translations, that is basically this algorithm is not dependent on translation.

If there are anchors of different scale in the algorithm, it results in "Pyramid of Anchors" not "Pyramid of Filters" making the process less time exhausting and more economical cost wise than any other algorithms. These anchors can be assigned labels depending on:

- The anchors with the most magnitude of Intersection-over-union overlap with a ground truth box.
- The anchors with Intersection-Over-Union Overlap more than magnitude of 0.7.

Ultimately, RPN is an algorithm that needs to be trained properly definitely has a Loss Function which can be calculated by [49]:

$$L(\{pi\},\{ti\} = \left(\frac{1}{Ncls}\right) * \sum Lcls(pi, pi) * +(Nreg + Lreg)) \quad \textbf{(3.5)}$$

i → Index of anchor, p → probability of being an object or not, t →vector of 4 parameterized coordinates of predicted bounding box, * represents ground truth box. L for Loss, cls represents Log Loss over two classes.

$$Lreg(ti, ti^*) = R(ti - ti^*) \quad \textbf{(3.6)}$$

p* with regression term in the loss function makes sure that, the only condition in which object is identified as yes, then only regression will count, otherwise p* will be null and hence the regression term will become zero in the loss function as well. Ncls and Nreg are the terms representing normalization. Default value of is positive and is done to scale classifier and regression on the same level [17]. Can be seen in Figure 3.15. Traffic Sign Recognition Process.

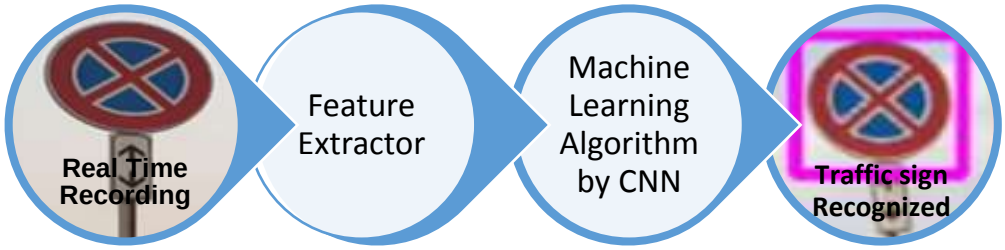

Figure 3.15 - Traffic Sign Recognition Process

3.13 Training Process of TSDR

Image processing is done to analyze, enhance or optimize the features of an image by operating and applying some techniques on an image, in order deduce some useful information from it. It can also put as a type of signal processing in which an image is received by the Image processing method and an image or characteristics/features associated with that image is generated from it. In today's day and age image processing is among quickly growing technologies so much so that it has become one of the critical areas for research within engineering and computer science disciplines too. Image processing broadly comprises of these steps:

- Obtaining image by the help image acquisition tools.
- Analyzing and modifying the image to make it suitable for next step.
- Deriving output in which can result in modified image or report that is based on image analysis.

3.14. German Traffic Sign Dataset (GTSRB) Description

To implement a recognition, we need to have a dataset, in all computer vision the dataset it would be an essential data to be compared with the detected to be recognized. in order to train an algorithm based on the images are used we need of training data is When doing computer vision projects, training data and test data is essential. If algorithms that train themselves based on images are used, the need of training data would be obvious. In Figure 3.16. German Traffic Sign Recognition Benchmark (GTRSB) Dataset. So, in this project German Traffic Sign Recognition Benchmark (GTSRB) Dataset is used as essential database and the algorithm is Convolutional Neural Network (CNN).

Figure 3.16 - German Traffic Sign Recognition Benchmark (GTSRB) Dataset

4 RESULTS AND DISCUSSION

In this chapter focused on the results and experimentation of the thesis. This includes deliberation of real-time tests carried out for the application as well as their results. To design the project, a smartphone is needed with Android operating system to install the application on it and subsequently opening the camera to record in real time. Installing Android Studio to write and implementing the program on it using Java language. Android Studio is the official Integrated Development Environment (IDE) built on JetBrains' IntelliJ IDEA software. This is a software is designed specifically for Android development and is also recommended by Google. It has better features and is user-friendly in comparison to Eclipse which is why this is the most suitable IDE for designing applications is Android.

4.1 Requirements for Installation on Windows Machine

- Minimum 4GB RAM and recommended memory is of eight gigabytes.
- Minimum disk space required is two gigabytes and recommended is four gigabytes where 500 megabytes is needed for integrated development environment and 1.5 gigabytes for Android software development kit and emulator system image.
- The system resolution's minimum specifications required is 1280 x 800.
- The Operating system which are supported are Microsoft Windows 7/8/10 (32 or 64-bit). The Android Emulator supports 64-bit Windows only. It also supports Mac OS.

4.1.1 Installation steps for Windows Machine

To install, the recommended step is to be followed which is double clicking the .exe file in order to start the installation process. If there is a zipped file instead of a system installer file. Then unzip the .zip file and copy he android-studio folder into the Program Files folder (usually in C drive or it may be other drives depends on the User's preference). After that open the android-studio then go to bin folder and open the studio64.exe by double clicking it (for a 64-bit machine) or studio.exe (for a 32-bit machine). For the next steps are the installation of any recommended packages or those packages needed by the user as per the requirements. If any problems are encountered during installation of packages, then Android Studio setup wizard can be followed in order to resolve the issues if any.

4.2 Adding Custom SDK Tool

Custom Installation option can be chosen if the user has software development kit files available with him/her. Then during installation process a pop-up arises asking you to provide the installation location in the SDK components once the SDK option is unchecked. Whenever there is a new update available for the software then a pop-up arises stating new version updated is available and click to download, provided your system has internet access. Then the user can download and reload the software to install the updates. If the user manually wants to check for updates, then he/she can browse the Help option in the IDE and click on 'Check for Updates' option.

4.3 Adding OpenCV Library 300

OpenCV or Open Source Computer Vision is a vast open source and cross-platform library which was developed by Intel for usage in the fields of Computer Vision, Image Processing and many more applications. It comes with the support of a wide variety of programming languages like C, C++, Java, Python, Julia. Computer Vision deals understanding how computers can gather and mimic or enhance high-level understanding from images, videos by processing images and videos to identify objects, signs, faces, digits, even the handwriting of a human as well as image enhancement, panorama stitching, etc. as seen in Figure 4.1. Project Architecture.

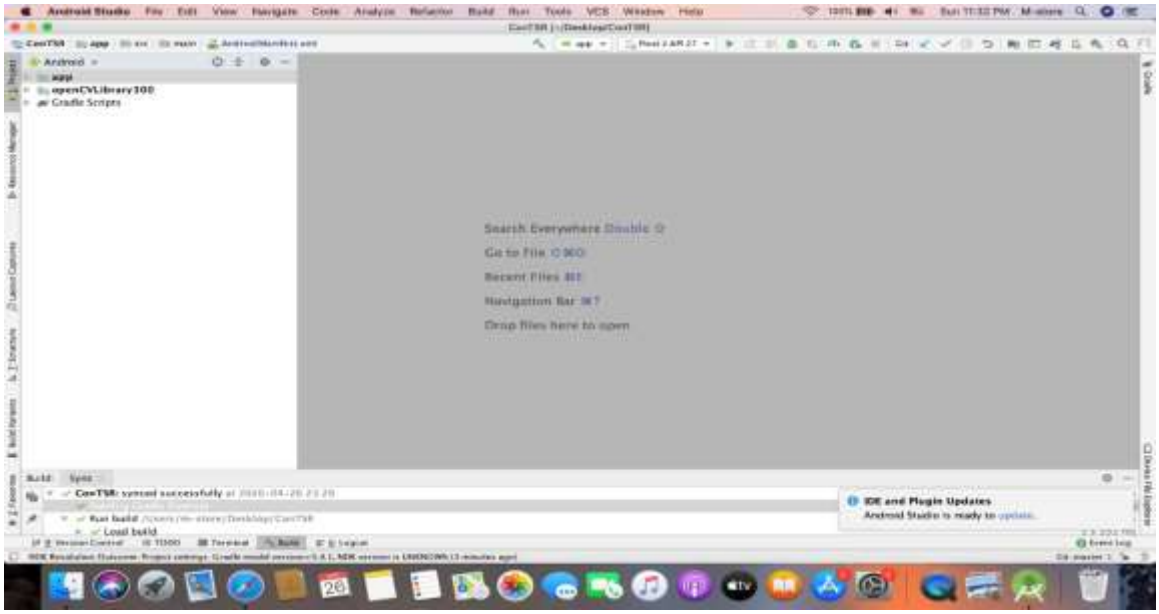

Figure 4.1 - Project Architecture

OpenCV manager provides the best version of the OpenCV library for the hardware of the User. This also receives the latest stability and performance updates for the library provided the

system has an internet access. And this way the user can add the OpenCV library to the Android studio and start working on the project shown in Figure 4.2. OpenCV files. It contains in three main parts are:

1- App
2- OpenCV library
3- Gradle scripts

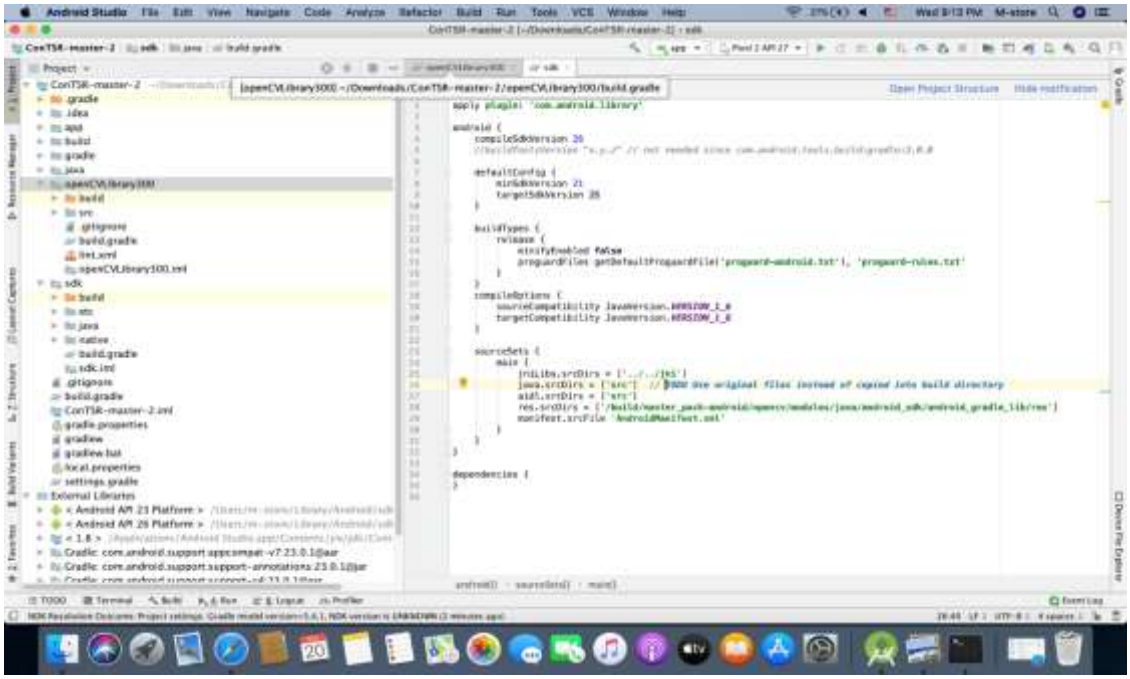

Figure 4.2 - OpenCV files

4.4 GRADLE

The entity which automates the build process, run testing as well as deployment tasks, compiles jar files, creates documentations and much more is known as Gradle. It is a very powerful open sourced build automation tool which supports Java virtual machine languages and this automation is possible via “Build.gradle” scripts. It can be easily understood considering a simple example of files being copied from one directory to another with the help of ‘build.gradle’ scripts before the start of real build process.

4.4.1. Necessity of Gradle

The generation of apk from .java and .xml files is quite trivial in all of the Android projects without which installation is not possible. To elaborate it, all the source files (java and XML) are taken by gradle and then befitting tools are applied: conversion of dex files from java files and the

last step involves all of the files being zipped into a single file known as apk used for installation on the consumer devices in order to use that software. ‘build.gradle’ has two different types of scripts one being the top level and the other being the module level scripts which are discussed briefly below.

4.4.2 Top-level build.gradle

Top level build.gradle scripts’ main objective is to specify the build configurations which can be administered to each of the modules involved in the project. The location of these scripts can be found in the root project directory and the application of the above process is discussed in the below steps as Figure 4.3. build.gradle.

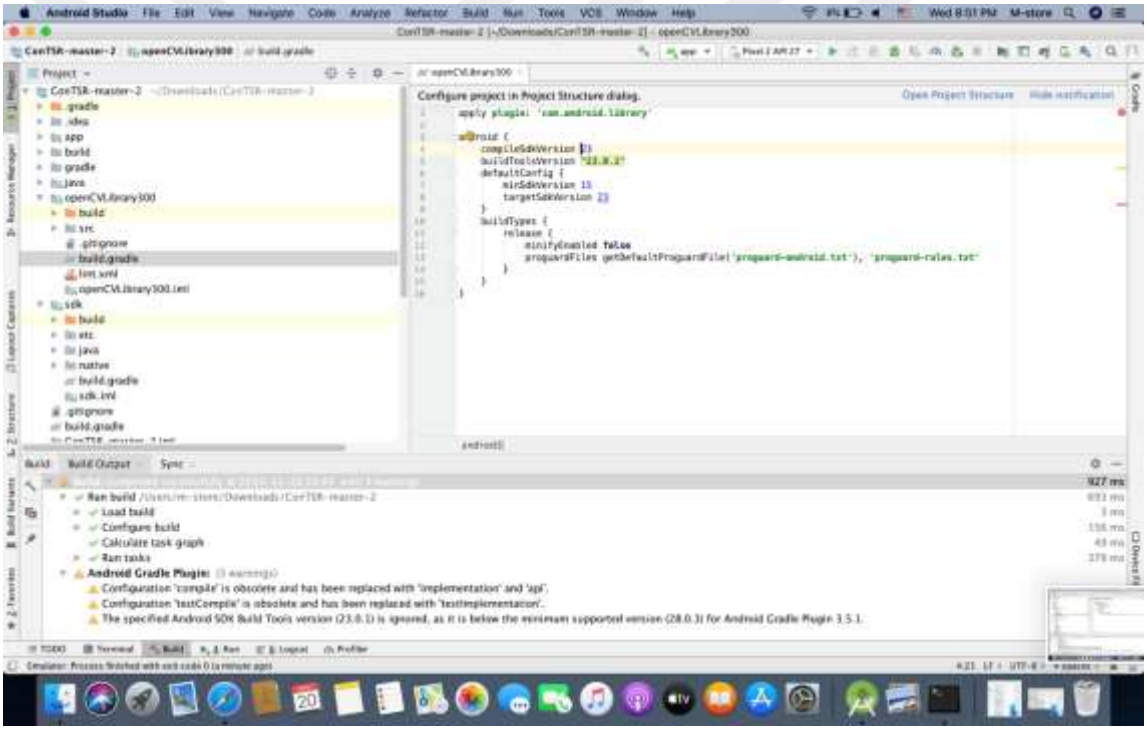

Figure 4.3 - Build.gradle

Different types of conversion of a program into lower level form for the top level are described in three brief steps as shown below:

1. **Buildscript**: The configuration of storage places for Gradle is done over here which is the main objective. The dependencies are also taken care of as well. Moreover, the dependency block helps in the program conversion to machine level code by configuring dependencies.

2. **Allprojects:** It is quite frivolous especially the inclusion of JCenter and Google's maven repository by default in the freshly created projects carried out by the android studio. The main task handled by this block is the configuration of the third-party plugins or libraries.
3. **Task clean (type:Delete):** To keep the project clean during the modification of some configuration files is done by the deletion of the directory every time the project is executed. For instance, a complete clean of the system is needed during modification of settings.gradle files.

4.4.3 Module-level build.gradle

There are many applications of this script. Two of them listed here are the additional type of program conversions into machine level code and settings override in the main/app manifest or *top-level* file. The location of module level build.gradle scripts can be found stored in the project/module directory where all the interdependencies are defined and also encloses the declarations of the sdk versions shown in Figure 4.4. App files and Architecture.

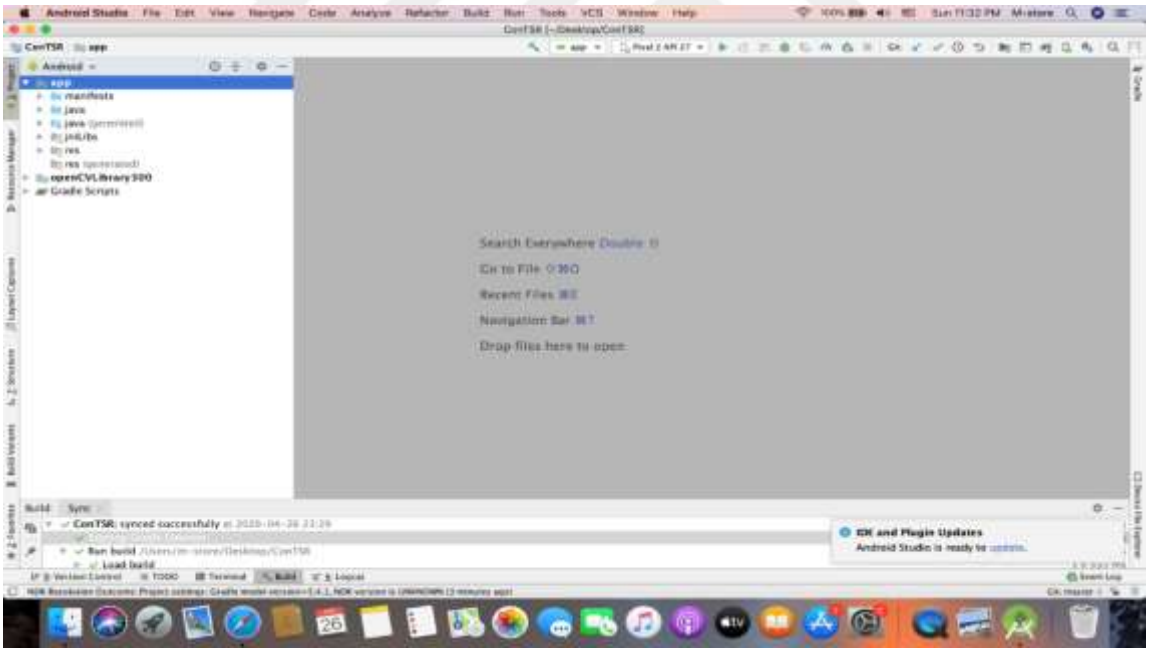

Figure 4.4 - App Files and Architecture

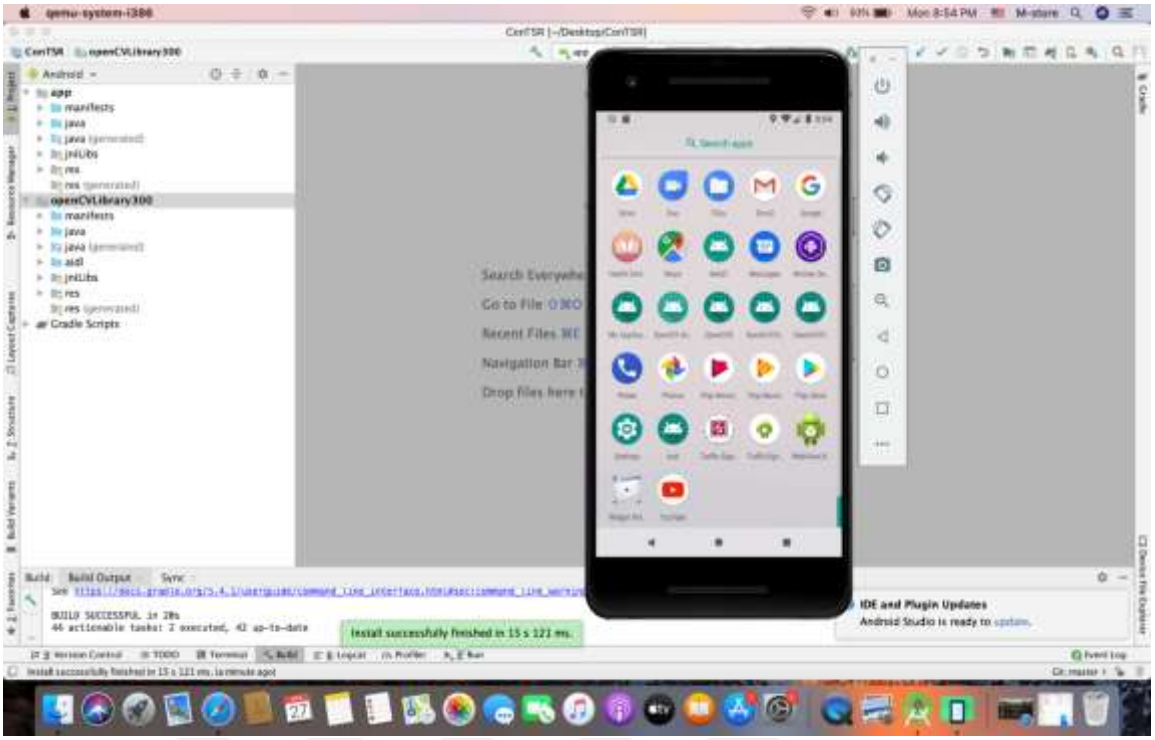

Figure 4.5 - Launching the Application

4.5 Android Manifest

The resource file which envelopes all the details of an android application is known as Manifest file. Manifest file being an xml files must be named as AndroidManifest.xml. It is a must for every Android application to have a manifest file and this manifest file is situated in application root. The system can't execute any of the application's code if it doesn't have the fundamental information of the application which is given by the Manifest to the Android system. The naming of the Java package for the application is also handled by the Manifest file which is one of the other functions of this file. As shown in Figure 4.5. Launching the Application. It is a .xml file and must always be named as AndroidManifest.xml. It is a key file working as a bridge between the android developer and the platform. Furthermore, helping the developer to pass on functionality as well as requirements of the application to the Android and is stored at application root. The application's necessity like packages, API, libraries is defined by the Manifest.xml file as well as details about permissions, set of classes needed before launch and basic building blocks of application like activities, services etc. As shown Figure 4.6. Android Manifest.xml file

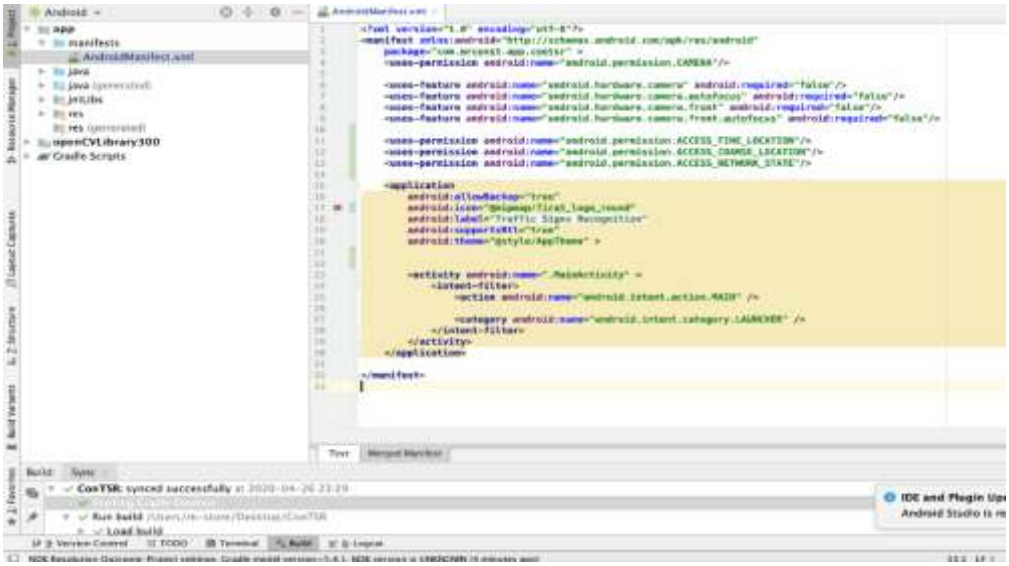

Figure 4.6 - Android Manifest.xml file

4.5.1 The importance of Android manifest file

Every project in Android includes a manifest file, which is Android Manifest. xml stored in the root directory of its project hierarchy. The manifest file is an important part of our app because it defines the structure and metadata of our application, its components, and its requirements.

4.5.2 Elements of AndroidManifest.xml

Elements for Application Properties that can show up in AndroidManifest.xml are shown below and no elements can be added to the list manually because it is restricted.

- Uses-permission – Some specific permissions are requested for security purposes which are established by uses-permission.
- Permission – The ‘permission’ element helps gain access to some specific component of the application. This is possible only because of the certain kind of privileges installed by this element.
- Permission-group – Instead of permission, which is carried out for a particular component, the group element enforces it for a set of components.
- Permission-tree – It could indicate to the owner or else the root node of the set of components.

- Instrumentation – Acts as an interface between android system and app to allow communication.
- Uses-sdk – The application's platform congruency is defined by this element.
- Uses-configuration – The requirements of the hardware and software set is defined by this element.
- Uses-feature – The single hardware and software requirements as well as the associated environment is defined by this element.
- Supports-screens, compatible-screens – These two elements handle the configuration mode and size of the screen.
- Supports-gl-texture – The filtering of application is established on the texture's specification carried out by this element.

4.5.3 The Elements of components of the Application

These should be enclosed in <application> container.

- Activity – Front end screen defines the set of attributes enveloped by this element.
- Activity-alias – The target activities are specified by this tag.
- Service – It is also known as a web-service running in the background (invisible) whose task is to provide key facts from the back-end code and the operation is provided by any library or application programming interface.
- Receiver – The message broadcasted by an application or even by outside system is received by the same application with the help of this tag.
- Provider – Accessing application data requires some structure provided by this element
- Uses-library – For the application to be executed a set of library files is defined by this element.

The system has to acquire all the information cited above to execute any file of the application. Thus, the creation of this file should take place not at the time of running the application but rather during installation.

4.5.4 Structure of AndroidManifest.xml

This file's location should be the root folder of application. And the file must be named as AndroidManifest.xml. Elements must be present in the file. They are allowed only once. The elements which do not depend on any particular order are present inside. All values of the elements should be endowed as attributes. It should not be supplied as CDATA. The Android manifests scripts of the project are mentioned in below:

“

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.mrconst.app.contsr" >
  <uses-permission android:name="android.permission.CAMERA"/>

  <uses-feature android:name="android.hardware.camera" android:required="false"/>
  <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
  <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
  <uses-feature android:name="android.hardware.camera.front.autofocus"
android:required="false"/>

  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

  <application
    android:allowBackup="true"
    android:icon="@mipmap/firat_logo_round"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme" >


    <activity android:name=".MainActivity" >
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>

</manifest>
```

”

4.6 Android Activity

An activity represents a single screen with a user interface just like window or frame of Java. Android activity is the subclass of ContextThemeWrapper class. When the program is written in C/C++, Java programming language or any object oriented language usually starts with a main() function similarly when the program coded in an Android system is initiated in an Activity starting with a call to onCreate() callback method. To initiate an activity a chain of callback methods and

then to further break it down is illustrated in [48] Figure 4.7. Android Activity Framework. Also considering as an Activity life cycle diagram.

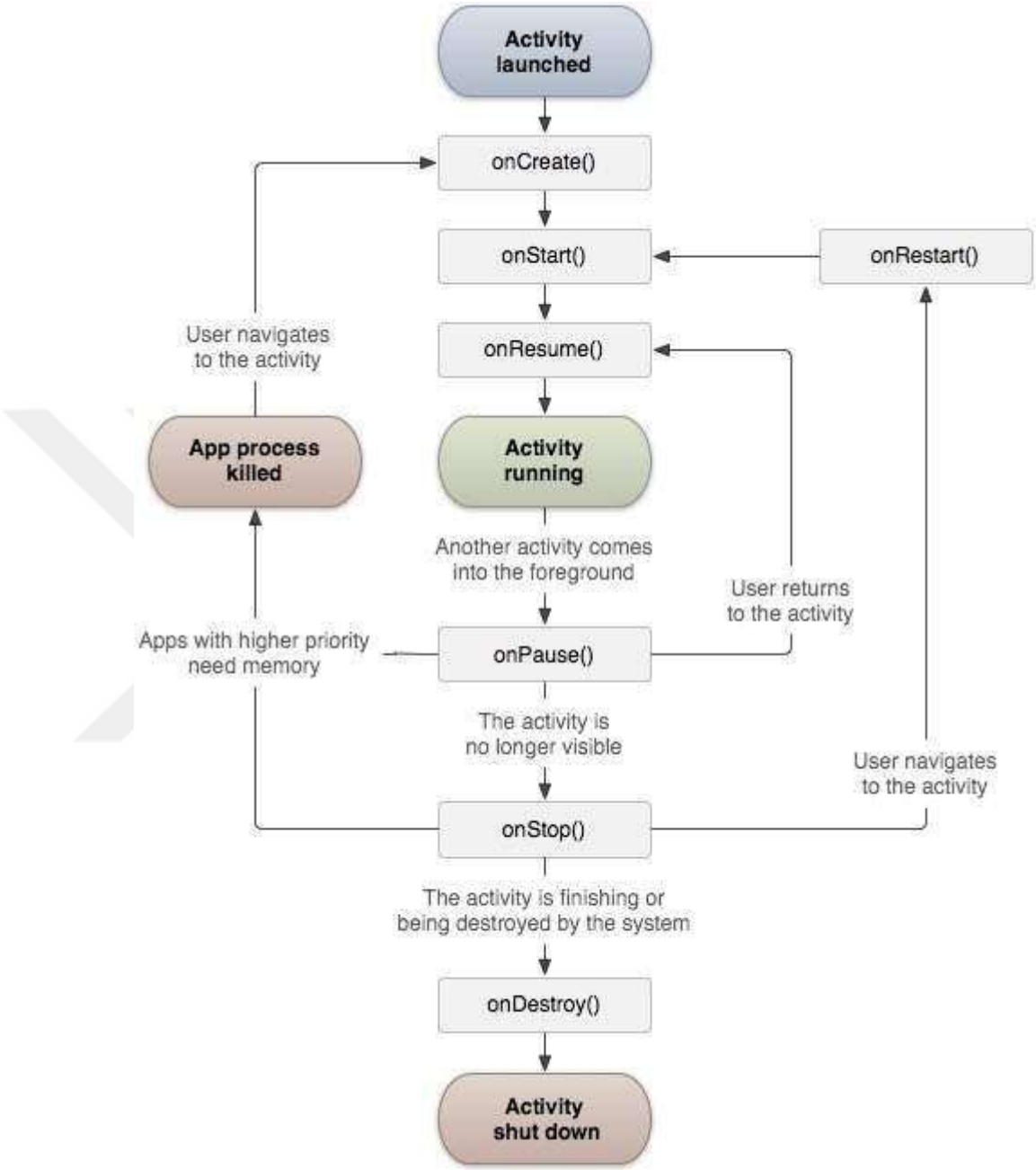

Figure 4.7 - Android Activity Framework

The Activity class defines the following call backs i.e. events and all the callback methods need not be implemented. However, it's important for the engineers to understand each one of the callback methods. Then those methods are to be implemented based on the understanding of each of the callback methods which ensures the app behaves the way users expect.

Scripts:

“

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   xmlns:opencv="http://schemas.android.com/apk/res-auto"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin"
   tools:context=".MainActivity">

   <org.opencv.android.JavaCameraView

      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:visibility="gone"
      android:id="@+id/HelloOpenCvView"
      opencv:show_fps="true"
      opencv:camera_id="any" />
</RelativeLayout>
```

“

4.7 Kalman Filter

Kalman Filter is a type of linear filter which is discrete in nature. It has a very high computationally potent as well as recursive which is highly useful in tracking objects. It is useful in cases where the capability of model is unknown and not accurate, but in cases we need to approximate past state of system along with the estimate of the present and future state of it. Taking the case of Road Sign Detection where the object to be traced is tracked until it arises on the scene, Kalman Filter comes very handy [48]. It is used for prediction of sign position and scale and to reduce the search region for the detector and there are other uses of Kalman Filter too like its extensive use in the field of cricket for ball tracking. This filter enables the road sign classifier to fully exploit the image sequence for the entire period when the sign is visible in the field of view of the camera as shown in Figure 4.8. Kalman Filter File.

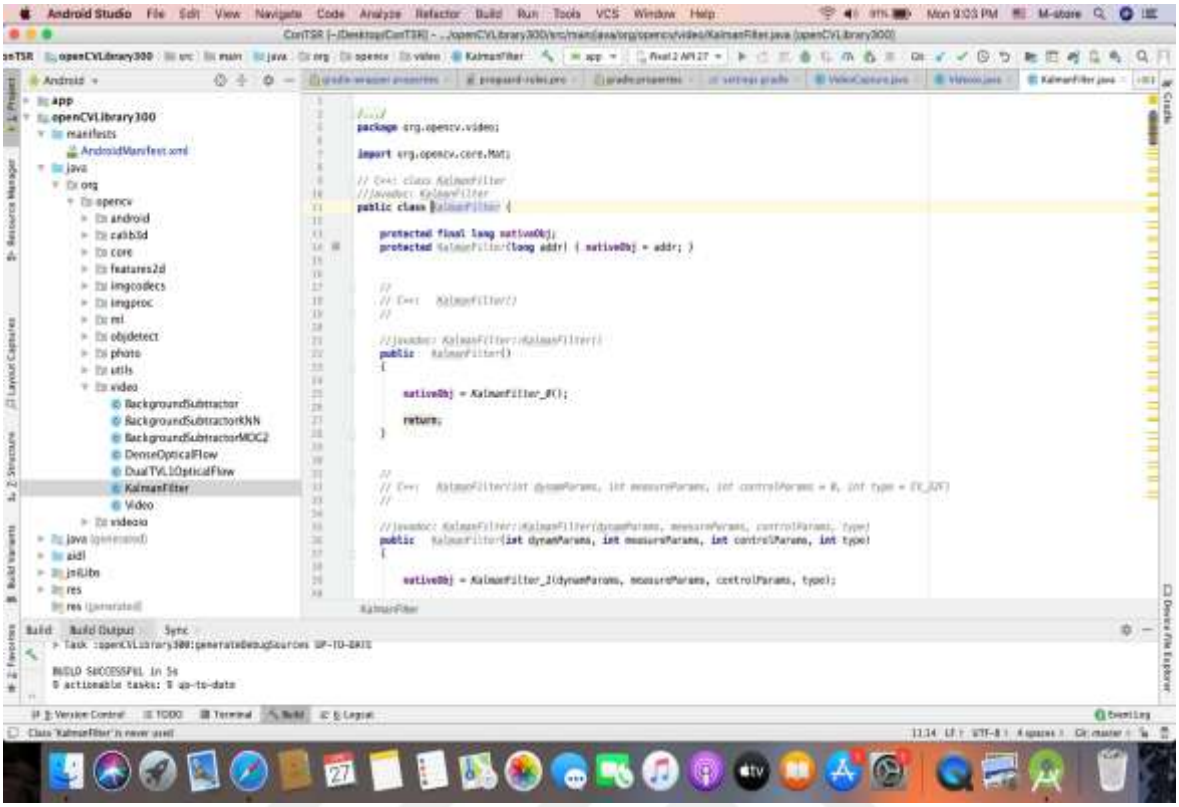

Figure 4.8 - Kalman Filter File

4.8 Running the System

The traffic-sign Recognition system works on Android operating systems on smart phones, it is designed and implemeted to recognize and detect the traffic signs during driving cars. The user opens the application as shows in Figure 4.9 The system Application Icon.

Figure 4.9 - The system Application Icon

When it opened, directly the phone's camera opens and live video feed of the surroundings act as an input to the software. Therefore, whenever a traffic sign appears in front of the camera, it will detect the sign and a pop-up will arrive saying "Traffic Sign Detected". As shown in Figure 4.10. When recognize a traffic sign given us notification alarm of recognizing.

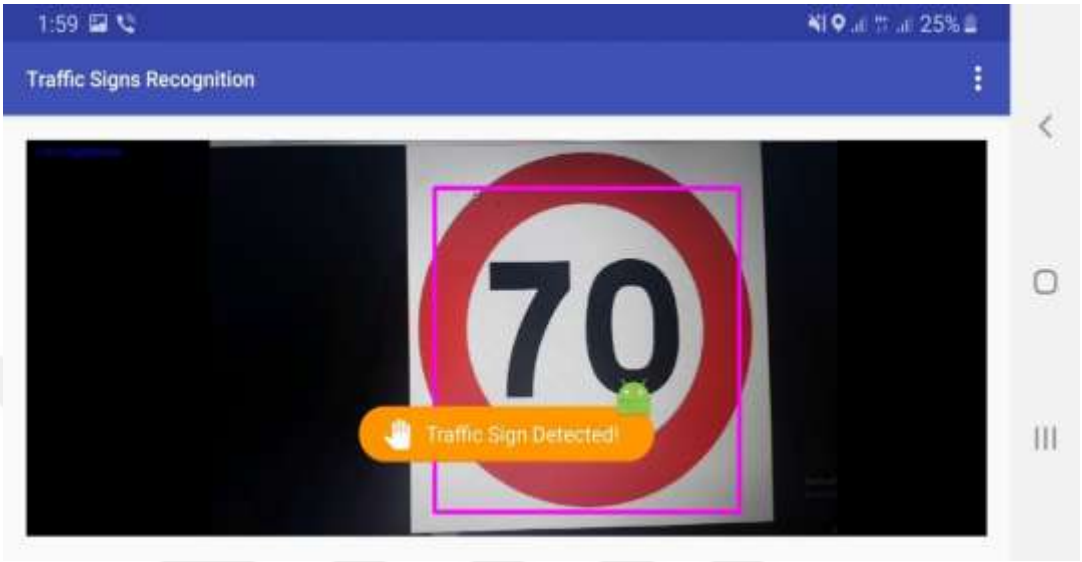

Figure 4.10 - When recognize a traffic sign gives us notification alarm of recognizing

in addition to the sign detection and recognition, the speed of the vehicle is also monitored. Also in Figure 4.11. Showing real time traffic sign recognition during driving.

Figure 4.11 - Showing real time traffic sign recognition during driving

Like when the vehicle moves at a speed of more than 120 KMPH then a pop up comes saying "You are over-speeding"

Lots of test runs as well as experiments were conducted to verify the system. The results were captured which as well are as shown in Figure 4.12. Day & Night Mode for detecting traffic signs:

Figure 4.12 - Day & Night modes for detecting traffic signs

First of all, the front page of the application which shows different options for detection shows us as below options available:

- **Normal:** for day mode traffic signs recognition.
- **HSV:** for night mode traffic sign recognition.
- **HSV Settings:** for adjust HSV functionalities and settings.

For changing the HSV parameters we have to go to the settings then below box opens through the line of each HUE lower max, HUE upper min, Saturation min, Value min parameters are capable of changing to be increases or decreased for night captures and recognized during night mode in order the process to be more smoothly and to be more accurate especially for the drivers whom has problem with night driving and could not see the traffic signs easily at nights. That's why this system helps the drives to be accurate during night driving and the system detect them the traffic signs in the best manner.

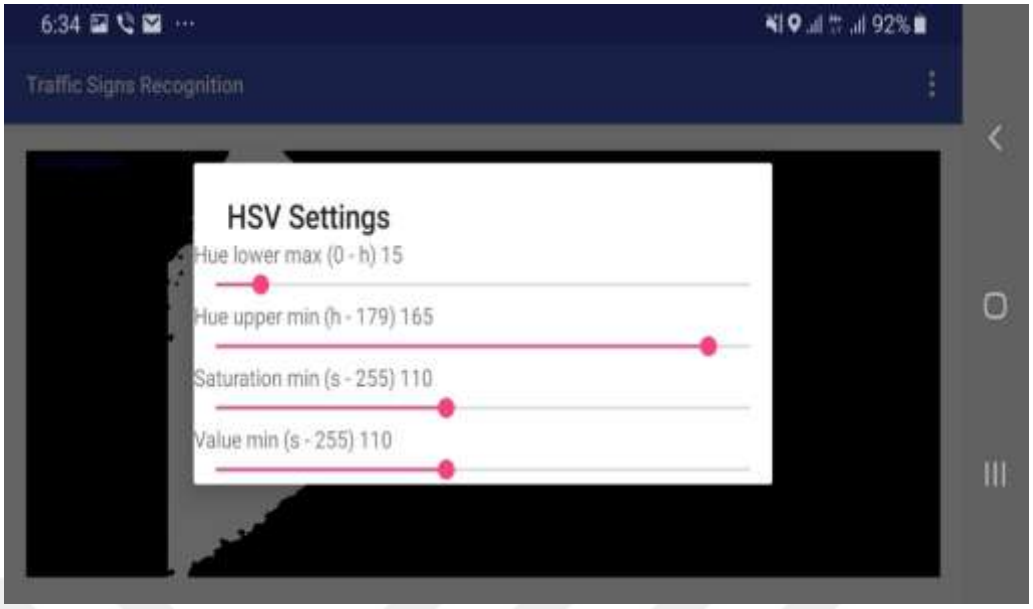

Figure 4.13 - HSV Settings

Therefore, the normal setting indicates the detection during daytime and HSV for nighttime. The HSV Settings option allows user to set the Hue, Saturation, Value according to his/her feasibility as shown in It can be seen in Figure 4.13. HSV Settings.

The capability of the system is not restricted or different between daytime or nighttime regarding recognition, as shown Figure 4.14. Traffic sign recognition during nighttime without setting mode.

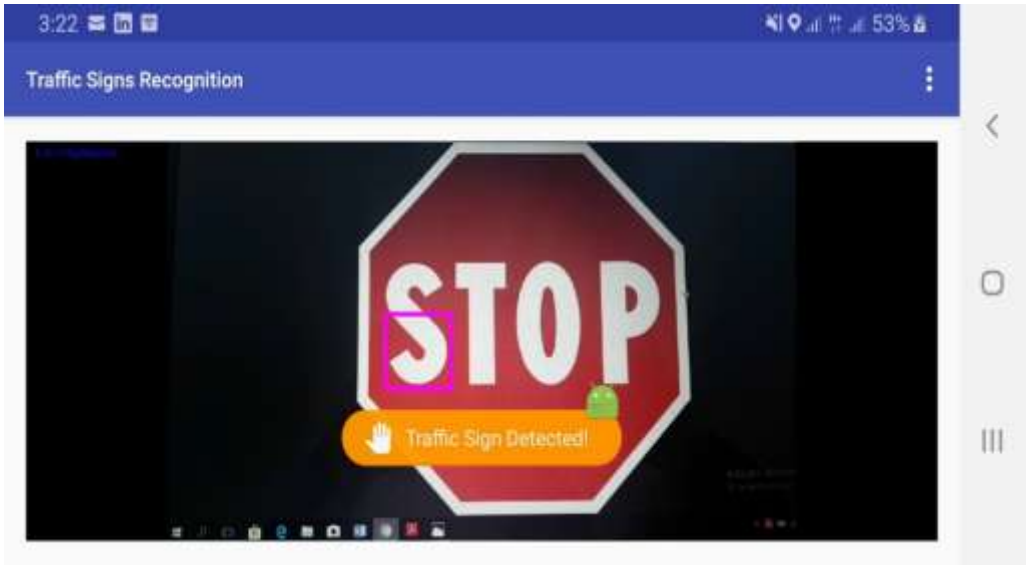

Figure 4.14 - Traffic sign recognition during nighttime without setting mode

If we repeat the same procedure to recognize the same traffic sign even the other traffic signs, we change the mode, the mode to HVS if the brightness or the illumination is not enlightenment, so in below Figure 4.15. Traffic sign recognition during nighttime.

Figure 4.15 - Traffic sign recognition during nighttime

Finally, we have checked the system in real time for several types of traffic signs have been re cognized. The following stills attached from the test runs depict the application's signs detection and recognition for different shapes of traffic signs as shown in Figure 4.16. Recognizing types of traffic signs:

Figure 4.16 - Recognizing types of traffic signs

In below image implies that the vehicle was over speeding which was more than 120kmph then an alarm gives us a notification that the driver is over speed as shown in Figure 4.17. Giving alarm when driving over speed.

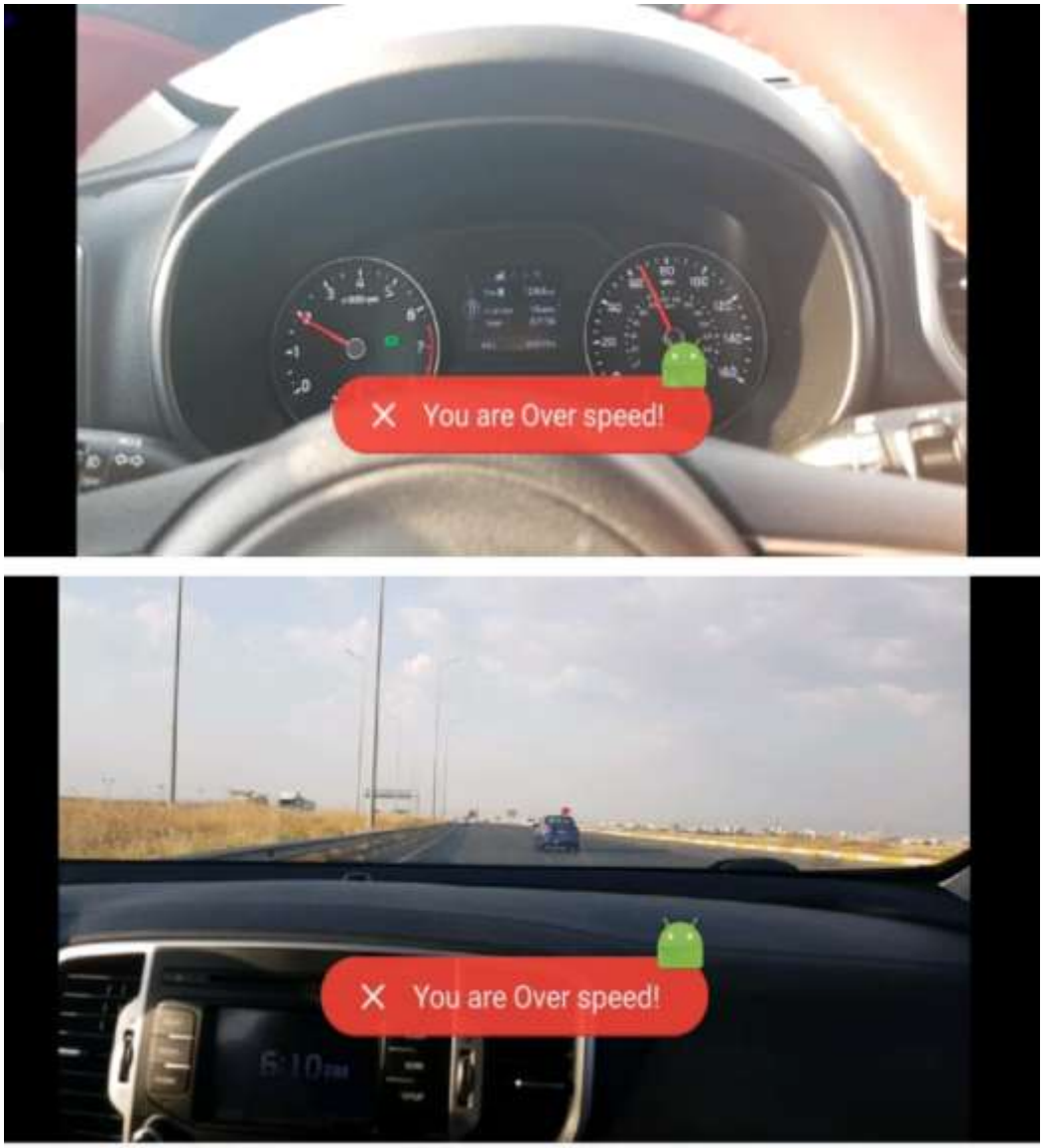

Figure 4.17 - Giving alarm when driving over speed

The above image implies that the vehicle was over speeding which was more than 120kmph then an alarm gives us a notification that the driver drives over speed.

All of the above predictions turn out to be accurate because of the Machine Learning techniques imbibed in OpenCV library. There are a lot of techniques which come up with OpenCV, amongst them Normal Bayes Classifier gave the best results followed by K-Nearest Neighbors, Support Vector Machine, Decision Trees and so on. The least accurate prediction was done by Logistic Regression

5. CONCLUSION AND FUTURE WORK

All of the above-described algorithms OpenCV {RGB, HSV (color segmentation), Region of Interest, Speeded-Up Robust Features}, Convolutional Neural Networks, Normal Bayes Classifier contribute to the research done in development of this software: a highly accurate with fast response and robust Road Signs Detection and Recognition. The application was designed in android studio with the help of OpenCV library programmed by Java language. We have used the Region of Interest (ROI) with Speed Up Robust Feature (SURF) for object recognition. And Red Green Blue model, Hue Saturation Value model for color segmentation during day light and dark respectively. The results achieved from the experimentation cited above prove that our system is a real-time video/image processing software when exposed to a video sequence rather than to a standard approach of traffic sign detection in static images. The results of the test run also prove the convenience of the system. Examining the publicly available data set in real time shows that this system has fast processing speed and robust traffic sign recognition.

As of now, the software doesn't have output in the form of speech. So, in the upcoming phase the detection can be improved by designing a user-friendly application with the addition of voice clarification and handset vibration instead of just giving a pop-up notification that a sign has been detected. The voice feature and vibration will provide the user a pleasant and smooth experience during the usage of the application. Then the traffic signs recognition phase can also be sped up using dimensionality reduction of feature vectors. Moreover, the system will include different kinds of traffic sign shapes (for example, triangular) by applying more advanced methods of feature extraction. Furthermore, with the application of optimization algorithms the efficiency and accuracy of traffic sign detection and classification can also be enhanced. Moreover, the plan is also to train the CNN to consider more traffic sign classes and under possible bad weather conditions.

This application has numerous applications like it will come in great aid to drivers with disabilities. In addition to this, it can help the drivers detect in advance the signs in critical weather conditions like fog, rain, thunderstorm, snowfall and so on. This application's programming interface can be integrated with Maps and Navigation software which in turn can give seamless detection of the traffic signs with speech notification during navigation. Not only this, it can tell at what speed the driver is going on currently and at what speed he/she needs to turn and that too in which direction. This is a very smooth application with precise and accurate predictions because of its design (coded in Java making the application faster compared to applications coded in Python) and the usage of OpenCV with neural networks.

REFERENCES

[1] Jose, A., Thodupunoori, H., & Nair, B. B. (2019). A novel traffic sign recognition system combining Viola–Jones framework and deep learning. In *Soft Computing and Signal Processing* (pp. 507-517). Springer, Singapore.

[2] Farhat, W., Faiedh, H., Souani, C., & Besbes, K. (2019). Real-time embedded system for traffic sign recognition based on ZedBoard. *Journal of Real-Time Image Processing*, *16*(5), 1813-1823.

[3] Jayaprakash, A., & KeziSelvaVijila, C. (2019). Feature selection using ant colony optimization (ACO) and road sign detection and recognition (RSDR) system. *Cognitive Systems Research*, *58*, 123-133.

[4] Wang, L., & Kamata, S. I. (2019, May). Forgery image detection via mask filter banks based CNN. In *Tenth International Conference on Graphics and Image Processing (ICGIP 2018)* (Vol. 11069, p. 110691P). International Society for Optics and Photonics.

[5] Li, G., & Yu, Y. (2016). Deep contrast learning for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 478-487).

[6] Zhong, Y., Ma, A., soon Ong, Y., Zhu, Z., & Zhang, L. (2018). Computational intelligence in optical remote sensing image processing. *Applied Soft Computing*, *64*, 75-93.

[7] Wang, R., Huang, Y., Zhang, Y., Pei, J., Wu, J., & Yang, J. (2019, November). An Inshore Ship Detection Method in SAR Images Based on Contextual Fluctuation Information. In *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)* (pp. 1-5). IEEE.

[8] Li, Ouyang, W., Sheng, L., Zeng, X., & Wang, X. (2019). Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1019-1028).

[9] Belghaouti, O., Handouzi, W., & Tabaa, M. (2020). Improved Traffic Sign Recognition Using Deep ConvNet Architecture. *Procedia Computer Science*, *177*, 468-473.

[10] Jin, Y., Fu, Y., Wang, W., Guo, J., Ren, C., & Xiang, X. (2020). Multi-Feature Fusion and Enhancement Single Shot Detector for Traffic Sign Recognition. *IEEE Access*, *8*, 38931-38940.

[11] Pattarapongsin, P., Neupane, B., Vorawan, J., Sutthikulsombat, H., & Horanont, T. (2020, July). Real-time Drowsiness and Distraction Detection using Computer Vision and Deep Learning. In *Proceedings of the 11th International Conference on Advances in Information Technology* (pp. 1-6).

[12] Epsiba, P., Suresh, G., & Kumaratharan, N. (2020). Edge Detection-Based Depth Analysis Using TD-WHOG Scheme. In *Intelligent Computing in Engineering* (pp. 397-407). Springer, Singapore.

[13] Ratnasari, R., Irawan, B., & Setianingsih, C. (2017, November). Traffic sign recognition application using scale invariant feature transform method and support vector machine based on android. In *2017 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)* (pp. 47-51). IEEE.

[14] Ellahyani, A., El Ansari, M., & El Jaafari, I. (2016). Traffic sign detection and recognition based on random forests. *Applied Soft Computing*, *46*, 805-815.

[15] Zhang, Q., He, N., & Chi, Y. (2018). Circular Traffic Sign Detection in Natural Scenes. *JCP*, *13*(1), 69-76.

[16] Priyanka, D., Dharani, K., Anirudh, C., Akshay, K., Sunil, M. P., & Hariprasad, S. A. (2017, November). Traffic light and sign detection for autonomous land vehicle using Raspberry Pi. In *2017 International Conference on Inventive Computing and Informatics (ICICI)* (pp. 160-164). IEEE.

[17] Shustanov, A., & Yakimov, P. (2017). CNN design for real-time traffic sign recognition. *Procedia engineering*, *201*, 718-725.

[18] Zuo, Z., Yu, K., Zhou, Q., Wang, X., & Li, T. (2017, June). Traffic signs detection based on faster r-cnn. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (pp. 286-288). IEEE.

[19] Romdhane, N. B., Mliki, H., & Hammami, M. (2016, June). An improved traffic signs recognition and tracking method for driver assistance system. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (pp. 1-6). IEEE.

[20] Credi, J. (2016). *Traffic sign classification with deep convolutional neural networks* (Master's thesis).

[21] Sharma, A. (2019). *Traffic Sign Recognition & Detection using Transfer learning* (Doctoral dissertation, Dublin, National College of Ireland).

[22] Omori, Y., & Shima, Y. (2019). HSV Color Space Based Lighting Detection for Brake Lamps of Daytime Vehicle Images. *JCP*, *14*(1), 25-30.

[23] Swathi, M., & Suresh, K. V. (2017, May). Automatic traffic sign detection and recognition in video sequences. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 476-480). IEEE.

[24] Jayaprakash, A., & KeziSelvaVijila, C. (2019). Feature selection using ant colony optimization (ACO) and road sign detection and recognition (RSDR) system. *Cognitive Systems Research*, *58*, 123-133.

[25] Shao, F., Wang, X., Meng, F., Zhu, J., Wang, D., & Dai, J. (2019). Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network. *Sensors*, *19*(10), 2288.

[26] Zhao, M., Jonasson, M., & Ohlsson, N. (2020). Methods and systems for generating and using a road friction estimate based on camera image signal processing,.*U.S. Patent No. 10,706,294*. Washington, DC: U.S. Patent and Trademark Office

[27] Somanal, S. (2019). *Traffic sign detection using high level synthesis* (Doctoral dissertation, California State University, Northridge).

[28] Bi, Z., Yu, L., Gao, H., Zhou, P., & Yao, H. (2020). Improved VGG model-based efficient traffic sign recognition for safe driving in 5G scenarios. *International Journal of Machine Learning and Cybernetics*, 1-12.

[29] Peng, D., Wang, L., Chen, Y., Xia, F., & Qian, Y. (2019). Research on infrared image segmentation and fusion of substation based on modified unit- linking- pulse coupled neural networks and affine speeded up robust feature. *Microwave and Optical Technology Letters*, *61*(6), 1640-1650.

[30] Alam, A., & Jaffery, Z. A. (2020). Indian Traffic Sign Detection and Recognition. *International Journal of Intelligent Transportation Systems Research*, *18*(1), 98-112.

[31] Shabani, S. (2019). Assessment of Driver's Attention to Traffic Signs through Analysis of Gaze and Driving Sequences.

[32] Saleh, S., Khwandah, S., Heller, A., Hardt, W., & Mumtaz, A. (2019). Traffic signs recognition and distance estimation using a monocular camera. In *Conference paper: Actual Problems of System and Software Engineering (APSSE 2019), Moscow, Russia* (pp. 1-13).

[33] Singh, Priya, Milind Sukram Suryawanshi, and Darshana Tak. "Smart Fleet Management System Using IoT, Computer Vision, Cloud Computing and Machine Learning Technologies." *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. IEEE, 2019.

[34] Yin, X., Zhang, J., Wu, X., Huang, J., Xu, Y., & Zhu, L. (2019). An improved lane departure warning algorithm based on fusion of F-Kalman filter and F-TLC. *Multimedia Tools and Applications*, *78*(9), 12203-12222.

[35] Raval, M. S. (2019). Hybrid project-based learning in computer vision. *The International Journal of Electrical Engineering & Education*, 0020720919857632.

[36] Okouneva, G. (2020). *U.S. Patent No. 10,567,748*. Washington, DC: U.S. Patent and Trademark Office.

[37] Bukała, A., Koziarski, M., Cyganek, B., Koç, O. N., & Kara, A. (2019, September). The impact of distortions on the image recognition with histograms of oriented gradients. In *International Conference on Image Processing and Communications* (pp. 166-178). Springer, Cham.

[38] Wang, A., Zhang, W., & Wei, X. (2019). A review on weed detection using ground-based machine vision and image processing techniques. *Computers and electronics in agriculture*, *158*, 226-240.

[39] Shoman, M., Simone, A., & Vignali, V. (2018). Looking behavior to vertical road signs on rural roads. *MOJ Civil Eng*, *4*(2), 75-79.

[40] Xie, X., Shi, F., Niu, J., & Tang, X. (2018, September). Breast ultrasound image classification and segmentation using convolutional neural networks. In *Pacific Rim Conference on Multimedia* (pp. 200-211). Springer, Cham.

[41] Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352-2449.

[42] Baruch, G. (2020). *U.S. Patent No. 10,580,140*. Washington, DC: U.S. Patent and Trademark Office.

[43] Ismaeil, M., Pritamdas, K., Devi, K. J. K., & Goyal, S. (2017, April). Performance analysis of new adaptive decision based median filter on FPGA for impulsive noise filtering. In *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)* (pp. 1-5). IEEE.

[44] Ti, C., Xu, G., Guan, Y., Teng, Y., & Zhang, Y. (2017). Holistic quaternion vector convolution filter for RGB-depth video contour detection. *Journal of Electronic Imaging*, *26*(3), 033010.

[45] Chakraborty, A. (2018, September). Image Processing and Image Pattern Recognition a Programming Tutorial. In *2018 First International Conference on Artificial Intelligence for Industries (AI4I)* (pp. 122-123). IEEE.

[46] Deshpande, A. V., Kannan, R., & Subashini, M. M. (2018, August). Study of Various Image De-Noising Methods Used for the Purpose of Traffic Sign Board Recognition in an Intelligent Advanced Driver Assistance System. In *2018 International Conference on Intelligent and Advanced System (ICIAS)* (pp. 1-6). IEEE.

[47] Mhaske, S., Ambre, N., & Kale, M. V. (2017). Real Time Object Tracking Using Kalman Filter. *Advances in Computational Sciences and Technology*, *10*(6), 1551-1556.

[48] Sung, K. (2017). *A Real-time Motion Detection with Differential Images and Tracking with Mean-Shift and Kalman Filter* (Doctoral dissertation).

[49] Hu, W., Zhuo, Q., Zhang, C., & Li, J. (2017). Fast branch convolutional neural network for traffic sign recognition. *IEEE Intelligent Transportation Systems Magazine*, *9*(3), 114-126.

CURRICULUM VITAE

Botan Hamza Hasan

PERSONAL INFORMATIONS

Birth of Place : Erbil - Iraq
Birth of Date : 1985 - 03- 21
Nationality : Iraqi
Address : Erbil – Roshnbiri Street
E-mail : botan.hamza@Gmail.Com
Languages : Turkish (B1), English (B2)

RESEARCHER INFORMATION

Student Orcid ID : 0000-0003-1487-5847
Advisor Orcid ID : 0000-0002-1730-8111

EDUCATION

Master Degree : "Traffic Sign Recognition"
Firat University, Faculty Graduate School, Department of Software Engineering, Year 2020
Supervisor: Assist Prof Dr. Mehmet KAYA

Bachelor : KOYA University, Engineering Faculty, Department of Software, Year 2010

High School : Ahmadi Khani, Erbil City, Year 2006

RESEARCH EXPERIENCES

- ✓ Computer Programming languages available (JAVA, MATLAB, Android Studio, VB. NET, PHP)
- ✓ Computer programs you can use (CISCO Routers, CISCO Packet Tracer Windows Servers, MSQL, Oracle)

WORK EXPERIENCE

May 2020 up to PRESENT
Supply Chain System Support Expert
Location: Erbil- Iraq
Company: Korek Telecom

ACADEMIC ACTIVITIES

Paper:

1. **Traffic Sign Recognition by OpenCV and Android Studio**

[1] Botan H. Hassan, Mehmet KAYA & Pavel Younus Abdullah, Traffic Sign Recognition by OpenCV and Android Studio.

2. **Software Engineers Role in Protecting HR Data**

[2] Hasan, B., Varol, N., Abdlrazaq, A., Azeez, S., & Varol, A. HR Verilerinin Korunmasında Yazılım Mühendislerinin Rolü Software Engineers Role in Protecting HR Data.