VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama",Belagavi – 590014, KARNATAKA, INDIA

Project Report

On

"ON ROAD VISIBILITY IMPROVEMENT USING THERMAL CAMERA AND DETECTION OF OBJECT AND TRACKING(VIDOT)"

Submitted in Partially fulfillment of the requirement for the award of degree

Of

Bachelor of Engineering

In

Computer Science & Engineering

Of Visvesvaraya Technological University, Belgaum.

Submitted by:

A R SREE HARISH: (1AM18CS001)

AKSHAY KUMAR H S: (1AM18CS013)

AMAN HARIS: (1AM18CS016)

DEVRAJ R: (1AM18CS049)

Under the Guidance of

Dr. DODDEGOWDA B J

Associate Professor, Dept. of CSE

Department of Computer Science and Engineering

AMC Engineering College

(NBA Accredited, Approved by AICTE, New Delhi & Affiliated to VTU, Belagavi)

18th K.M, Bannerghatta Main Road, Bangalore-560 083

2021-2022

AMC ENGINEERING COLLEGE

(NBA ACCREDITED, APPROVED BY AICTE, NEW DELHI & AFFILIATED TO VTU, BELAGAVI)

BENGALURU– 560083

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Project work entitled "ON ROAD VISIBILITY IMPROVEMENT USING THERMAL CAMERA AND DETECTION OF OBJECT AND TRACKING(VIDOT)" carried out by bonafide students A R SREE HARISH(1AM18CS001), AKSHAY KUMAR H S(1AM18CS013), AMAN HARIS(1AM18CS016) and DEVRAJ R(1AM18CS049) of AMC Engineering College in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum during the year 2021-2022. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said Bachelor of Engineering degree.

Project Guide:	HOD:	Principal:
Dr. Doddegowda B J	Dr. Nagaraja R	Dr. Girisha C
Associate Professor	Professor & Head	AMCEC
Dept. Of CSE, AMCEC	Dept. Of CSE, AMCEC	

External Viva

Examiners:	Signature with Date:
1. _____	1. _____
2. _____	2. _____

DECLARATION

We the undersigned students of 8th semester Department of Computer Science & Engineering, AMC Engineering College, declare that our project work entitled "ON ROAD VISIBILITY IMPROVEMENT USING THERMAL CAMERA AND DETECTION OF OBJECT AND TRACKING(VIDOT)" is a bonafide work of ours. Our project is neither a copy nor by means a modification of any other engineering project.

We also declare that this project was not entitled for submission to any other university in the past and shall remain the only submission made and will not be submitted by us to any other university in the future.

Name	USN	Signature
A R Sree Harish	1AM18CS001	_____
Akshay Kumar H S	1AM18CS013	_____
Aman Haris	1AM18CS016	_____
Devraj R	1AM18CS049	_____

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project report would be complete only with mention of people who made it possible, whose support rewarded our effort with success.

First and foremost, I would like to thank the almighty.

I have a great pleasure in expressing my deep sense of gratitude to founder Chairman Dr. K.R. Paramahamsa for having provided me with a great infrastructure and well-furnished labs for successful completion of my seminar. I express my sincere thanks and gratitude to our Principal Dr. Girisha C for providing me all the necessary facilities for successful completion of my seminar.

I would like to extend my special thanks to Dr. R Nagaraja HOD, Department of CSE, for his support and encouragement and suggestions given to me in the course of my seminar work. We greatly express our gratitude to our guide Mrs. Snigdha Kesh, Assistant Professor, Dept., of Computer Science & Engineering, for taking a keen interest in the progress of the project suggestions given to us.

I am grateful to my guide Dr. Doddegowda B J, Associate Prof., Department of CSE, AMC Engineering College, Bengaluru for his unfailing and constant motivational & timely help, encouragement and suggestion, given to me in the course of my seminar work

We also would like to acknowledge all the teaching and non-teaching staff of the department for their kind cooperation during the project. In the end, we sincerely thank our beloved family members and friends for their valuable suggestions, encouragement and unwavering support.

Name	USN
A R SREE HARISH	1AM18CS001
AKSHAY KUMAR H S	1AM18CS013
AMAN HARIS	1AM18CS016
DEVRAJ R	1AM18CS049

ABSTRACT

Every year the lives of approximately 1.3 million people die as a result of a road traffic crash. Road traffic injuries cause considerable economic losses to individuals, their families, and to nations as a whole. Road traffic crashes cost most countries 3% of their gross domestic product. Road traffic injuries constitute a major public health and development crisis, and are predicted to increase if road safety is not addressed adequately by Member States. The World Health Organization (WHO) has been concerned with this issue for over four decades. Road traffic injuries are a major but neglected global public health problem, requiring concerted efforts for effective and sustainable prevention. To see and be seen is a fundamental prerequisite for the safety of all road users. Detailed studies in Australia, Germany and Japan have shown that visual errors play a very important role in the cause of crashes.

In low-income and middle-income countries, the phenomenon of pedestrians and vehicles not being properly visible is frequently a serious problem. In these places, there are fewer roads with adequate illumination and some may not be lit at all. In addition, it is more common for large numbers of bicycles and other vehicles to have no lights and for road space to be shared by fast-moving and slow-moving road users. Traffic collisions in India are a major source of deaths, injuries and property damage every year. India has only 1% of the world's vehicles but 11% of the global deaths from road accidents occur in India, according to a report by the World Bank released earlier this year. About 450,000 accidents take place in India annually, of which 150,000 people die. “India has the highest number of casualties in road accidents,” said the report. “There are 53 road accidents in the country every hour and one death every four minutes.” During the calendar year 2020, road crashes in India claimed about 1.3 lakh lives and caused injuries to more than 3.4 lakh people. Around 40% of these accidents occur due to low visibility.

To avoid these accidents and to help save the lives of both driver and passenger, an automated drivers assistance system (ADAS) is proposed in this system. The proposed ADAS can detect objects on the road irrespective of the weather and lighting conditions and displays an accurate information of the identity and location of the detected object on the road.

CONTENTS

List of Figures

INTRODUCTION

CHAPTER 1

INTRODUCTION

Object detection is a computer vision technique for locating instances of objects in images or videos. Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in (or how they move through) a given scene. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results. When humans look at images or videos, we can recognize and locate objects of interest within a matter of moments. The goal of object detection is to replicate this intelligence using a computer. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

Object tracking is an important task in computer vision. Object trackers are an integral part of many computer vision applications that process the video stream of cameras. Object tracking is an application of deep learning where the program takes an initial set of object detections and develops a unique identification for each of the initial detections and then tracks the detected objects as they move around frames in a video. In other words, object tracking is the task of automatically identifying objects in a video and interpreting them as a set of trajectories with high accuracy. Often, there's an indication around the object being tracked, for example, a surrounding square that follows the object, showing the user where the object is on the screen. Video tracking is an application of object tracking where moving objects are located within video information. Hence, video tracking systems are able to process live, real-time footage and also recorded video files.

A thermal camera is a device that creates an image using infrared (IR) radiation, similar to a normal camera that forms an image using visible light. Instead of the 400–700 nanometre (nm) range of the visible light camera, infrared cameras are sensitive to wavelengths from about 1,000 nm (1 micrometre or μm) to about 14,000 nm (14 μm). Thermal imaging cameras are installed in some luxury cars to aid the driver (automotive night vision). The potential uses for thermal cameras are nearly limitless. Originally developed for surveillance and military operations, thermal cameras are now widely used for building inspections, firefighting, autonomous vehicles and automatic braking, skin temperature screening, industrial inspections, scientific research, and much more.

The above said technologies are being actively used in autonomous vehicles for advanced driver assistance systems. So then why not use them to improve the visibility of a rider/driver while riding/driving during bad weather and lighting situations? Weather condition affects road surface condition and the visibility of the motorist, thereby increasing the chances of mishaps. Adverse weather conditions such as heavy rain, thick fog and hail storms make driving riskier as visibility reduces and road surface gets slippery. This can help prevent a lot of accidents caused due to those reasons. This will also improve riders'/drivers' cognitive sense and help them ride/drive safely. Our proposed idea helps every rider/driver using different types of vehicles to ride/drive safely during bad weather and lighting conditions.

1.1 Aim and Objective

The aim is to improve the way riders/drivers ride/drive during bad weather conditions(fog and rain) and bad lighting conditions(night). The objective of this project is to create an advanced driver assistance system that will help riders/drivers ride/drive safely using the latest computer technologies.

1.2 About the System

In developing countries such as India, where roadways are the major transportation network, also have poorly built infrastructure to support the same. This causes a lot of accidents every year. Even with advancements in technology, there is no advanced specific system to help tackle issues such as bad weather conditions and lighting conditions while driving. In order to help solve all these issues, advanced driver assistance software is programmed using object detection and object tracking algorithms. The software takes thermal imagery as input and the thermal camera can capture the feed of the road irrespective of weather and lighting conditions, unlike the RGB cameras whose video capture gets affected by bad weather and lighting conditions. The software detects and tracks the objects on the road and displays it on the HUD of the vehicle, helping the driver understand what's on the road, which improves the cognitive sense of the driver and helps to prevent avoidable mistakes.

1.3 Existing System

There is currently no advanced system to combat bad weather and lighting conditions. There are a few basic safety measures employed, which are listed below.

- A windshield wiper is a device used to remove rain, snow, ice, washer fluid, water, or debris from a vehicle's front window.
- A fog lamp on the front of a vehicle that is used to help the driver see better in fog.
- A defogger, demister, or defroster to clear condensation and thaw frost from the windshield, backglass, or side windows of a motor vehicle.
- A rear-view mirror and side-view mirror designed to allow the driver to see the rearward and sideward of the vehicle.
- Headlights to view the road during nighttime and low light.

Other than headlights, all other systems are for 4-wheelers and not for 2 and 3-wheelers.

1.4 Disadvantages

- Windshield wipers can get damaged very easily and don't clean the front window neatly.
- Fog lamps only illuminate the ground immediately in front of your vehicle. They provide an inadequate spread of light, aren't aimed properly, and don't improve the driver's vision.
- Defogger, demister, and defroster can be expensive and don't provide a complete solution to the problem.
- Headlights don't provide proper illumination of the road and objects.

1.5 Problem Statement

Every year the lives of approximately 1.3 million people are cut short as a result of a road traffic crash. Adverse weather conditions such as heavy rain, thick fog and hail storms make driving riskier as visibility reduces and road surface gets slippery. In India alone, 0.13 million people die due to road accidents caused by bad weather and lighting conditions.

This can be avoided by integrating the latest computer technologies into the vehicle and we have the solution is by developing an automated driver's assistance system called "VIDOT - on-road Visibility Improvement using a thermal camera and Object Detection and Tracking"

1.6 Proposed System

- To solve the problem, we have built an automated drivers assistance system using thermal camera, object detection algorithm and object tracking algorithm.
- The system captures the live feed through the camera and displays the processed video on a HUD of the vehicle.
- The thermal camera captures a live infrared feed of the road and passes it to the Yolo v4 algorithm for object detection. Yolo v4 identifies and labels the detected object and passes it to the DeepSort algorithm for object tracking. DeepSort gives the object an ID and tracks it during its entire time in the frame.
- This output is then displayed on a HUD using OpenCV library. Corresponding symbols are used to indicate the identity and location of the detected object.
- The system also displays current weather conditions, temperature, and time. Voice commands are used to start and end the software.

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

A literature survey or a literature review in a project report shows the various analyses and research made in the field of interest and the results already published, taking into account the various parameters of the project and the extent of the project.

Literature survey describes about the existing work on the given project. It deals with the problem associated with the existing system and also gives user a clear knowledge on how to deal with the existing problems and how to provide solution to the existing problems.

2.1 Objectives of Literature Survey

- Concentrate on your own field of expertise– Even if another field uses the same words, they usually mean completely
- It improves the quality of the literature survey to exclude sidetracks.
- Learning the definitions of the concepts.
- Access to latest approaches, methods and theories.
- Discovering research topics based on the existing research.

[1] You Only Learn One Representation: Unified Network for Multiple Tasks

Authors: Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao

Institute of Information Science, Academia Sinica, Taiwan2Elan Microelectronics Corporation, Taiwan

People "understand" the world via vision, hearing, tactile, and also the past experience. Human experience can be learned through normal learning (we call it explicit knowledge), or subconsciously (we call it implicit knowledge). These experiences learned through normal learning or subconsciously will be encoded and stored in the brain. Using these abundant experience as a huge database, human beings can effectively process data, even they were unseen beforehand. In this paper, we propose a unified network to encode implicit knowledge and explicit knowledge together, just like the human brain can learn knowledge from normal learning as well as subconsciousness learning. The unified network can generate a unified representation to simultaneously serve various tasks. We can perform kernel space alignment, prediction refinement,

and multi-task learning in a convolutional neural network. The results demonstrate that when implicit knowledge is introduced into the neural network, it benefits the performance of all tasks. We further analyze the implicit representation learnt from the proposed unified network, and it shows great capability on catching the physical meaning of different tasks.

[2] Simple Online and Realtime Tracking with a Deep Association Metric

Authors: Nicolai Wojke, Alex Bewley, Dietrich Paulus

University of Koblenz-Landau, Queensland University of Technology

Simple Online and Realtime Tracking (SORT) is a pragmatic approach to multiple object tracking with a focus on simple, effective algorithms. In this paper, we integrate appearance information to improve the performance of SORT. Due to this extension we are able to track objects through longer periods of occlusions, effectively reducing the number of identity switches. In spirit of the original framework we place much of the computational complexity into an offline pre-training stage where we learn a deep association metric on a largescale person re-identification dataset. During online application, we establish measurement-to-track associations using nearest neighbor queries in visual appearance space. Experimental evaluation shows that our extensions reduce the number of identity switches by 45%, achieving overall competitive performance at high frame rates.

[3] Thermal Object Detection in Difficult Weather Conditions Using YOLO

Authors: Mate Kristo, Marina Ivasic-Kos, Miran Pobar

Department of Informatics University of Rijeka, Rijeka, Croatia

Global terrorist threats and illegal migration have intensified concerns for the security of citizens, and every effort is made to exploit all available technological advances to prevent adverse events and protect people and their property. Due to the ability to use at night and in weather conditions where RGB cameras do not perform well, thermal cameras have become an important component of sophisticated video surveillance systems. In this paper, we investigate the task of automatic person detection in thermal images using convolutional neural network models originally intended for detection in RGB images. We compare the performance of the standard state-of-the-art object detectors such as Faster R-CNN, SSD, Cascade RCNN, and YOLOv3, that were retrained on a dataset of thermal images extracted from videos that simulate illegal movements around the border and in protected areas. Videos are recorded at night in clear weather, rain, and in the fog, at different ranges, and with different movement types. YOLOv3 was significantly faster than other detectors while achieving performance comparable with the best, so it was used in further

experiments. We experimented with different training dataset settings in order to determine the minimum number of images needed to achieve good detection results on test datasets. We achieved excellent detection results with respect to average accuracy for all test scenarios although a modest set of thermal images was used for training. We test our trained model on different well known and widely used thermal imaging datasets as well. In addition, we present the results of the recognition of humans and animals in thermal images, which is particularly important in the case of sneaking around objects and illegal border crossings. Also, we present our original thermal dataset used for experimentation that contains surveillance videos recorded at different weather and shooting conditions.

REQUIREMENTS SPECIFICATION

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 Hardware Requirements

- Processor: Intel i5 8th Gen (min)
- Speed: 2.4 GHz (min)
- RAM: 8 GB (min)
- Hard Disk: 50 GB (min)

3.2 Software Requirements

- Operating System: Windows 8/10
- Programming Language: Python
- IDE: Spyder IDE

3.3 Functional Requirements

- It detects the various objects on road.
- It displays accurate current weather conditions.
- Thermal Camera is used to capture on road video.
- Yolo v4 is used to detect the object and DeepSort is used to track the object.

3.4 Non-Functional Requirements

- **Usability**
  The client acknowledge be typical nearly the buyer interfaces and committed to ask for ambush pressure in relocating to a unique framework with another condition.
- **Reliability**
  The progressions made by the Programmer ought to be obvious both to the Project pioneer and in addition the Test design.
- **Security**
  Counting bug following the framework must give important security and must secure the entire procedure from slamming.
- **Performance**
  The framework will be facilitated on a solitary web server with a solitary database server out of sight, consequently execution turns into a noteworthy concern.
- **Portability**
  This is required when the web server, which is facilitating the framework stalls out because of a few issues, which requires their framework to be taken to another framework.

SYSTEM DESIGN AND ARCHITECTURE

CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

4.1 Design

4.1.1 System Architecture

The architectural configuration procedure is concerned with building up a fundamental basic system for a framework. It includes recognizing the real parts of the framework and interchanges between these segments. The beginning configuration procedure of recognizing these subsystems and building up a structure for subsystem control and correspondence is called construction modeling outline and the yield of this outlined procedure is a portrayal of the product structural planning. The proposed architecture for this system is given below.

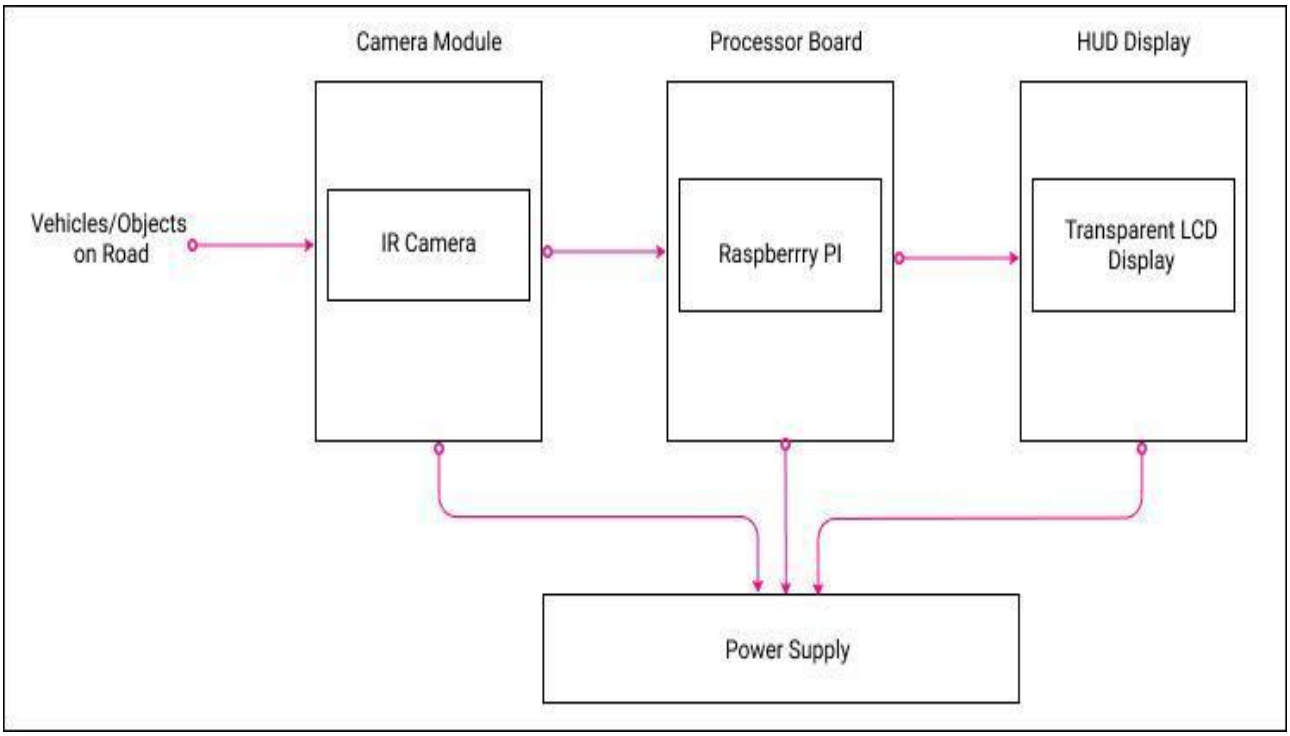

Fig 4.1.1:- System Architecture

4.1.2 Software Architecture

Two algorithms are used to build the backbone software of this system.

YOLOv4 is a two-stage detector with several components to it. The first stage detector consists of input, backbone, neck, and dense prediction. This stage is connected to a sparse prediction block. Input

takes images, patches, or pyramids. Backbone consists of VGG16, ResNet-50, SpineNet, EfficientNet-B0-B7, CSPResNext50, CSPDarknet53. The neck consists of additional blocks: SSP, ASPP, RFB, SAM, and path-aggregation blocks: FPN, PAN, NAS-FPN, Fully-connected FPN, BiFPN, ASFF, and SFAM. Dense prediction consists of either RPN, SSD, YOLO, RentinaNet (anchor-based) or CornerNet, CenterNet, or MatrixNet (anchor-free). Sparse prediction consists of either Faster R-CNN, R-FCN, Mask R-CNN (anchor-based), or RepPoints (anchor-free). You can see that YOLOv4 can be implemented in any combination of input, backbone, neck, and head.

DeepSort is a machine learning model for tracking people, and assigning IDs to each person. Using the bounding boxes detected by YOLO v4, we can assign an ID and track a person by mapping bounding boxes of similar size and similar motion in the previous and following frames. In DeepSort, the process is as follows:

- Compute bounding boxes using YOLO v4 (detections)
- Use Sort (Kalman filter)and ReID (identification model) to link bounding boxes and tracks
- If no link can be made, a new ID is assigned and it is newly added to tracks.

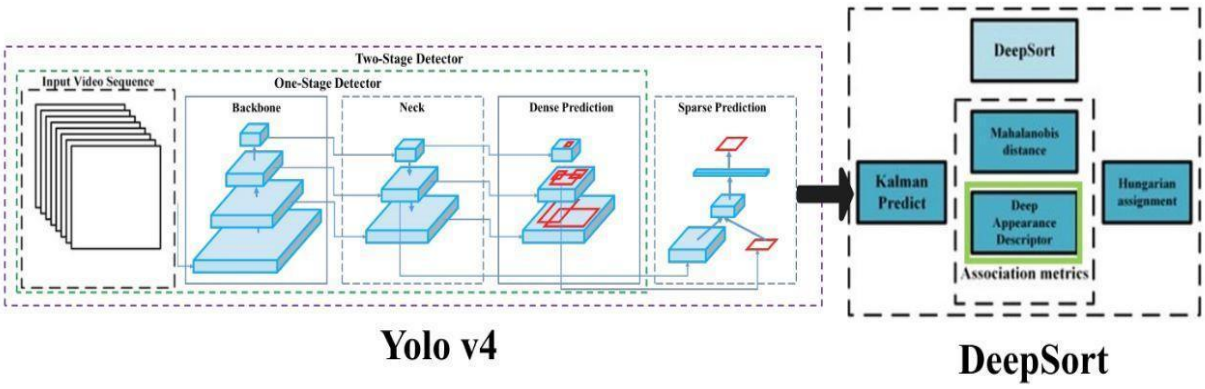

Fig 4.1.2 :- Software Architecture

4.1.2 Flowchart

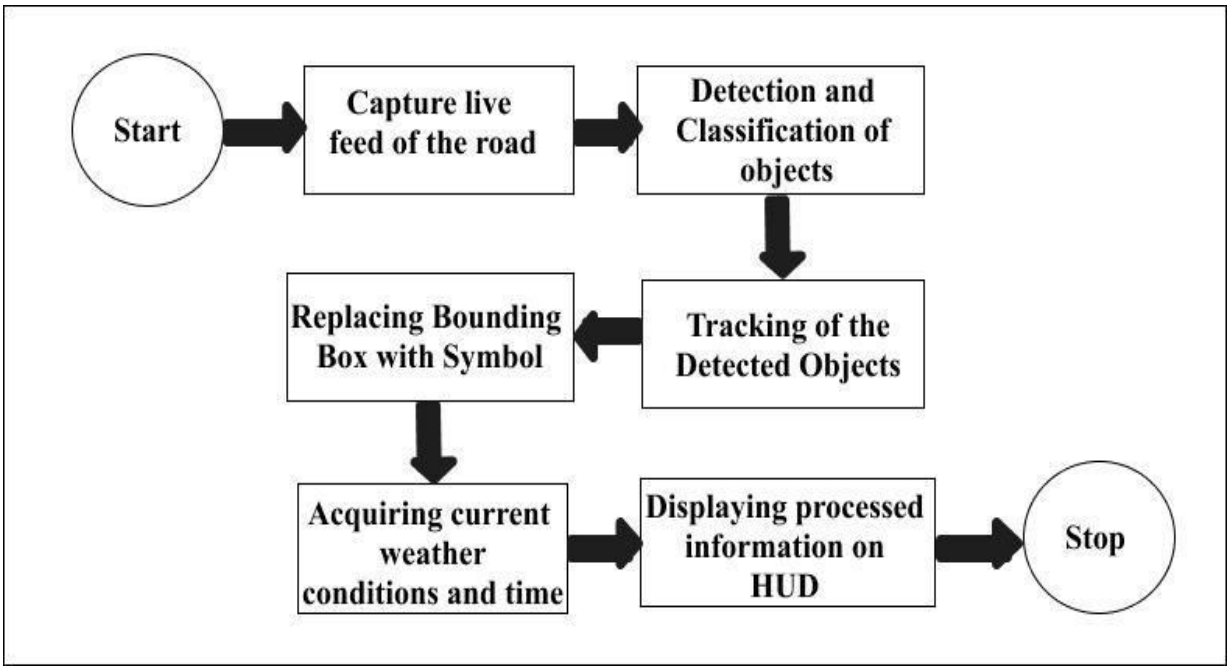

Fig 4.1.2:- System Flow Chart

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.

The application will start with a start command, this application is integrated with voice command. Once the product or device is turned on, with voice input like, "START" the application will get started. Once the application is started, the camera starts to record the video in live to get the feed from road. This feed is given to detection and classification of object algorithm for detection and classification.

Then the objects are tracked using a algorithm names DeepSort algorithm, once the objects get detected, classified and tracked, we will replace the bounding box with a symbol in order to make the interface hassle free for the drivers.

In addition, there is a well designed interface, that shows the weather condition and time at the top corners of the display. Where we are extracting this information from the public API using python programming.

Once all the data is gathered, the output is displayed in the HUD display.

With the "STOP" voice command the application will get terminated, else it can be done manually.

4.2 Algorithms

4.2.1 Yolo v4

The original YOLO (You Only Look Once) was written by Joseph Redmon in a custom framework called Darknet. Darknet is a very flexible research framework written in low-level languages and has produced a series of the best real-time object detectors in computer vision: YOLO, YOLOv2, YOLOv3, and now, YOLOv4. YOLOv4 is a two-stage detector with several components to it, which are given below.

Backbone is the deep learning architecture that basically acts as a feature extractor. All of the backbone models are basically classification models. There are three more models that we can use in the backbone other than the models mentioned above namely SqueezeNet, MobileNet, and ShuffleNet, but all of them are meant for CPU training only.

Neck is a subset of the bag of specials, it basically collects feature maps from different stages of the backbone. In simple terms, it's a feature aggregator.

Head is also known as the object detector, it basically finds the region where the object might be present but doesn't tell which object is present in that region. We have two-stage detectors and one stage-detectors which are further subdivided into anchor-based and anchor-free detectors.

Bag of freebies is those methods that only change the training strategy or only increase the training cost (nothing to do with inference). You can see from the above diagram that there are an insane amount of things you can try, but we will discuss only the most important ones.

Bag of specials is those Plugin modules and post-processing methods that only increase the inference cost by a small amount but can significantly improve the accuracy.

As we have seen that there are a lot of possibilities in choosing the architecture. So, what exactly do we look for while choosing the architecture. Selection criteria are based on the optimal balance between input network resolution (input image size), number of convolution layers, number of parameters, and number of output layers (filters). Also, we have an additional block for increasing the receptive field (bag of specials) and the best method from different backbone levels to different detector levels (Necks). The final architecture used for this project is CSPDarkNet53 (Head) => SSP + PANet (Neck)=> YOLOv3 (head)

4.2.2 DeepSort

Traditionally, tracking has used an algorithm called Sort (Simple Online and Realtime Tracking), which uses the Kalman filter. Using the bounding boxes detected by YOLO v4, we can assign an ID and track a person by mapping bounding boxes of similar size and similar motion in the previous and following frames. However, Sort presents the limitation that if a person hid behind an object and then reappeared, it

is assigned a different ID. DeepSort solves this problem by using an AI model that compares similarities between people, thus reducing the issue of switching people's identities.

In DeepSort, the process is as follows: Compute bounding boxes using YOLO v4 (detections), Use Sort (Kalman filter)and ReID (identification model) to link bounding boxes and tracks

If no link can be made, a new ID is assigned and it is newly added to tracks. What is referred to as "detections" is the list of people in one frame, and "tracks" is the list of people currently being tracked. Each item of tracks is assigned an ID, and by assigning a bounding box to each one of those items, you can assign an ID to the person. ReID is mainly used when linking bounding boxes and tracks. The distance between the feature vectors computed by ReID from the person image of the current tracking target (tracks) and the feature vectors also calculated by ReID from the person image cut out by the bounding box (detections) in YOLO v4, is used to link bounding boxes and tracks. Simply put, the object with the smallest distance is considered to be the same person and assigned a track ID. To calculate the vector distance, feature vectors for the last 100 frames for each track are used. At this time, the coordinate information of the track is not taken into account. The cost function is defined as Sort distance * λ + ReID distance, but in the paper, $\lambda = 0$ turned out to empirically give good results, so the coordinate information is not taken into account. If the position in the current frame, which is assumed based on the Sort past tracking information, is too far apart, the ID will not be assigned. When a bounding box is left without any ID, Sort is used to assign one. If the bounding box is "lost" for 70 frames, it will be removed from the tracking. The ReID model is trained from 1,100,000 images of 1,261 pedestrians from the large-scale person re-identification dataset.

IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

5.1 Language Used For Implementation

5.1.1 Python

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Python consistently ranks as one of the most popular programming languages. It is used by many organizations and companies. Pixar, Disney, Instagram, and the developers of the Linux Kernel are among many of its high-profile users, which includes many developers of Free and Open source software.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including metaprogramming and metaobjects [magic methods]). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It uses dynamic name resolution (late binding), which binds method and variable names during program execution. Its design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML. Rather than building all of its functionality into its core, Python was designed to be highly extensible via modules. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology.

5.1.2 Features of Python

- Free and Open Source: Python language is freely available on the official website and you can download it from its website. Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

- Object-Oriented Language: One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.
- GUI Programming Support: Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.
- High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.
- Python is an Integrated language: Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc

5.1.3 Advantages of Python

- Python is a high-level programming language that has English-like syntax. This makes it easier to read and understand the code.
- Python is a very productive language. Due to the simplicity of Python, developers can focus on solving the problem.
- Python shows only one error even if the program has multiple errors. This makes debugging easier.
- In many languages like C/C++, you need to change your code to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere.
- Python automatically assigns the data type during execution. The programmer doesn't need to worry about declaring variables and their data types.

5.2 Software Used For Implementation

5.2.1 Anaconda (Python distribution)

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250

packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command-line interface (CLI).

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution that allows users to launch applications and manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS, and Linux. The following applications are available by default in Navigator:

- 1.JupyterLab
- 2.Jupyter Notebook
- 3.QtConsole
- 4.Spyder
- 5.Glue
- 6.Orange
- 7.RStudio
- 8.Visual Studio Code

5.2.2 Spyder IDE

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy, and Cython, as well as other open-source software.It is a free and open-source scientific environment written in Python, for Python, and designed by and for scientists, engineers, and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package. It is released under the MIT license.Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. It is also known as the Scientific Python Development IDE and has a huge set of remarkable features.

Spyder is extensible with first-party and third-party plugins, includes support for interactive tools for data inspection, and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint, and Rope. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE, and Ubuntu. Spyder uses Qt for its GUI and is designed to use either the PyQt or PySide Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either

backend. It has customizable syntax highlighting and availability of breakpoints (debugging and conditional breakpoints). It provides interactive execution which allows you to run line, file, cell, etc, and run configurations for working directory selections, command-line options, current/ dedicated/ external console, etc. It can clear variables automatically (or enter debugging). Navigation through cells, functions, blocks, etc can be achieved through the Outline Explorer. It provides real-time code introspection. There is automatic colon insertion after if, while, etc, and supports all the IPython magic commands. It provides an inline display for graphics produced using Matplotlib and also provides features such as help, file explorer, finds files, etc.

5.2.3 Python Modules

5.2.3.1 Opencv

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage and then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting in 2011, OpenCV features GPU acceleration for real-time operations. OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. OpenCV is used for all sorts of image and video analysis, like facial recognition and detection, license plate reading, photo editing, advanced robotic vision, optical character recognition, and a whole lot more. OpenCV is a great tool for image processing and performing computer vision tasks. The library is equipped with hundreds of useful functions and algorithms, which are all freely available to us. Some of these functions are really common and are used in almost every computer vision task.

5.2.3.2 Numpy

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is open-source software and has many contributors. NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms are written for this version of Python often run much slower than compiled equivalents due to the absence of compiler optimization. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays; using these requires rewriting some code, mostly inner loops, using NumPy. Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations. Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with

multiple channels are simply represented as three-dimensional arrays, indexing, slicing, or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as a universal data structure in OpenCV for images, extracted feature points, filter kernels, and many more vastly simplifies the programming workflow and debugging.

5.2.3.3 Tensorflow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as Javascript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors.Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and developers easily build and deploy ML-powered applications. TensorFlow is a Python-friendly open source library for numerical computation that makes machine learning and developing neural networks faster and easier.

5.2.3.4 Datetime

The datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation. A date in Python is not a data type of its own, but we can import a module named datetime to work with dates as date objects. Python Datetime module comes built into Python, so there is no need to install it externally. Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times, and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

5.2.3.5 Pyowm

PyOWM is a client Python wrapper library for OpenWeatherMap web APIs. OpenWeather is a team of IT experts and data scientists that has been practicing deep weather data science. For each point on the globe, OpenWeather provides historical, current, and forecasted weather data via light-speed APIs. PyOWM allows quick and easy consumption of OWM data from Python applications via a simple object model and in a human-friendly fashion.With PyOWM you can integrate into your code any of the OpenWeatherMap web APIs such as Weather API v2.5 + OneCall API which provides current weather data, weather forecasts, and weather history, Agro API v1.0 which provides soil data and satellite imagery search and download, Air Pollution API

v3.0 which provides data about CO, O3, NO2 and SO2, UV Index API v3.0 which provides data about Ultraviolet exposition, Weather Alerts API v3.0 which allows to set triggers on weather conditions and areas and poll for spawned alerts, Image tiles for several map layers provided by OWM and Geocoding API v1.0 which allows to perform direct/reverse geocoding.

5.2.3.6 os

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system. This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level files and directory, handling sees the shutil module. It is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

5.2.3.7 absl-py

This repository is a collection of Python library code for building Python applications. The code is collected from Google's own Python code base and has been extensively tested and used in production. Features of this module are: Simple application startup, Distributed commandline flags system, Custom logging module with additional features, and Testing utilities.

5.2.3.8 pyttsx3

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. is very easy-to-use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by "sapi5" for windows.

5.2.3.9 Speech_recognition

Recognizing speech requires audio input, and SpeechRecognition makes retrieving this input really easy. Instead of having to build scripts for accessing microphones and processing audio files from scratch, SpeechRecognition will have you up and running in just a few minutes. The SpeechRecognition library acts as a wrapper for several popular speech APIs and is thus extremely flexible. One of these—the Google Web Speech API—supports a default API key that is hard-coded into the SpeechRecognition library. The flexibility and ease of use of the SpeechRecognition package make it an excellent choice for any Python project. It allows: Easy speech recognition from the microphone, Makes it easy to transcribe an audio file, It also lets us save audio data into an audio file, It also shows us recognition results in an easy-to-understand format.

5.3 Observation And Result

The proposed system is fed with a pre-recorded video data captured from a thermal camera the result is shown in the form of a video displayed with accurate object detection and object tracking along with weather details, time, and date.

SYSTEM TESTING

CHAPTER 6

SYSTEM TESTING

Testing of any product comprises of giving the product an arrangement of test information and watching if the product carries on not surprisingly, if the product neglects to carry on obviously, then the conditions under which disappointment happens are noted for investigating and amendment. At last, the framework in general is tried to guarantee that blunders in past countenances are revealed and the venture acts as determined.

6.1 Basics Of Software Testing

6.1.1 Black Box Testing

Black box testing is done to find the following

- Incorrect or missing functions
- Interface errors
- Errors on external database access
- Performance error
- Initialization and termination error

6.1.2 White Box Testing

This allows the tests to

- Check whether all independent paths within a module have been exercised at least once
- Exercise all logical decisions on their false sides
- Execute all loops and their boundaries and within their boundaries
- Exercise the internal data structure to ensure their validity
- Ensure whether all possible validity checks and validity lookups have been provided to validate data entry.

6.1.3 TYPES OF TESTING

Following are the different types of testing

- Unit Testing
- Integration Testing
- System Testing
- Performance Testing
- Validation Testing
- Acceptance Testing

Let us consider each testing and discuss it in detail. Firstly we move to the first testing and give its detailed description.

6.1.4 Unit Testing

Singular parts are tried to guarantee that they work accurately. Every part is tried freely, without another framework segment. This framework was tried with the arrangement of legitimate test information for every module and the outcomes were checked with the normal yield. Unit testing centers around confirmation exertion on the littlest unit of the product outline module. This is otherwise called MODULE TESTING. This testing is done amid stages, every module is observed to work agreeable with respect to the normal yield from the module.

6.1.5 Integration Testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Table 6.1 Integration Testing Test Case

Name of the Test	Integration Testing
Test plan	To check the system works properly when all the modules are integrated.
Test Data	Sample video data

6.2 System Testing

System testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and system testing to be conducted as two distinct phases.

6.2.1 Test strategy and approach

Field testing will be performed manually, and functional tests will be written in detail.

6.2.2 Test objectives

- Video capturing happens properly
- Detection and tracking of objects on road.
- Display accurate information on the display.
- Check the speed and accuracy of the software.

6.3 Performance Testing

The execution testing guarantee that the yield being delivered inside as far as possible and time taken for the framework aggregating, offering reaction to the clients and demand being send to the framework so as to recover the outcomes.

6.4 Validation Testing

The approval testing can be characterized from multiple points of view, however a straightforward definition is that. Approval succeeds when the product capacities in a way that can be sensibly expected by the end client.

6.5 Acceptance Testing

This is the last phase of testing procedure before the framework is acknowledged for operational utilize. The framework is tried inside the information provided from the framework procurer instead of recreated information.

6.6 Test Results

All the test cases discussed above have passed successfully.

CONCLUSION AND FUTURE WORK

CHAPTER 7

CONCLUSION

In this system we have come up with one of the solutions for the safety of drivers and pedestrians. Where we gathered information and literature surveys and came up with a solution that uses the implementation of multiple technologies to integrate with the real world using camera and using this data, with the help of object detection and tracking we display the output.

For the hardware part, we used an LCD display making it transparent, where the output of the video is displayed. The hardware implementation that we have currently done is just a POC of the actual hardware system and to demo of how the HUD display would work.

The application will start with a start command, this application is integrated with voice command. Once the product or device is turned on, with voice input like, "START" the application will get started. Once the application is started, the camera starts to record the video in live to get the feed from road. This feed is given to detection and classification of object algorithm for detection and classification. Then the objects are tracked using a algorithm names DeepSort algorithm, once the objects get detected, classified and tracked, we will replace the bounding box with a symbol in order to make the interface hassle free for the drivers. In addition, there is a well designed interface, that shows the weather condition and time at the top corners of the display. Where we are extracting this information from the public API using python programming. Once all the data is gathered, the output is displayed in the HUD display. With the "STOP" voice command the application will get terminated, else it can be done manually.

FUTURE SCOPE

Our Future Scope is to run the software on a raspberry pi processing board and to display the output using a HUD display. We also plan to improve the accuracy and processing speed of our software. We can also add additional features such as Google Maps to help in navigation.

The project is built is such a way that, it can be used in multiple different types of vehicles. Like during rainy times the lorry drivers stop there vehicles due to lack of proper visibility, this causes a loss of productivity which will directly affect the business.

So if we implement this project in big screens like lorry, bus, car and etc. This may be a game changer.

So our another future plan is to take this idea and build a, robust version of our older project, so that it can be used in every vehicle, from low range to high range vehicles.

APPENDIX

APPENDIX A

REFERENCES

[1] Color-to-Grayscale: Does the Method Matter in Image Recognition? by Christopher Kanan ,Garrison W. Cottrell

[2] Apparent Greyscale: A Simple and Fast Conversion to Perceptually Accurate Images and Video by Kaleigh Smith,Pierre-Edouard Landes,Joëlle Thollot,Karol Myszkowski

[3] Real-time object detection and tracking on a moving camera platform by Cheng-Ming Huang; Yi-Ru Chen; Li-Chen Fu

[4] You Only Learn One Representation: Unified Network for Multiple Tasks by Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao

[5] Real – Time Object detection tracking by Hammad naeem, Jawad ahmad and Muhammad tayyab, HITEC University Taxila cantta, Pakistan.

[6] Youtube.com

[7] GitHub.com

[8]Geeksforgeeks.org

[9]Stackoverflow.com

[10]Medium.com

APPENDIX B

SNAPSHOTS

Fig A.1:- Detecting Humans in thermal camera

Fig A.2:- Detecting Vehicles in thermal camera

Fig A.3:- Detecting Vehicles and Humans in RGB camera

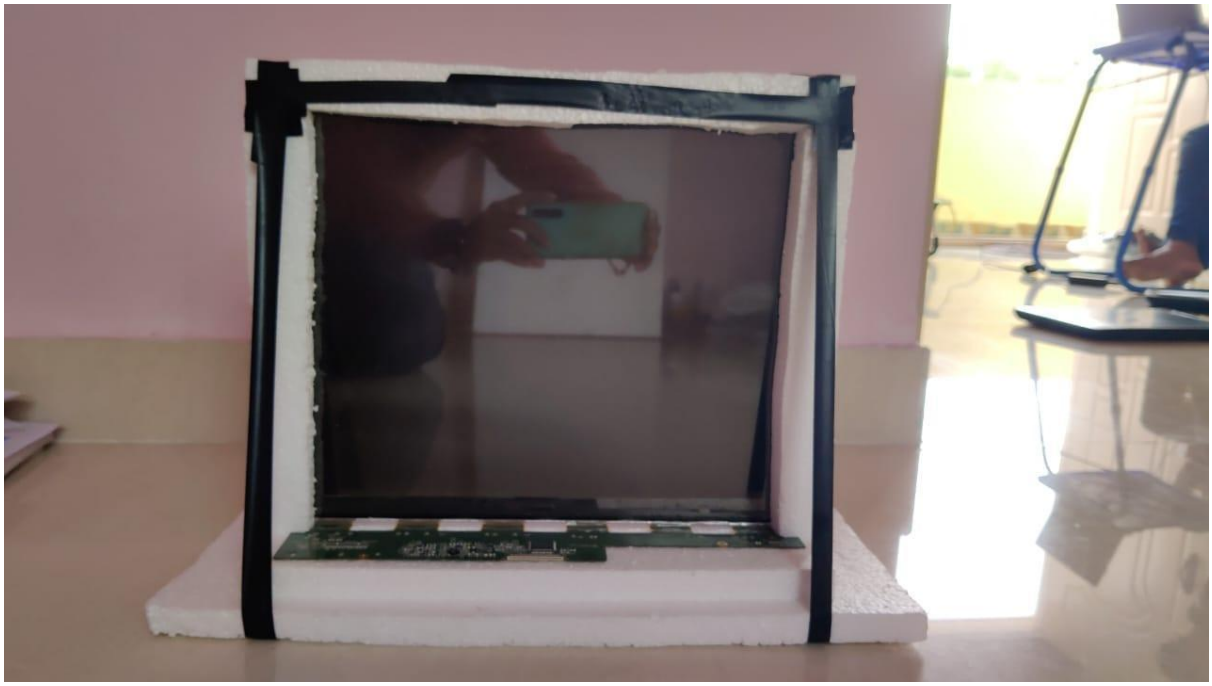

Fig A.4:- Display Prototype

APPENDIX C

SAMPLE CODE

```
import os
# comment out below line to enable tensorflow logging outputs
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
import time
import tensorflow as tf
physical_devices = tf.config.experimental.list_physical_devices('GPU')
if len(physical_devices) > 0:
    tf.config.experimental.set_memory_growth(physical_devices[0], True)
from absl import app, flags, logging
from absl.flags import FLAGS
import core.utils as utils
from core.yolov4 import filter_boxes
from tensorflow.python.saved_model import tag_constants
from core.config import cfg
from PIL import Image
import cv2
import cvzone
import addsymbol
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession
# deep sort imports
from deep_sort import preprocessing, nn_matching
```

```
from deep_sort.detection import Detection
from deep_sort.tracker import Tracker
from tools import generate_detections as gdet
import dateweather
def dot(_argv):
    # Definition of the parameters
    max_cosine_distance = 0.4
    nn_budget = None
    nms_max_overlap = 1.0
    # initialize deep sort
    model_filename = 'model_data/mars-small128.pb'
    encoder = gdet.create_box_encoder(model_filename, batch_size=1)
    # calculate cosine distance metric
    metric = nn_matching.NearestNeighborDistanceMetric("cosine", max_cosine_distance,
nn_budget)
    # initialize tracker
    tracker = Tracker(metric)
    # load configuration for object detector
    config = ConfigProto()
    config.gpu_options.allow_growth = True
    session = InteractiveSession(config=config)
    Model = 'yolov4'
    STRIDES, ANCHORS, NUM_CLASS, XYSCALE = utils.load_config(Model)
    input_size = 416
    video_path = './data/video/test.mp4'
    weights = './checkpoints/yolov4-416'
    saved_model_loaded = tf.saved_model.load(weights, tags=[tag_constants.SERVING])
    infer = saved_model_loaded.signatures['serving_default']
```

```
# begin video capture
try:
    vid = cv2.VideoCapture(int(video_path))
except:
    vid = cv2.VideoCapture(video_path)
out = None
# get video ready to save locally if flag is set
output = True
output_format = 'XVID'
if (output == True):
    # by default VideoCapture returns float instead of int
    width = int(vid.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(vid.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = int(vid.get(cv2.CAP_PROP_FPS))
    codec = cv2.VideoWriter_fourcc(*output_format)
    out = cv2.VideoWriter('./outputs/demo.avi', codec, fps, (width, height))
frame_num = 0
# while video is running
while True:
    return_value, frame = vid.read()
    if return_value:
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image = Image.fromarray(frame)
    else:
        print('Video has ended or failed, try a different video format!')
        break
    frame_num +=1
    print('Frame #: ', frame_num)
```

```
frame_size = frame.shape[:2]
image_data = cv2.resize(frame, (input_size, input_size))
image_data = image_data / 255.
image_data = image_data[np.newaxis, ...].astype(np.float32)
start_time = time.time()
batch_data = tf.constant(image_data)
pred_bbox = infer(batch_data)
for key, value in pred_bbox.items():
    boxes = value[:, :, 0:4]
    pred_conf = value[:, :, 4:]
boxes, scores, classes, valid_detections = tf.image.combined_non_max_suppression(
    boxes=tf.reshape(boxes, (tf.shape(boxes)[0], -1, 1, 4)),
    scores=tf.reshape(
        pred_conf, (tf.shape(pred_conf)[0], -1, tf.shape(pred_conf)[-1])),
    max_output_size_per_class=50,
    max_total_size=50,
    iou_threshold=0.45,
    score_threshold=0.50
)
# convert data to numpy arrays and slice out unused elements
num_objects = valid_detections.numpy()[0]
bboxes = boxes.numpy()[0]
bboxes = bboxes[0:int(num_objects)]
scores = scores.numpy()[0]
scores = scores[0:int(num_objects)]
classes = classes.numpy()[0]
classes = classes[0:int(num_objects)]
```

```
    # format bounding boxes from normalized ymin, xmin, ymax, xmax ---> xmin, ymin, width,
height

    original_h, original_w, _ = frame.shape

    bboxes = utils.format_boxes(bboxes, original_h, original_w)

    # store all predictions in one parameter for simplicity when calling functions

    pred_bbox = [bboxes, scores, classes, num_objects]

    # read in all class names from config

    class_names = utils.read_class_names(cfg.YOLO.CLASSES)

    # by default allow all classes in .names file

    allowed_classes = list(class_names.values())

    # custom allowed classes (uncomment line below to customize tracker for only people)

    allowed_classes = ['person', 'bicycle', 'car', 'motorcycle', 'bus', 'train', 'truck', 'boat', 'traffic light',
'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant']

    # loop through objects and use class index to get class name, allow only classes in
allowed_classes list

    names = []

    deleted_indx = []

    for i in range(num_objects):

        class_indx = int(classes[i])

        class_name = class_names[class_indx]

        if class_name not in allowed_classes:

            deleted_indx.append(i)

        else:

            names.append(class_name)

    names = np.array(names)

    # delete detections that are not in allowed_classes

    bboxes = np.delete(bboxes, deleted_indx, axis=0)

    scores = np.delete(scores, deleted_indx, axis=0)
```

```
        # encode yolo detections and feed to tracker
        features = encoder(frame, bboxes)
        detections = [Detection(bbox, score, class_name, feature) for bbox, score, class_name, feature
in zip(bboxes, scores, names, features)]
        #initialize color map
        cmap = plt.get_cmap('tab20b')
        colors = [cmap(i)[:3] for i in np.linspace(0, 1, 20)]
        # run non-maxima supression
        boxs = np.array([d.tlwh for d in detections])
        scores = np.array([d.confidence for d in detections])
        classes = np.array([d.class_name for d in detections])
        indices = preprocessing.non_max_suppression(boxs, classes, nms_max_overlap, scores)
        detections = [detections[i] for i in indices]
        # Call the tracker
        tracker.predict()
        tracker.update(detections)
        # update tracks
        for track in tracker.tracks:
            if not track.is_confirmed() or track.time_since_update > 1:
                continue
            bbox = track.to_tlbr()
            class_name = track.get_class()
        # draw bbox on screen
            color = colors[int(track.track_id) % len(colors)]
            color = [i * 255 for i in color]
            #cv2.rectangle(frame, (int(bbox[0]), int(bbox[1])), (int(bbox[2]), int(bbox[3])), color, 2)
            #cv2.rectangle(frame, (int(bbox[0]), int(bbox[1]-30)),
(int(bbox[0])+(len(class_name)+len(str(track.track_id)))*17, int(bbox[1])), color, -1)
```

```
            cv2.putText(frame, class_name + "-" + str(track.track_id),(int(bbox[0]), int(bbox[1]-10)),0,
0.75, (255,255,255),2)

            #add symbol

            path = './symbol/'+class_name+'.png'

            symbol = cv2.imread(path, cv2.IMREAD_UNCHANGED)

            w = 75

            h = 75

            size = (w,h)

            symbol = cv2.resize(symbol,size)

            x = int(bbox[0])

            y = int(bbox[1])

            frame = addsymbol.overlay_transparent(frame,symbol,x,y)

        # date and weather

            cd,ct= dateweather.dttm()

            temp = str(dateweather.temp())

            status = str(dateweather.status())

            edge = int(frame.shape[1])

            cv2.putText(frame, cd, (0,50), 0, 0.75, (255,0,0), 2)

            cv2.putText(frame, ct, (0,100), 0, 0.75, (255,0,0), 2 )

            cv2.putText(frame, temp, (edge-200,50), 0, 0.75, (255,0,0), 2)

            cv2.putText(frame, status, (edge-200,100), 0, 0.75, (255,0,0), 2)

        # calculate frames per second of running detections

        fps = 1.0 / (time.time() - start_time)

        print("FPS: %.2f" % fps)

        result = np.asarray(frame)

        result = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)

        cv2.namedWindow("Output Video", cv2.WND_PROP_FULLSCREEN)

        cv2.setWindowProperty("Output
Video",cv2.WND_PROP_FULLSCREEN,cv2.WINDOW_FULLSCREEN)
```

```
        cv2.imshow("Output Video", result)
        # if output flag is set, save video file
        o = 1
        if (o == 1):
            out.write(result)
        if cv2.waitKey(1) & 0xFF == ord('q'): break
    cv2.destroyAllWindows()
if __name__ == '__main__':
    try:
        app.run(dot)
    except SystemExit:
        pass
```

AMC ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Project Outcomes

The student should have the ability to:

CO1: Students will identify the problem through intense literature survey.
CO2: Using the algorithms for making reasonable projections about the future.
CO3: Implement suitable engineering tool to carry out project work using programmingtechniques leading to a logical solution.
CO4: Able to analyze performance data collected over time and under different conditions..
CO5: Able to write the technical report writing .

CO-PO-PSO Mapping

CO-PO-PSO
CO1->PO1,PO2
CO2->PO4,P05,P06
CO3->PO1,P02,P03,P04
CO4->PO2,PO4,PO9,PO10
CO5-> PO9,P010,P012

CO No.	Statement	Bloom's Cognitive level	POs/PSOs
C478.1	**CO1:** Students will identify the problemthrough intense literature survey.	**Understanding**	PO1 ,PO2/PSO 1
C478.2	**CO2:** Using the algorithms for making reasonable projections about the future.	**Apply**	PO4,PO5, PO6/ PSO2
C478.3	**CO3:** Implement suitable engineering tool tocarry out project work using programming techniques leading to a logical solution.	**Understand &Apply**	PO1,PO2, PO3, PO4/PSO 1,PSO 2
C478.4	**CO4:** Able to analyze performance datacollected over time and under different conditions.	**Analyze**	PO2,PO4 ,PO9, PO10/ /PSO1,PS O2
C478.5	**CO5:** Able to write the technical reportwriting.	**Apply**	PO2,PO1 1,PO12/ PSO1,PS O2

PROGRAM OUTCOMES:

PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

PEO 1	Graduates possess advanced knowledge of Computer Science & Engineering and excel inleadership roles to serve the society.
PEO 2	Graduates of the program will apply Computer Engineering tools in core technologies forimproving knowledge in the Interdisciplinary Research and Entrepreneurs.
PEO 3	Graduates adapt Value-Based Proficiency in solving Real Time problems.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO-1	**Professional Skills:** Ability of applying the Computing Concepts, Data Structure,Computer Hardware, Computer Networks and Suitable Algorithm.
PSO-2	**Software Skills:** Ability to build Software Engineering System with Development Life Cycle by using analytical knowledge in Computer Science & Engineering and applyingmodern methodologies.

PO-PSO Mapping

PO's	PSO1	PSO2
PO1	2	1
PO2	3	3
PO3	1	1
PO4	2	2
PO5	1	2
PO6		1
PO7		
PO8		
PO9	1	1
PO10	1	1
PO11	1	1
PO12	1	1

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PS O1	PS O2
C478.1	3	2	-	-	-	-	-	-	-	-	-	-	3	
C478.2				2	3	3								3
C478.3	3	1	2	1									3	2
C478.4		3		2					1	1			3	3
C478.5						1	1	1	2	2	2	2	3	3
Average	**3**	**2**	**2**	**1.6**	**3**	**2**	**1**	**1**	**1.5**	**1.5**	**2**	**2**	**3**	**2.75**

Strength of CO Mapping to PO/PSOs with Justification

CO->PO Mapping	Justification
C478.1->PSO1(3)	Problem may require the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems hence strongly recommended.
C478.1->PO1(3)	Problems may require use of laws of physics, or bring in some mathematical tools in which the problem can be framed hence strongly recommended
C478.1->PO2(2)	Helps in review research literature, and analyze complex engineering problems
C478.2->PSO2(3)	Knowledge of algorithm to analyse the data is used.
C478.2->PO4(2)	The project is built by applying modern methodologies like Internet of Things(IoT) and robotics.
C478.2->PO5(3)	The knowledge of various learning algorithms in robotic vehicle and smart dustbin is depicted.
C478.2->PO6(3)	Apply the IOT methodologies to asses societal ,security and legal issues relevant to our project .
C478.3->PSO1(3)	Understanding and applying the suitable solution for the smart waste management system using IOT.
C478.3->PSO2(2)	Ability to build a suitable solution by applying the project based technologies .
C478.3->PO1(3)	Understand the knowledge of the existing systems like smart garbage system.
C478.3->PO2(1)	Can be used to identify , formulate and review research literature of smart waste management system.
C478.3->PO3(2)	Design the solution for the waste management problem that meet the specified needs with appropriate consideration of public security and conserving their valuable time.

C478.3->PO4(1)	Interpret the data available and propose the best solution and apply it in real time waste management system.
C478.4->PSO1(3)	Analyze the concepts of the project and able to apply the segregation of waste and robobtic vehicle.
C478.4->PSO2(3)	Jusify the solution proposed for the trash barrel guide using the latest technology called Internet of Things(IoT).
C478.4->PO2(3)	Project components are analyzed using principles of mathematics ,science and engineering .
C478.4->PO4(2)	Conduct the proposed solution based on testing methodology like segreagtion of waste,indication when trash is full and the working of robotic vehicle.
C478.4->PO9(1)	Ability to develop expert waste managemen to solve the real time issue faced by the society.
C478.4->PO10(1)	Communicate effectively on the solution and write effective reports and design the documentation smart waste management system and make effective presentation .
C478.5->PSO1(3)	Identify, formulate, and analyze complex problems reaching substantiated conclusions by the application of learning techniques hence medium
C478.5->PSO2(3)	Use research-based knowledge and research methods including design of experiments with the application hence medium
C804.5->PO9(2)	Individual and team work with respect to multidisciplinary applications can be exhibited hence mediumly recommended.
C804.5->PO10(1)	Effective communication and presentation of report can be done hence weakly recommended
C804.5->PO12(2)	Identify and formulate new knowledge from existing knowledge base by using resolution and other inference techniques for lifelong learning hence medium recommended.