INTELLI-DRIVE SYSTEM FOR DETECTING ROAD OBSTACLES

A PROJECT REPORT

Submitted by

R. BRINTHA	**712219104007**
M. HARISHKUMAR	**712219104011**
K. JAYAPRAKASH	**712219104012**
V.N. LAKSHANAA PRIYA	**712219104014**
R. VALLARASU	**712219104028**

In partial fulfillment of the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PARK COLLEGE OF ENGINEERING AND TECHNOLOGY

KANIYUR, COIMBATORE-641659

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2023

INTELLI-DRIVE SYSTEM FOR DETECTING ROAD OBSTACLES

A PROJECT REPORT

Submitted by

R. BRINTHA	**712219104007**
M. HARISHKUMAR	**712219104011**
K. JAYAPRAKASH	**712219104012**
V.N. LAKSHANAA PRIYA	**712219104014**
R. VALLARASU	**712219104028**

In partial fulfillment of the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PARK COLLEGE OF ENGINEERING AND TECHNOLOGY

KANIYUR, COIMBATORE-641659

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report "**Intelli-drive System for Detecting Road Obstacles**" is a Bonafide work "**R.BRINTHA, M.HARISHKUMAR, K.JAYAPRAKASH, V.N.LAKSHANAA PRIYA, R.VALLARASU**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.VIJAYALAKSHMI M.E., Ph.D.	Mr.G.PRABANCHAN., M.E.
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Assistant Professor
Department of Computer Science and Engineering,	Department of Computer Science and Engineering,
Park College of Engineering And Technology,	Park College of Engineering And Technology,
Kaniyur, Coimbatore-641 659	Kaniyur, Coimbatore-641 659

Submitted for the University Project VIVA -VOCE examination held on________________.

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

First and foremost we praise and thank god and our parents, from the bottom of our hearts for bestowing his benediction on us throughout the course of this project.

We wish to express our sincere thanks to our beloved Chairman **Dr.P.V.RAVI Ph.D., MISTE**, for providing an opportunity to work on this project.

We would like to express our deep sense of gratitude to the Chief Executive Officer **Dr. R.ANUSHA B.E., M.S.(USA), Ph.D.,** for her valuable support and encouragement.

We extend sincere thanks to our Principal **Dr. D.LAKSHMANAN M.E., Ph.D.** for supporting and encouraging us to carry out our project.

We are highly indebted and grateful to our beloved Head of the Department **Dr.S.VIJAYALAKSHMI M.E, Ph.D.**, for his valuable suggestions and constant encouragement.

We express our gracious gratitude to our beloved guide **Mr.G.PRABANCHAN M.E,** for her guidance and support throughout our project. We also thank all the faculty members, teaching and non-teaching staff and the lab in charge of our department who helped us to complete our project successfully.

ABSTRACT

The Intelli-Drive System is designed to enhance road safety by detecting and identifying various obstacles on the road. It employs two key technologies: YOLO (You Only Look Once) and OpenCV. Users can conveniently access and utilize the system's functionalities by integrating all the necessary components into a single app. The Kivy app serves as a user interface, providing real-time feedback and visualizations of the detected road obstacles. A test file is used to simulate different driving scenarios to evaluate the system's performance. The test file contains recorded video footage or a sequence of images representing various road conditions. By processing the test file, the Intelli-Drive System demonstrates its ability to detect and classify road lanes, potholes, obstacles, and parking spaces accurately. Furthermore, the system is optimized to convert the evaluated detections into real-time detections. This means that it can process video frames or camera feeds on the fly, continuously analyzing the input and providing instant feedback about road obstacles. This capability is crucial for effective implementation in a real-world driving environment, where timely detection of obstacles is essential for driver awareness and safety. In summary, the Intelli-Drive System combines the power of YOLO and OpenCV within a Kivy app to detect and classify road obstacles, such as road lanes, potholes, objects, and parking spaces. It undergoes evaluation using a test file and is optimized to deliver real-time detections, enabling it to be deployed in real-world driving scenarios.

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ACRONYM		ABBREVIATION
YOLO	-	**Y**ou **O**nly **L**ook **O**nce
OpenCV	-	**Open**-source **C**omputer **V**ision
SSD	-	**S**ingle **S**hot **D**etector
UI	-	**U**ser **I**nterface
GUI	-	**G**raphical **U**ser **I**nterface
CNN	-	**C**onvolutional **N**eural **N**etwork
ML	-	**M**achine **L**earning
DT	-	**D**ecision **T**ree
KNN	-	**K-N**earest **N**eighbor
SVM	-	**S**upport **V**ector **M**achine
LiDAR	-	**Li**ght **D**etection **A**nd **R**anging
BA-CNN	-	**B**at **A**lgorithm **B**ased **C**onvolutional **N**eural **N**etwork
ROI	-	**R**egion **O**f **I**nterest
CODNet	-	**C**lickable **O**bject **D**etection **Net**work
RGB	-	**R**ed **G**reen **B**lue
HFFNet	-	**H**ierarchical **F**eature **F**usion **Net**work
VGG	-	**V**isual **G**eometry **G**roup
R-CNN	-	**R**egion-Based **C**onvolutional **N**eural **N**etwork
DNN	-	**D**eep **N**eural **N**etwork
SZ	-	**S**afety **Z**one
CCTV	-	**C**losed **C**ircuit **T**ele**V**ision

UAV	-	**U**nmanned **A**erial **V**ehicles
FPS	-	**F**rames **P**er **S**econd
GMM	-	**G**aussian **M**ixture **M**odel
AR	-	**A**ugmented **R**eality
HSV	-	**H**ue **S**aturation **V**alue

CHAPTER 1

INTRODUCTION

1.1 INTELLI-DRIVE SYSTEM

Intelli Drive System is a state-of-the-art app designed to improve the driving experience of car owners by utilizing the front and back cameras of their vehicles. The system's advanced computer vision and machine learning algorithms enable it to detect various road features and obstacles, providing drivers with valuable information to avoid accidents and car damage. With its front camera, the Intelli Drive System can recognize the road lane and potholes. Rather than alerting drivers with information, the system displays the detected features on the app's user-friendly interface, making it easy for drivers to stay informed and adjust their driving accordingly.

Similarly, the system's back camera is designed to detect obstacles and parking spaces, displaying them on the app's interface rather than providing information. This feature makes it easier for drivers to navigate through tight spaces and park in busy areas. Overall, the Intelli Drive System is an incredibly useful tool for any driver who wants to enhance their driving experience and stay safe on the road. Its advanced technology and user-friendly interface make it easy to use and understand, providing drivers with an added layer of convenience and safety.

1.2 DETECTION

Detection is the process of recognizing or identifying the presence or existence of a particular object, event, or condition by analyzing a set of signals or data. It involves looking for specific patterns or features in the data that can indicate the presence of the object or event of interest.

1.3 DEEP LEARNING

Deep learning is a subset of machine learning that involves training artificial neural networks with multiple layers to learn and extract features from data. The term "deep" refers to the multiple layers of neurons used in these networks, which allow for the learning of increasingly complex representations of the data as information flows through the network.

1.4 OPENCV

OpenCV (Open Source Computer Vision Library) is a popular open-source computer vision and machine learning software library. It provides developers with a comprehensive set of tools and algorithms for developing applications that involve image and video processing.

One of the most common uses of OpenCV is in object detection. Object detection is the process of identifying and locating objects within an image or video. OpenCV provides several powerful object detection algorithms, including Haar

Cascade Classifiers, HOG (Histogram of Oriented Gradients), and deep learning-based methods such as YOLO (You Only Look Once) and SSD (Single Shot Detector).

1.5 YOLO

YOLO (You Only Look Once) is a real-time object detection algorithm that uses deep learning to identify objects in images or video frames.

YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell. This allows YOLO to make predictions much faster than other algorithms, as it only needs to make a single forward pass through the neural network.

1.6 KIVY

Kivy is an open-source Python framework for creating multi-touch applications that can run on various platforms, including Windows, macOS, Linux, Android, and iOS.

It was designed with user interface (UI) development in mind and provides tools for creating graphical user interfaces (GUIs) that are both easy to use and aesthetically pleasing.

CHAPTER 2

LITERATURE SURVEY

1. **TITLE : Detection Object Open Angle and Direction Using Machine Learning [2]**

AUTHOR : Yiwen Zhang 1, Yongxiong Ren 2, Guodong Xie2, Zhi Wang 1, Hao Zhang 1, Tianxu Xu1, Haoyuan Xu1, Hao Huang2, Changjing Bao2, Zhongqi Pan 3, (Senior Member, IEEE), And Yang Yue1

YEAR : 2020

Object detection techniques using ML algorithms based on received light beam's intensity profiles have seen rapid development and have numerous applications in various industries. The proposed technique's ability to efficiently reduce hardware complexity by using a single-shot image to identify object open angle and direction is impressive. The reliability of the simulation results was confirmed experimentally for 14 open angles and 32 directions. The use of convolutional neural network (CNN) outperforming other traditional ML algorithms like decision tree (DT), k-nearest neighbor algorithm (KNN), and support vector machine (SVM) for image-related tasks is not surprising. The variant of CNN, MobileNet, is particularly interesting due to its relatively simplified iteration algorithm and ability to reduce computational power while maintaining high accuracy for identification issues. The advancements in ML algorithms for object detection have great potential for various industries.

2. **TITLE : Object Detection in Driving Datasets Using a High-Performance Computing Platform: A Benchmark Study [5]**

AUTHOR : Tahir Emre Kalayci, Gabriela Ozegovic, Bor Bricelj, Marko Lah, And Alexander Stocker

YEAR : 2022

Nowadays, machine learning methods are increasingly used in different parts of autonomous driving and driving assistance systems. Yet, data and computational requirements can be enormous with these methods. Thus, providing several datasets containing many and diverse cases for the target problem and sufficient hardware for training and application of ML methods are too critical for achieving accurate results when applying them. Hence, we present an object detection benchmark study implementation the knowledge graph-based data integration framework to meet the data requirements and run the implementation on a big data and high-performance computing (HPC) platform, namely EVOLVE. We applied different object detection models to widely known open datasets and compared the results on three different hardware setups, including EVOLVE.

3. TITLE : CNN-Based Classification for Point Cloud Object with Bearing Angel Image[4]

AUTHOR : Chien-Chou Lin, Chih-Hung Kuo, and Hsin-Te Chiang

YEAR : 2022

Convolutional neural network (CNN), one of the branches of deep neural networks, has been widely used in image recognition, natural language processing, and other related fields with great success recently. This paper proposes a novel framework with CNN to classify objects in a point cloud captured by LiDAR on urban streets. The proposed BA-CNN algorithm is composed of five steps: (i) removing ground points, (ii) clustering objects, (iii) transforming to bearing angle images, (iv) ROI selection, and (V) identifying objects by CNN. In the first step, ground points are removed by multi-threshold-based ground detection to reduce the processing time. Then, a flood-fill-based clustering method is used for object segmentation. Those individual point cloud objects are converted to bearing angle (BA) images. Then, a well-trained CNN is used to classify objects with BA images. The main contribution of this paper is proposing an efficient recognition method that uses information from point clouds only. In contrast, because most 3D object classifiers use the fusion of point clouds and color images, their models are very complicated and take a colossal amount of memory to store the parameters.

4. TITLE : Hierarchically Classification of Very Small Objects: Application to the Detection of Arthropod Species.[1]

AUTHOR : Paul Tresson (Student Member, Ieee), Dominique Carval, Philippe Tixier, And William Puech (Senior Member, Ieee).

YEAR : 2021

Automated image analysis and deep learning tools such as object detection models are being used increasingly by biologists. However, biological datasets often have constraints that are challenging for the use of deep learning. Classes are often imbalanced, similar, or too few for robust learning. In this paper, we present a robust method relying on hierarchical classification to perform very small object detection. We illustrated our result on a custom dataset featuring 22 classes of arthropods used to study biodiversity. This dataset shows several constraints that are frequent when using deep learning on biological data with a high-class imbalance, some classes learned on only a few training examples, and a high similarity between classes. We propose to first perform detection at a super-class level, before performing a detailed classification at a class level. We compare the obtained results with our proposed method to a global detector, trained without hierarchical classification. Our method succeed in obtaining a mAP of 75%, while the global detector only achieves a mAP of 48%.

5. TITLE : Clickable Object Detection Network for a Wide Range of Mobile Screen Resolutions [3]

AUTHOR : Boseon Kang 1, Minseok Jo 2, And Chang-sung Jeong 3, (Member, Ieee)

YEAR : 2022

Recently, as the development cycle of applications has been shortened, it is important to develop rapid and accurate application testing technology. Since application testing required a lot of costs, mobile GUI component detection technology using deep learning is essential to prevent the use of expensive human resources. In this paper, we shall propose a Clickable Object Detection Network (CODNet) for mobile component detection in a wide range of mobile screen resolutions. CODNet consists of three modules: feature extraction, deconvolutional, and prediction modules in order to provide performance improvement and scalability. Feature extraction module uses squeeze and excitation blocks to efficiently extract features by changing the ratio of the input image 1:2 close to that of the mobile screen. The deconvolutional module provides a feature map of various sizes by up sampling the feature map through top-down pathways and lateral connections.

6. TITLE : Hierarchical Feature Fusion Network for Salient Object Detection [7]

AUTHOR : Xuelong Li, Fellow, IEEE, Dawei Song, and Yongsheng Dong, Senior Member, IEEE

YEAR : 2020

Convolutional Neural Network (CNN) has shown their advantages in salient object detection. CNN can generate great saliency maps because it can obtain high-level semantic information. And the semantic information is usually achieved by stacking multiple convolutional layers and pooling layers. However, multiple pooling operations will reduce the size of the feature map and easily blur the boundary of the salient object. Therefore, such operations are not beneficial to generate great saliency results. To alleviate the issue, we propose a novel edge information-guided hierarchical feature fusion network (HFFNet). Our network fuses feature hierarchically and retain accurate semantic information and clear edge information effectively. Specifically, we extract image features from different levels of VGG. The proposed HFFNet has been extensively evaluated on five traditional benchmark datasets. The experimental results demonstrate that the proposed model is fairly effective in salient object detection compared with 10 state-of-the-art models under different evaluations.

7. TITLE : Thermal Object Detection in Different Weather Conditions Using YOLO[6]

AUTHOR : Mate Kristo, Marina Ivasic-kos (Member, Ieee), And Miran Pobar

YEAR : 2020

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 be used at night and in weather conditions where RGB cameras do not perform well, thermal cameras have been 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 R-CNN, 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 and recorded at night in clear weather, rain, and in fog, at different ranges, and with different movement types. YOLOv3 was significantly faster than other detectors while achieving performance comparable to the best, so it was used in further experiments.

8. TITLE : Online Safety Zone Estimation and Violation Detection for Nonstationary Objects in workplaces [2]

AUTHOR : Hyunjoong Cho 1, Kyuiyong Lee 1, Nakkwan Choi 1, Seok Kim2, Jinhwi Lee2, And Seungjoon Yang 1

YEAR : 2022

This study presents a deep neural network (DNN)-based safety monitoring method. Nonstationary objects such as moving workers, heavy equipment, and pallets were detected, and their trajectories were tracked. Time-varying safety zones (SZs) of moving objects were estimated based on their trajectories, velocities, proceeding directions, and formations. SZ violations are defined by set operations with sets of points in the estimated SZs and the object trajectories. The proposed methods were tested using images acquired by CCTV cameras and virtual cameras in 3D simulations in plants and on loading docks. DNN-based detection and tracking provided an accurate online estimation of time-varying SZs that were adequate for safety monitoring in the workplace. The set operation-based SZ violation definition was flexible enough to monitor various violation scenarios that are currently monitored in workplaces. The proposed methods can be incorporated into existing site monitoring systems with single-view CCTV cameras at vantage points.

9. TITLE : Low-altitude small-sized object detection using lightweight feature-enhanced convolutional neural network

AUTHOR : Ye tao1, Zhao zongyang1, Zhang jun1, Chai xinghua2, And Zhou fuqiang3

YEAR : 2021

Unauthorized operations referred to as “black flights” of unmanned aerial vehicles (UAVs) pose a significant danger to public safety, and existing low-attitude object detection algorithms encounter difficulties in balancing detection precision and speed. Additionally, their accuracy is insufficient, particularly for small objects in complex environments. To solve these problems, we propose a lightweight feature-enhanced convolutional neural network able to perform detection with high precision detection for low-attitude flying objects in real-time to provide guidance information to suppress black-flying UAVs. The proposed network consists of three modules. The proposed method achieves a detection speed of 147 frames per second (FPS) and a mean average precision (mAP) of 90.97% for a dataset composed of flying objects, indicating its potential for low-altitude object detection.

10. TITLE : Fast Synthetic Dataset for Kitchen Object Segmentation in Deep Learning

AUTHOR : Ruben Sagues-tanco 1,2, (Graduate Student Member, Ieee), Luis Benages-pardo 1,2, Gonzalo López-nicolás 1,2, (Senior Member, Ieee), And Sergio Llorente 3

YEAR : 2020

Object recognition has been widely investigated in computer vision for many years. Currently, this process is carried out through neural networks, but there are very few public datasets available with mask and class labels of the object for the training process in usual applications. In this paper, we address the problem of fast generation of synthetic datasets to train neural models because creating a handcrafted labeled dataset with object segmentation is a very tedious and time-consuming task. We propose an efficient method to generate a synthetic labeled dataset that adequately combines background images with foreground segmented objects. The synthetic images can be created automatically with random positioning of the objects or, alternatively, the method can produce realistic images by keeping the realism in the scales and positions of the objects.

11. TITLE : MD3D: Mixture-Density-Based 3D Object Detection in Point Clouds [8]

AUTHOR : Jaeseok Choi 1, Yeji Song 1, Yerim Kim 1, Jaeyoung Yoo 2, And Nojun Kwak 1, (Senior Member, Ieee)

YEAR : 2022

The design factors of anchor boxes, such as shape, placement, and target assignment policy, greatly influence the performance and latency of the 3D object detectors. Unlike image-based 2D anchors, 3D anchors must be placed in a 3D space and determined differently for each class for different sizes. This imposes a significant burden on the design complexity. To tackle the issue, various studies have been conducted on how to set the anchor form. Consequently, only objects that are similar in shape and size to an anchor can obtain high accuracy. In this paper, we propose a Mixture-density based 3D Object Detection (MD3D) in point clouds to predict the distribution of 3D bounding boxes using a Gaussian Mixture Model (GMM). With an anchor-free detection head, MD3D requires few hand-crafted design factors and eliminates the inefficiency of separating the regression channel for each class and thus offering both latency and memory benefits. MD3D is designed to utilize various types of feature encoding; therefore, it can be applied flexibly by replacing only the detection head of the existing detectors.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system for road lanes, potholes, obstacles, and parking space detection relies on image and video analysis technology. However, this system is not able to detect these features in real time and provides low accuracy in detecting these features. The image and video-based technology work by analyzing pre-recorded data, either from still images or recorded video footage, to identify the presence of road lanes, potholes, obstacles, and parking spaces. While this approach can provide valuable insights into road conditions and assist drivers in navigating challenging situations, it is not suitable for real-time detection. Moreover, the accuracy of the system is limited, as it relies on the quality of the images and videos available for analysis. If the images or video footage are blurry, poorly lit, or otherwise compromised, the system may not be able to accurately detect these features, leading to potential safety risks for drivers.

Overall, while the existing system for road lane, pothole, obstacle, and parking space detection provides some valuable insights into road conditions, its limitations in terms of real-time detection and accuracy highlight the need for more advanced technology to improve road safety and enhance the driving experience.

3.2 PROBLEM STATEMENT

The problem with the existing system for road lane, pothole, obstacle, and parking space detection is that it relies on image and video analysis technology that is not capable of real-time detection and provides low accuracy. This approach can only analyze pre-recorded data from still images or recorded video footage, which is not suitable for detecting these features in real time. Additionally, the accuracy of the system is limited by the quality of the images and videos available for analysis, which may lead to potential safety risks for drivers if the system fails to accurately detect these features. Therefore, there is a need for more advanced technology that can provide real-time detection and higher accuracy to improve road safety and enhance the driving experience.

3.3 PROPOSED SYSTEM

A proposed system for enhancing the existing image and video-based technology for road lane, pothole, obstacle, and parking space detection is to develop a real-time system that utilizes the car's front and back cameras. The system would incorporate advanced computer vision and machine learning algorithms, such as YOLO (You Only Look Once) and OpenCV, to analyze the live video feed from the car's cameras and identify these features with greater accuracy and speed.

Using the front camera, the system could detect road lanes and potholes in real-time and indicate their presence and location on an augmented reality (AR) display.

The back camera could similarly detect obstacles and available parking spaces, also displaying them on the AR display to assist the driver in navigating through tight spaces and parking in busy areas.

The use of YOLO and OpenCV would enable the system to detect these features with high accuracy by leveraging deep learning techniques to learn and adapt to different road conditions and environments. These algorithms allow the system to analyze the video feed in real time, identifying and tracking objects with high precision and low latency. Overall, the proposed system would significantly enhance the existing technology for road lane, pothole, obstacle, and parking space detection by providing real-time detection and greater accuracy, leading to improved road safety and a better driving experience for car owners.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM WORKFLOW

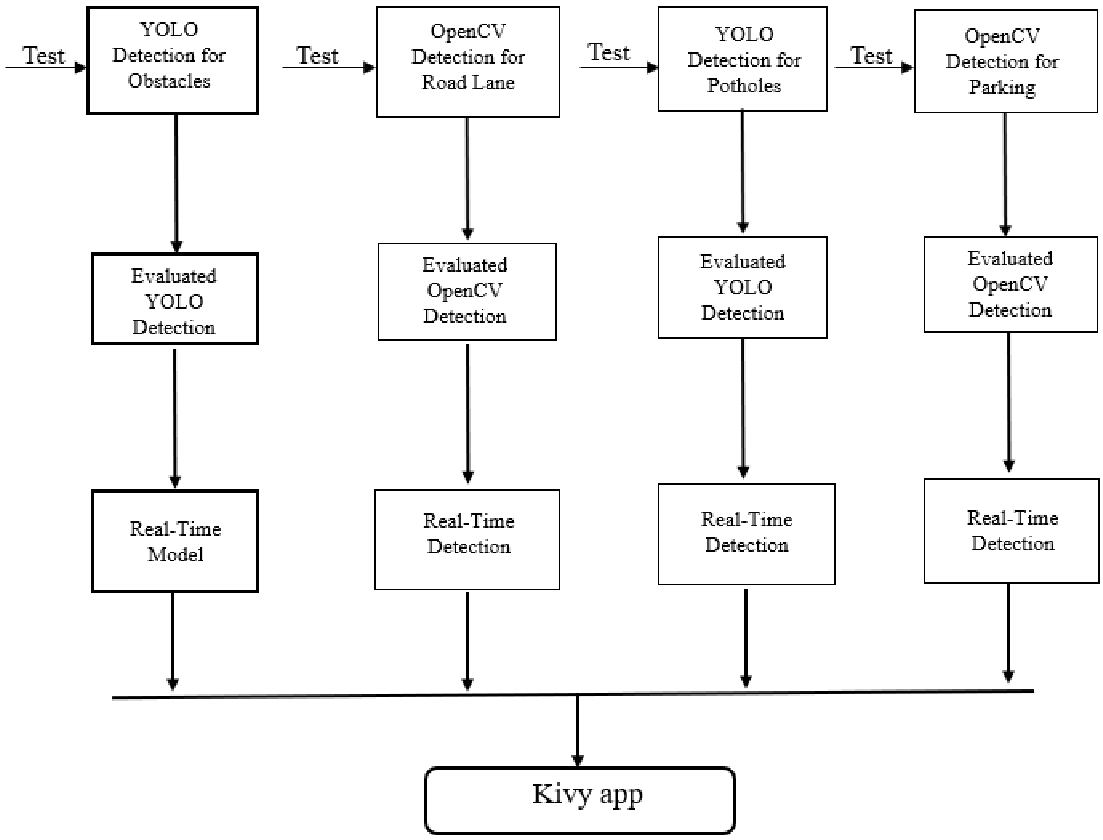

Figure 4.1 System Workflow

The flowchart illustrates the process of object and pothole detection using YOLO, which is evaluated with the test file and then converted into real-time detection. Similarly, the road lane, and parking space detection using OpenCV are also evaluated with the test file and converted into real-time detections. All of these detections are working under a single Kivy app. In this project, we evaluated the YOLO model using a test file and converted it into real-time detection. Additionally, we also implemented road lane, and parking space detection using OpenCV. Like YOLO, these detectors were evaluated with a test file and then converted into real-time detection. Finally, all these detectors were integrated into a single Kivy app, which allows users to switch between different detection modes seamlessly. The app displays real-time video from the camera feed and overlays the detected objects and road features on top of it. This app can be useful for a variety of applications, such as autonomous driving, traffic monitoring, and parking lot management.

4.2 APP INTERFACE DESIGN

Figure 4.2 App Initial Screen

The initial screen of the Intelli-Drive System displays a logo and serves as the entry point for the application. When the user interacts with the logo, it redirects them to the next screen, which is the main screen of the application.

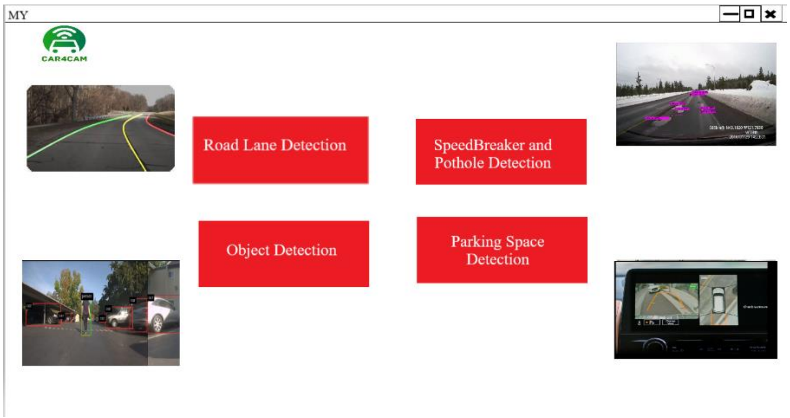

Figure 4.3 App Interface

The application's main screen features four buttons, each representing a specific functionality or mode of the Intelli-Drive System. These buttons serve as the user interface for accessing different real-time detection capabilities.

CHAPTER 5

IMPLEMENTATION

5.1. FRONT CAMERA DETECTIONS

5.1.1 ROAD LANE DETECTION

The concept behind the code is to use computer vision techniques to detect and draw lanes in a video. In this detection, a region of interest in the video frames where the lanes are expected to be present. This helps narrow down the focus and reduces the computational load. Then the frames are preprocessed to enhance the lane edges. The frames are converted to grayscale to simplify the image data. Dilation is applied to thicken the lane lines and make them more distinguishable. After the preprocessing, the Canny edge detection algorithm is applied to identify the edges in the preprocessed grayscale frames. This helps identify the boundaries between different objects in the image. A mask is created using the defined ROI, which is then applied to the edge-detected frames. This restricts the further analysis and processing to the region of interest only.

The Hough line transform is used to detect lines in the region of interest. It converts the edge-detected image into a parameter space, where lines can be represented by certain patterns. This helps in detecting straight lines that correspond to the lanes. The detected lines from the Hough transform are drawn on the original frames. This visually highlights the detected lanes in the video. The processed frames with the drawn lanes are saved into a new video file using a VideoWriter. After processing, the frames are saved and displayed in a video then it can be converted into real-time using cv2.VideoCapture(0) and cv2.imshow().

By repeating these steps for each frame of the video, the code is able to detect and visualize the lanes in the video, providing a lane detection output.

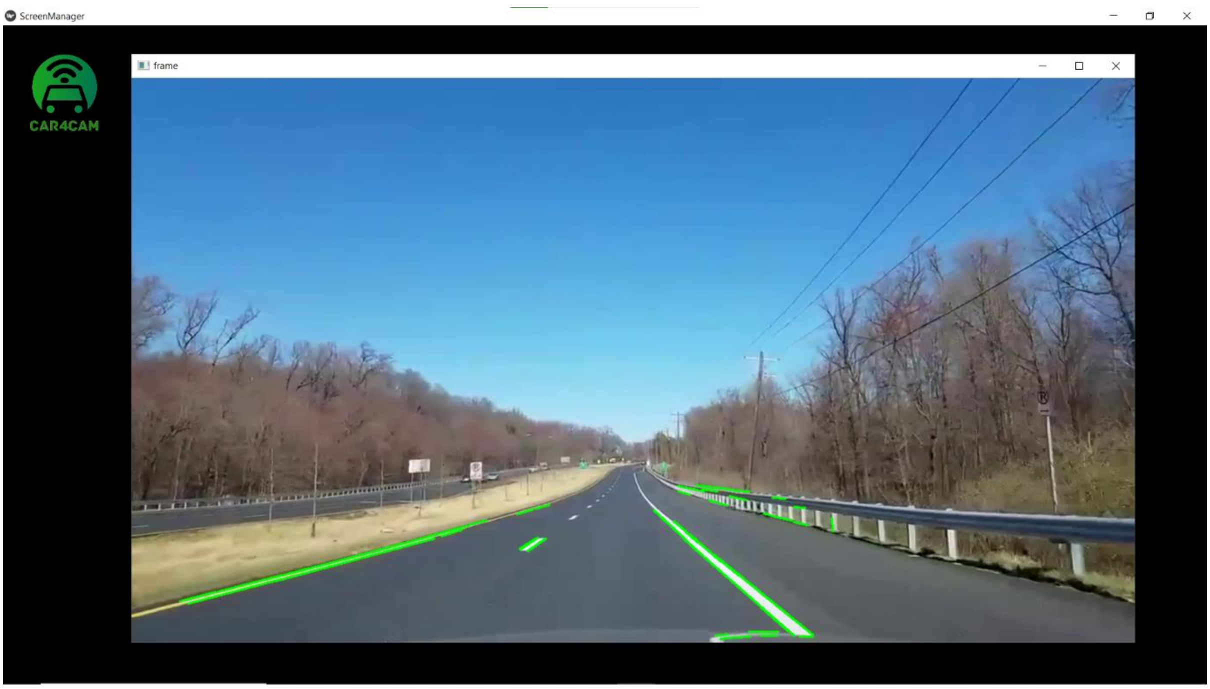

Figure 5.1 Tested Road Lane Detection

5.1.2 POTHOLE DETECTION

In this pothole detection, YOLOv4 can be trained to detect the pothole in the road. The YOLOv4 Tiny model is loaded using its pre-trained weights and configuration files. A video source (camera or video file) is defined to capture frames for analysis. The loaded YOLOv4 Tiny model is used to detect potholes in each frame of the video stream.

For each detected pothole, a bounding box is drawn on the frame.: The coordinates of the detected potholes are saved in text files, and the frames with the bounding boxes are saved as images. The frames per second (FPS) are calculated based on the time taken to process each frame. The processed frames with the bounding boxes are displayed in real-time and saved into a video file.

Once the video stream is finished or the user terminates the program, the necessary resources are released and windows are closed.

In summary, the trained YOLOv4 Tiny model can be used to detect potholes in a video stream. It draws bounding boxes around the detected potholes, saves their coordinates and images, calculates the FPS, and displays the processed frames in a video and it can be converted into real-time.

Figure 5.2 Tested Pothole Detection

5.2 BACK CAMERA DETECTIONS

5.2.1 OBSTACLE DETECTION

Obstacle Detection using YOLO performs real-time object detection on a video stream from a camera. We set up the necessary parameters, such as the path to the pre-trained model, the confidence score threshold for detections, and the size of the input image.

It then loads the pre-trained model and initializes the video stream from the default camera. Then read the frames from the video stream, resizes the frames, and convert them to blobs. These blobs are then passed through the SSD model to obtain the detections.

For each detection, it checks if the confidence score exceeds the defined threshold. If so, it draws a rectangle around the object and displays the label of the object and its confidence score.

When the user presses the "q" key, it calculates and displays the elapsed time and approximate frames per second (FPS) at the end. This detection can be used as a starting point for building more advanced applications that require real-time object detection, such as security systems, robotics, or self-driving cars.

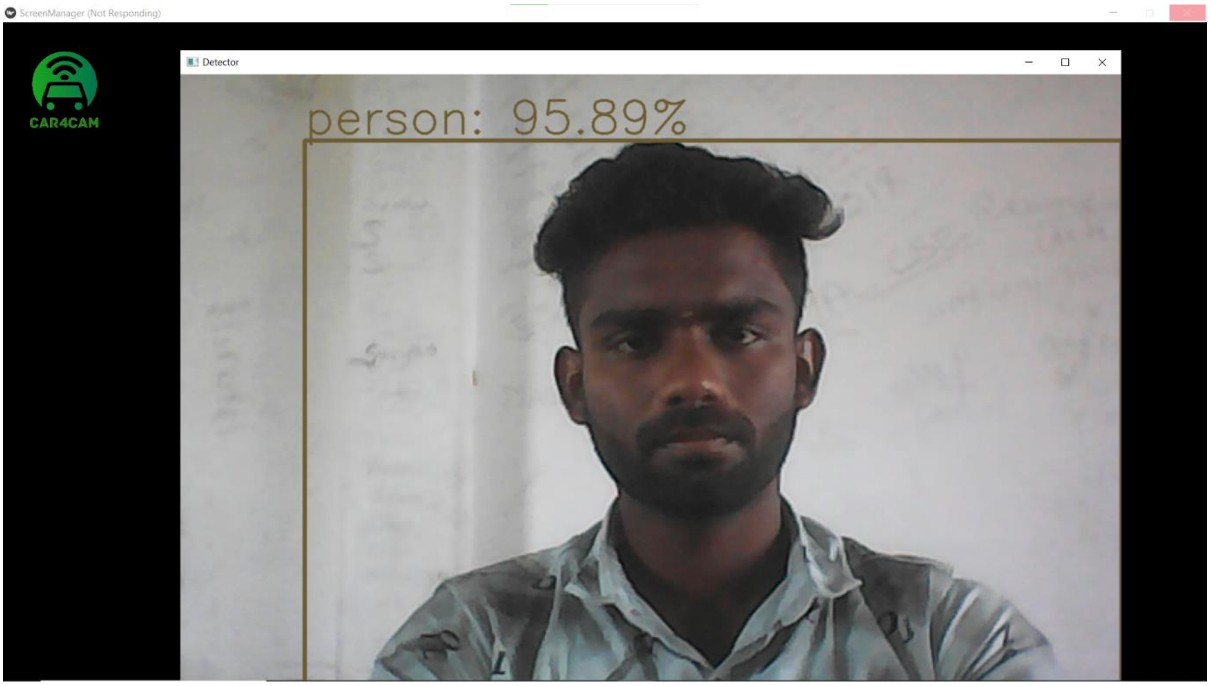

Figure 5.3 Obstacle Detection

5.2.2 PARKING DETECTION

The Parking Detection can be implemented by a computer vision-based parking assistant system using the OpenCV library. The detection utilizes the live video stream from a camera, and detects whether a parking spot is vacant or occupied by any obstacles

. This detection aims to perform real-time color and modification within a specific parallelogram region in video frames captured from a camera. It begins by initializing the camera capture and defining a function called draw_parallelogram for processing each frame. Inside this function, the parallelogram region is determined based on the frame dimensions, and a mask is created to isolate that region. The frame is then converted to the HSV color space for color detection. The code defines a range of green color in HSV and creates a mask to detect green objects within the parallelogram region. Contours of green objects are found in the mask, and if any contour is detected, the code replaces the green color within the contour with transparent and adds a text annotation indicating “Detecting Some Obstacles”. If no object is detected, the parallelogram region remains green, and a different text annotation suggests “You can park here”. The modified parallelogram region is combined with the original frame, and the resulting frame is displayed in real-time. The code continuously captures frames, processes them, and presents the modified frames until the user presses the 'q' key to exit.

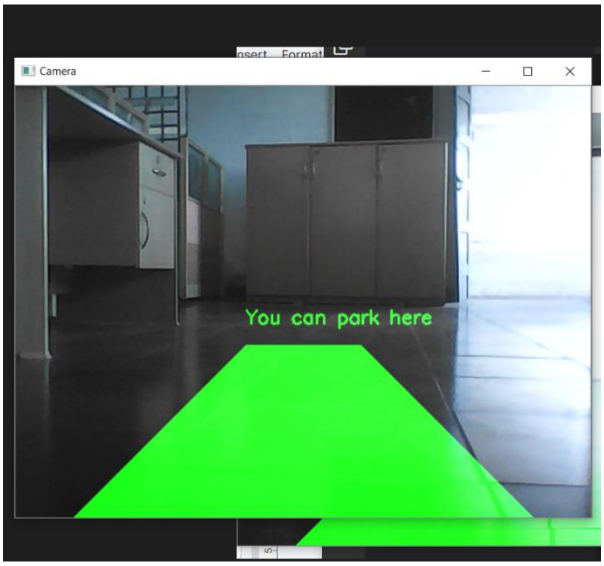

Figure 5.4 Parking Detected

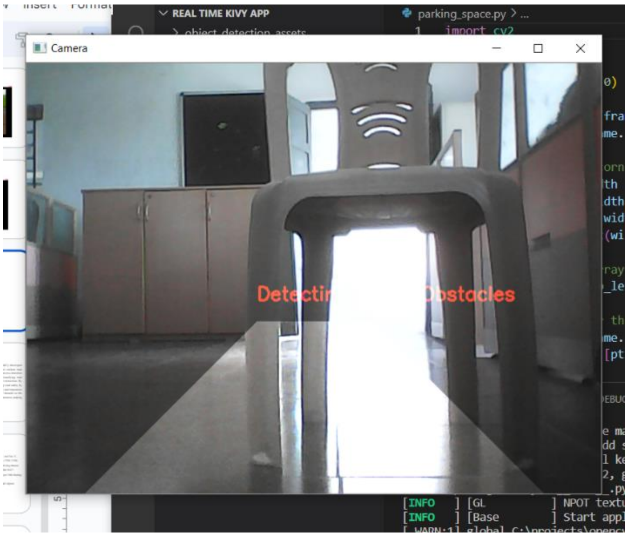

Figure 5.5 Obstacle Detected

5.3 APP IMPLEMENTATION

The main screen of the application features four buttons, each representing a specific functionality or mode of the Intelli-Drive System. These buttons serve as the user interface for accessing different real-time detection capabilities.

Road Lane Detection: Clicking on the "Road Lane Detection" button activates the road lane detection mode. In this mode, OpenCV algorithms process the camera feed or video frames to identify and track the road lanes. The system detects the lane markings, highlights them on the screen, and provides guidance to the driver to stay within the lanes.

Pothole Detection: Clicking on the "Pothole Detection" button activates the pothole detection mode of the system. In this mode, the camera feed or video input is processed using the YOLO algorithm trained for pothole detection. The system analyzes the frames in real time, identifies and marks the presence of potholes on the road, and provides visual feedback to the user.

Obstacle Detection: The "Obstacle Detection" button triggers the object detection mode. In this mode, the system utilizes YOLO to detect and classify various objects in the camera feed or video input. It can recognize and label objects such as vehicles, pedestrians, or road signs in real time. The detected objects are displayed on the screen, allowing the user to be aware of potential obstacles on the road.

Parking Detection: When the user selects the "Parking Detection" button, the system switches to the parking detection mode. OpenCV algorithms are employed to analyze the camera feed or video frames and identify available parking spaces. The system detects whether a parking spot is vacant or occupied by obstacles in real-time.

Overall, the main screen of the Intelli-Drive System provides access to these four important functionalities through buttons. Each mode utilizes specific algorithms (YOLO and OpenCV) to perform real-time detection of road obstacles, such as potholes, objects, parking, and road lanes. By interacting with the buttons, the user can choose the desired mode and benefit from the system's visual feedback

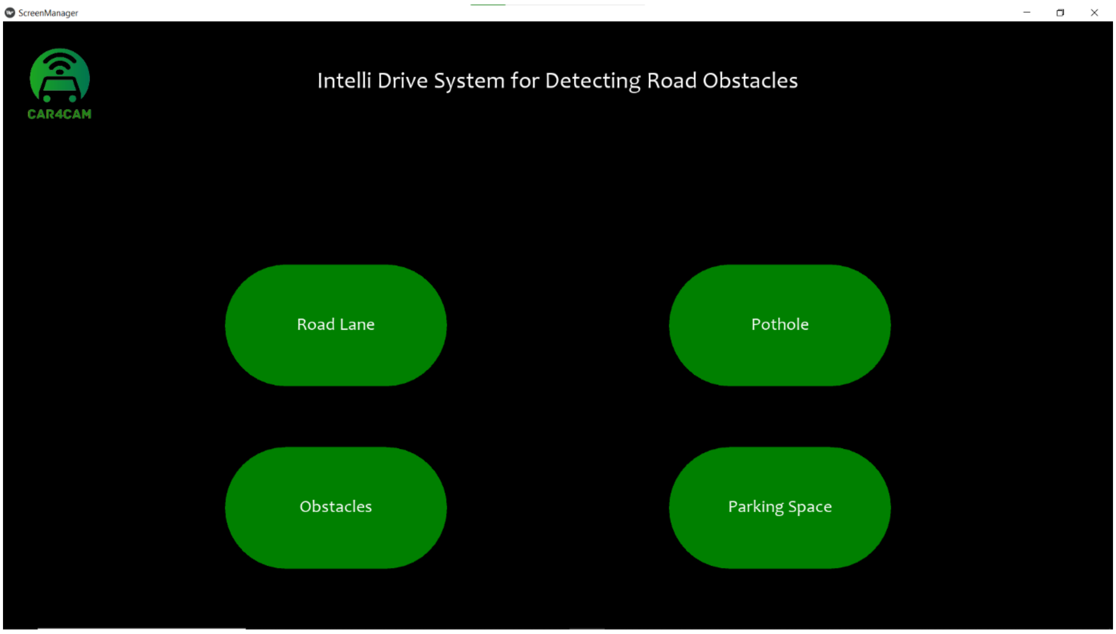

Figure 5.6 Integrating Detections into Kivy App

CHAPTER 6

RESULTS AND DISCUSSIONS

In the Intelli-Drive System for detecting road obstacles, the performance can be evaluated based on the detection percentage in object and pothole detection, as well as the Frames Per Second (FPS) achieved during the detection process.

Detection Percentage:

The detection percentage refers to the accuracy and effectiveness of the system in correctly identifying and classifying road obstacles, such as objects and potholes. To calculate the detection percentage, the system compares the detections made by YOLO (for objects and potholes) against ground truth or manual annotations. The percentage is determined by dividing the number of correctly detected objects or potholes by the total number of objects or potholes present in the test files.

Detection Percentage = (Number of Correctly Detected Objects / Total Number of Objects) * 100

The system achieves a detection percentage **above 85%**, indicating a high level of accuracy in identifying and classifying road obstacles. This accuracy is crucial for ensuring the safety and reliability of the system's detections.

Frames Per Second (FPS):

FPS measures the number of frames processed by the system per second. It is an important metric to evaluate the real-time performance of the Intelli-Drive System. Higher FPS values indicate that the system can process video frames more quickly, allowing for smoother and more responsive detections. The FPS can be calculated by dividing the total number of frames processed during a specific period by the time taken to process those frames. the system achieves an average Frames Per Second (FPS) rate of 6-7, demonstrating its real-time processing capabilities. This means that the system can efficiently analyze and process video frames, providing timely and responsive detections to the driver.

The formula for calculating FPS is as follows:

FPS = Number of Frames / Time Taken

where,

Number of Frames: This refers to the total count of frames processed by the system within a specific period. It can be obtained by tracking the number of frames as they are being processed during the detection phase.

Time Taken: This represents the duration of time it takes for the system to process the frames. It can be measured using a timer or by recording the start and end times of the detection process.

To calculate the FPS, divide the number of frames by the time taken. The resulting value represents the average number of frames processed per second.

CHAPTER 7

CONCLUSION

In conclusion, the Intelli-Drive System for detecting road obstacles has been successfully developed with satisfactory performance. The system utilizes YOLO and OpenCV to detect various road obstacles, including objects, potholes, road lanes, and parking spaces. The system achieves a detection percentage above 80%, indicating a high level of accuracy in identifying and classifying road obstacles. This accuracy is crucial for ensuring the safety and reliability of the system's detections. By achieving these performance metrics, the System proves to be effective in enhancing road safety by providing real-time information about road obstacles. With its high detection accuracy and responsive processing speed, the system can assist drivers in navigating and avoiding potential hazards on the road. Overall, it has the potential to greatly improve driving safety and situational awareness, making it a valuable tool for drivers and contributing to safer roads.

REFERENCES

[1] Zhang, Y., Ren, Y., Xie, G., Wang, Z., Zhang, H., Xu, T., Xu, H., Huang, H., Bao, C., Pan, Z. and Yue, Y., 2020. Detecting object open-angle and direction using machine learning. IEEE Access, 8, pp.12300-12306.

[2] Kalayci, T.E., Ozegovic, G., Bricelj, B., Lah, M. and Stocker, A., 2022. Object detection in driving datasets using a high-performance computing platform: A benchmark study. IEEE Access, 10, pp.61666-61677.

[3] Lin, C.C., Kuo, C.H. and Chiang, H.T., 2021. CNN-Based Classification for Point Cloud Object With Bearing Angle Image. IEEE Sensors Journal, 22(1), pp.1003- 1011.

[4] Tresson, P., Carval, D., Tixier, P. and Puech, W., 2021. Hierarchical classification of very small objects: Application to the detection of arthropod species. IEEE Access, 9, pp.63925-63932.

[5] Kang, B., Jo, M. and Jeong, C.S., 2022. Clickable Object Detection Network for a Wide Range of Mobile Screen Resolutions. IEEE Access, 10, pp.115051-115060.

[6] Li, X., Song, D. and Dong, Y., 2020. Hierarchical feature fusion network for salient object detection. IEEE Transactions on Image Processing, 29, pp.9165-9175.

[7] Krišto, M., Ivasic-Kos, M. and Pobar, M., 2020. Thermal object detection in difficult weather conditions using YOLO. IEEE Access, 8, pp.125459-125476.

[8] Cho, H., Lee, K., Choi, N., Kim, S., Lee, J. and Yang, S., 2022. Online Safety Zone Estimation and Violation Detection for Nonstationary Objects in Workplaces. IEEE Access, 10, pp.39769-39781.

[9] Tao, Y., Zongyang, Z., Jun, Z., Xinghua, C. and Fuqiang, Z., 2021. Low-altitude small-sized object \ detection using lightweight feature-enhanced convolutional neural network. Journal of Systems Engineering and Electronics, 32(4), pp.841-853.

[10] Sagues-Tanco, R., Benages-Pardo, L., Lopez-Nicolas, G. and Llorente, S., 2020. Fast synthetic dataset for kitchen object segmentation in deep learning. IEEE Access, 8, pp.220496-220506.

[11] Choi, J., Song, Y., Kim, Y., Yoo, J. and Kwak, N., 2022. MD3D: Mixture-DensityBased 3D Object Detection in Point Clouds. IEEE Access, 10, pp.104011-104022.