Real-time Detection of Infrastructure Obstacles for Electric Scooters

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Master of Science

by

Zeyang Zheng

May 2025

Abstract

The growing adoption of electric scooters (e-scooters) in urban environments has corresponded with a notable increase in traffic accidents and injuries. Due to their smaller wheels, lack of suspension system, and significant variances in the system center of gravity among different riders, e-scooters are more susceptible to the negative effects of uneven surfaces.

While deep learning-based object detection has been successfully applied to improve automobile safety, its potential for obstacle detection especially for e-scooters remains unexplored. This study introduces a novel ground obstacle detection system for e-scooters, integrating an RGB camera, and a depth camera to enhance real-time road hazard detection. Additionally, the Inertial Measurement Unit (IMU) measures linear vertical acceleration to identify surface vibrations, guiding the selection of six obstacle categories: tree branches, manhole covers, potholes, pine cones, non-directional cracks, and truncated domes. All sensors, including the RGB camera, depth camera, and IMU, are integrated within the Intel RealSense Camera D435i. A deep learning model powered by YOLO (You Only Look Once) detects road hazards and utilizes depth data to estimate obstacle proximity. Evaluated on the seven hours of naturalistic riding dataset, the system achieves a high mean average precision (mAP) of 0.827 and demonstrates excellent real-time performance.

This approach provides an effective solution to enhance e-scooter safety through advanced computer vision and data fusion. This system enhances e-scooter rider safety by detecting road infrastructure hazards, leveraging advanced computer vision, and data fusion to provide real-time hazard awareness.

Acknowledgments

I am deeply grateful to my advisor, Professor Arsalan Heydarian, for his insightful guidance, continuous encouragement, and patience throughout my master's studies. His mentorship has been instrumental in shaping both my academic growth and this research.

I also sincerely appreciate Professor Dong Chen, a former postdoctoral researcher, for his expertise and dedication, which significantly contributed to the progress of this project. His valuable insights and unwavering support have been indispensable.

Furthermore, I extend my heartfelt gratitude to my colleague and collaborator, Arman Hosseini, whose guidance and advice greatly accelerated this dissertation. His contributions have been invaluable.

Finally, I express my appreciation to my committee members, Professor Brad Campbell and Professor Somayeh Asadi, for their thoughtful feedback and continuous support throughout my graduate studies. Their perspectives and insights have been essential in refining this work.

Executive summary

With increasing adoption of electric scooters (e-scooters) in urban environments, the number of related traffic accidents and injuries has risen significantly. Because of their small wheels, lack of suspension systems, and variations in riders' centers of gravity, e-scooters are more vulnerable to uneven surfaces, which increases the risk of accidents. Researchers have successfully applied deep learning to object detection in autonomous vehicles, but its potential for e-scooter obstacle detection remains largely unexplored.

This study develops a real-time infrastructure obstacle detection system to enhance e-scooter rider safety. Its objectives are as follows:

1. Reviewing related work identifies gaps in e-scooter safety research.
2. Construct a dedicated obstacle dataset for e-scooters, focusing on obstacle categories that affect ride stability.
3. Evaluate and select an object detection model that balances real-time performance and accuracy when identifying road obstacles relevant to e-scooters.
4. Develop a real-time obstacle detection system that integrates an RGB camera and a depth camera for obstacle distance estimation.

This study employs the Intel RealSense D435i as the main sensor, which includes an RGB camera, a depth camera, and an IMU for data collection. The dataset features six types of infrastructure

obstacles (tree branches, manhole covers, potholes, pine cones, non-directional cracks, and truncated domes). These obstacles were chosen based on the IMU's measurement of linear vertical acceleration, which helps assess vibrations caused by different obstacles. The method compares multiple efficient YOLO models to identify the most suitable model for low-power computing devices. The system uses YOLO object detection for real-time hazard identification and incorporates depth information for obstacle distance estimation.

Experimental results show strong performance in detecting e-scooter road obstacles:

- **Object Detection Performance**: YOLOv5s offers the best trade-off between speed and accuracy, achieving a mean Average Precision (mAP50) of 0.827 on the test dataset with 15.8 GFLOPs. Among the obstacle categories, manhole covers (mAP50: 0.917) and truncated domes (mAP50: 0.985) reached the highest detection accuracy, whereas non-directional cracks (mAP50: 0.649) and potholes (mAP50: 0.725) performed less effectively.

- **Real-time Processing**: On a personal laptop equipped with an NVIDIA GeForce RTX 3050 (6GB) GPU, the system maintained an inference speed exceeding 100 FPS, requiring under 10 ms per frame on average.

- **Obstacle Distance Estimation**: Depth data integration enabled accurate distance estimation for obstacles within 10 meters.

This study introduces the first real-time infrastructure obstacle detection system specifically tailored for e-scooters, demonstrating effective detection accuracy and real-time performance. Future research directions include:

1. **Dataset Expansion**: Incorporate more obstacle categories (e.g., stones, broken glass) and diverse environmental conditions (e.g., lighting, weather) to improve model generalization. Collaborate with other universities to gather additional data and further strengthen the dataset's robustness.

2. **Model Optimization**: Explore additional models to enhance accuracy and reduce computational costs.
3. **Deployment on Embedded Devices**: Implement the model on platforms such as the NVIDIA Jetson Orin Nano or similar low-power AI boards suitable for e-scooters.
4. **Warning Systems**: Provide more intuitive rider alerts through audio signals, haptic feedback (e.g., vibrating handlebars), or augmented reality (AR) interfaces.

This research offers a practical solution for e-scooter infrastructure obstacle detection, enhances rider safety, and lays the foundation for future intelligent micro-mobility systems.

Contents

Chapter 1

Introduction

1.1 Motivation and problem statement

E-scooters have emerged as a popular and convenient mode of short-distance transportation. According to a market analysis report [1], the global electric scooter market is valued at USD 4.3 billion in 2024, with projections estimating growth to USD 12.4 billion by 2030, at a compound annual growth rate (CAGR) of 18.9%. Figure 1.1 illustrates historical trends and projected growth in the global e-scooter market from 2018 to 2030, with projections beginning in 2024.

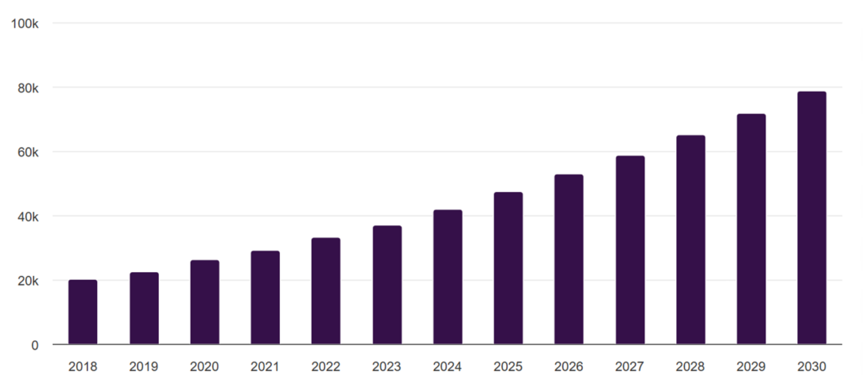


Figure 1.1: Global E-Scooter Market Growth and Projections (2018–2030) [1]

Despite their growing popularity, the number of e-scooter-related accidents has risen significantly in recent years. In the United States, e-scooter injuries increased dramatically from 8,566 in 2017 to 56,847 in 2022, with hospitalizations from these injuries showing a consistent upward trend [2]. Figure 1.2 presents the increase in e-scooter-related injuries in the United States from 2017 to 2022. The Injury means those resulting in hospitalization.

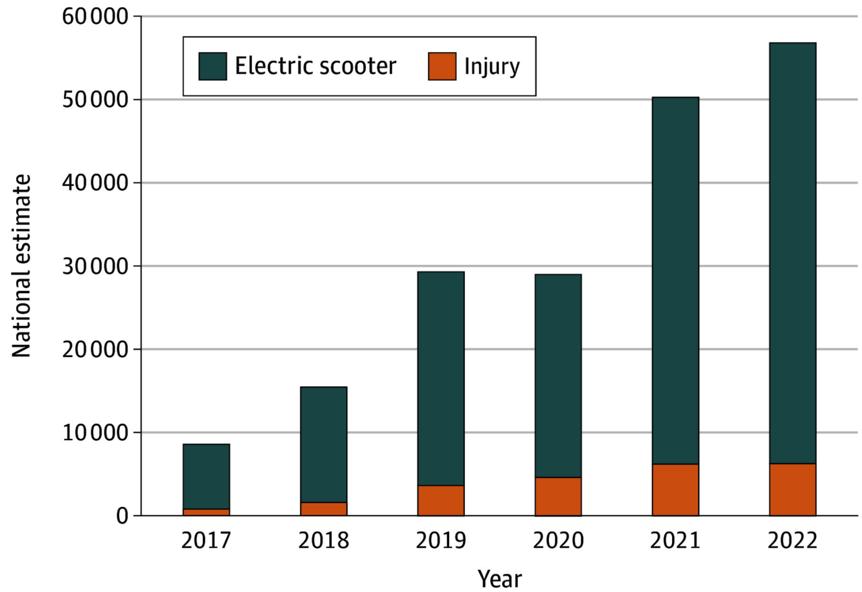


Figure 1.2: Trends in E-Scooter-Related Injuries in the United States (2017–2022) [2]

Several factors contribute to the high incidence of e-scooter-related injuries. The current design of small-wheeled electric scooters often overlooks the complex relationship between wheel size and stability, resulting in improper steering geometries [3]. As a consequence, these scooters require higher speeds to achieve self-stability and demand greater rider control to maintain balance at lower speeds. Infrastructure inadequacies exacerbate these challenges, as road obstacles such as potholes and manholes pose significant hazards for e-scooter users. Among various obstacles, riders have identified potholes and rough road conditions as their primary concerns [4]. Figure 1.3 illustrates the responses of 309 e-scooter riders regarding their concerns about encountering different obstacles.

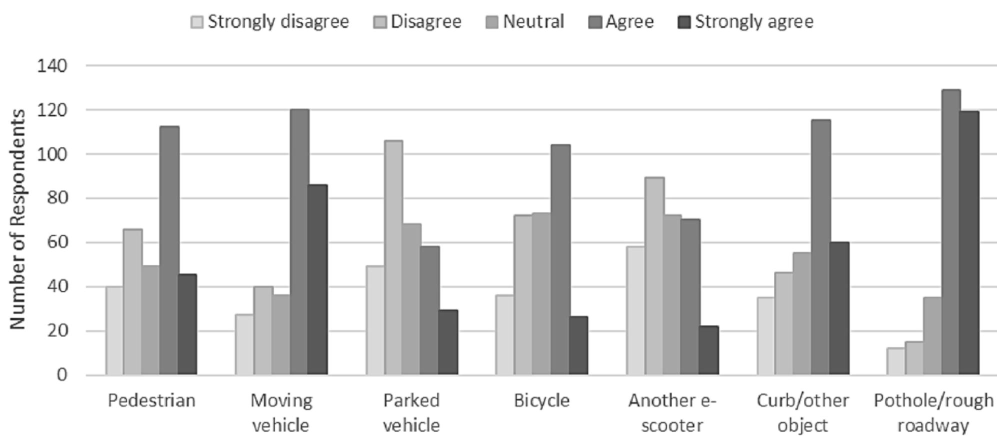


Figure 1.3: E-Scooter Riders' Concerns About Road Obstacles [4]

Additionally, smaller obstacles like pinecones along roadways have been shown to increase riders' visual attention and cognitive load as they navigate to avoid them [5]. Moreover, high speeds (>15km/h) greatly increase e-scooter vibrations, which negatively impacts comfort and stability [6]. The center of gravity and mass play critical roles in maintaining the e-scooter's stability. Furthermore, a rider's position can shift the system center of gravity, indirectly influencing control stability and introducing uncertainty. Compounding these issues, most e-scooters lack shock absorbers that could alleviate severe vibrations. Consequently, electric scooters experience higher vibration frequencies and provide lower comfort levels compared to bicycles [7].

Deep learning has gained significant attention across various disciplines, particularly in transportation safety-related research. One prominent application is object detection in Advanced Driver Assistance Systems (ADAS), which play a critical role in enhancing the safety of Autonomous Vehicles (AVs) [8]. ADAS leverage object detection algorithms to efficiently identify objects such as vehicles, micromobility devices, and road obstacles in real time, while LiDAR sensors measure the distances to nearby objects. Despite the extensive study of object detection in the context of autonomous vehicles, its application to micromobility, particularly e-scooters, remains largely unexplored. Recently, YOLO object detectors have gained significant attention due to continuous advancements, making them highly suitable for real-time detection tasks. In Redmon et al. [9], YOLO has also proven to be a practical approach for computer vision-based object detection in e-scooter applications.

In conclusion, detecting infrastructure obstacles, such as potholes and manhole covers, in front of e-scooters is crucial for enhancing rider safety. To the best of our knowledge, no prior study has specifically addressed ground obstacle detection for e-scooters. This research utilizes YOLO object detectors to identify roadway obstacles in real-time, focusing on six obstacle classes selected based on the significant vibrations they cause to riders, as measured by an IMU sensor. Furthermore, by integrating object detection results with depth data from the depth camera, the system is able to calculate the distance between the e-scooter and detected obstacles.

1.2 Research goal

Enhance the safety of e-scooter riders through automated object detection of infrastructure obstacles.

1.3 Research objectives and questions

1.3.1 Related work

Research questions:

1. What infrastructure obstacle datasets are available?
2. What are current object detection algorithms?
3. What are the gaps in knowledge about e-scooter safety?

1.3.2 Create a ground obstacle dataset tailored to e-scooters

Research questions:

1. How can we classify infrastructure obstacles that affect e-scooters?
2. How much data is needed to be collected for the dataset?

1.3.3 Identify object detection model for e-scooter road obstacle dataset

Research questions:

1. What is the suitable object detection model?
2. What are evaluation metrics?

1.3.4 Develop a real-time ground obstacle detection system for e-scooters

Research questions:

1. What is the real-time evaluation metric for this system?
2. What is the system's detection distance?

1.4 Research contributions

This study introduces a novel real-time infrastructure obstacle detection system for e-scooters. Six categories of infrastructure obstacles that generate strong vibrations, as measured by an IMU, are selected. The e-scooter infrastructure obstacle dataset is constructed from several hours of natural riding. The system employs the YOLO object detection module to identify obstacles in

RGB images. Additionally, by fusing depth and RGB data, it calculates the distance between the e-scooter and detected obstacles.

Chapter 2

Related Work

2.1 Infrastructure obstacle datasets

Pinggera et al. introduced the Lost and Found dataset, designed to identify small hazards and lost cargo on roads for autonomous vehicles [10]. This dataset comprises 2,104 annotated frames extracted from over 100 videos, featuring various objects, including a bobby car, a front bumper, a cardboard box, a tire, and square timber. Figure 2.1 illustrates the object classes in the Lost and Found dataset.

Figure 2.1: Collected Objects in the Lost and Found dataset [10]

Arya et al. developed the Road Damage Dataset (RDD2022) [11], an extension of RDD2018 [12], with incremental expansions in RDD2019 [13], RDD2020 [14], and RDD2022 [11]. Figure 2.2 illustrates the dataset's progression from RDD2018 to RDD2022. It comprises 47,420 images captured using smartphones, cameras, and Google Street View. Data collection involved vehicles such as cars, motorbikes, and drones. This dataset includes roads from multiple countries, including Japan, India, the Czech Republic, Norway, the United States, and China. It categorizes road damage into eight classes: longitudinal crack, lateral crack, alligator crack, rutting, bump, pothole, separation, crosswalk blur, and white line blur.

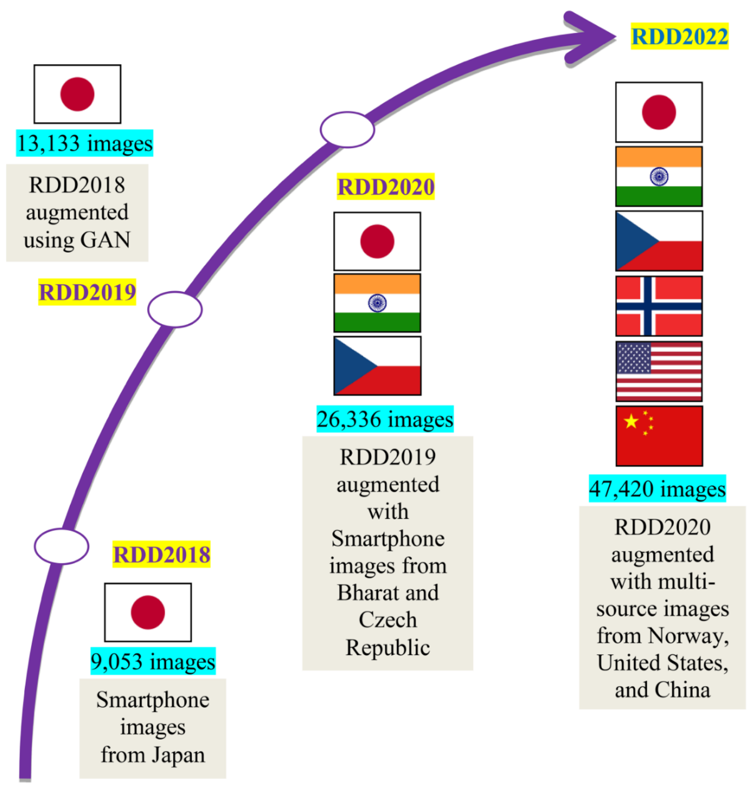
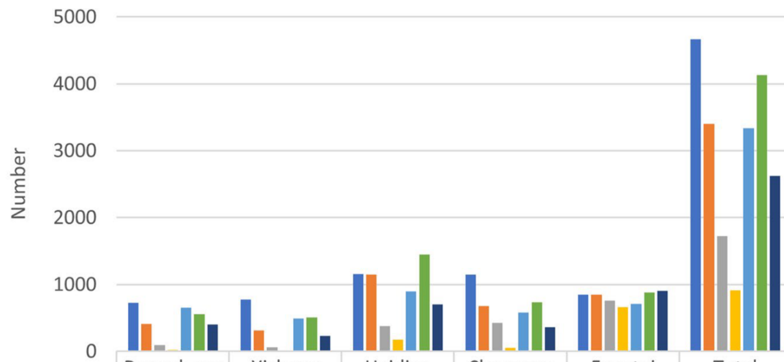


Figure 2.2: Development Progress of RDD from RDD2018 to RDD2022 [11]

Ren et al. developed the Street View Image Dataset for Automated Road Damage Detection (SVRDD) [15], consisting of 8,000 street-view images from Beijing's five administrative districts, sourced from Baidu Maps, with over 20,000 annotated instances. This dataset defines seven damage classes: longitudinal crack, transverse crack, alligator crack, pothole, manhole cover, longitudinal patch, and transverse patch. Figure 2.3 presents the distribution of instances across these categories.



	Dongcheng	Xicheng	Haidian	Chaoyang	Fengtai	Total
Longitudinal Crack	731	775	1160	1147	852	4665
Transverse Crack	415	311	1153	677	848	3404
Alligator Crack	95	64	378	428	763	1728
Pothole	20	5	173	55	665	918
Manhole Cover	654	496	896	578	715	3339
Longitudinal Patch	557	505	1447	734	885	4128
Transverse Patch	407	237	706	364	908	2622
Total #instances	2879	2393	5913	3983	5636	20804
Total #images	1000	1000	2000	2000	2000	8000

Figure 2.3: Distribution of Instances in the SVRDD Dataset [15]

The Lost and Found dataset, RDD2022, and SVRDD provide available infrastructure obstacle datasets, addressing the research question: What infrastructure obstacle datasets are available?

2.2 Object detection models

Object detection plays a crucial role in computer vision by simultaneously identifying and localizing objects within an image. Unlike image classification, which only determines the presence of an object, object detection provides its precise location. Additionally, compared to instance segmentation, object detection generally offers superior real-time performance. This technique strikes a balance between localization accuracy and computational efficiency, making it a practical choice for various applications.

Advancements in deep learning have significantly improved object detection. Object detection algorithms are generally categorized into two types: one-stage and two-stage detectors. Figure 2.4 illustrates the structures of both detector types.

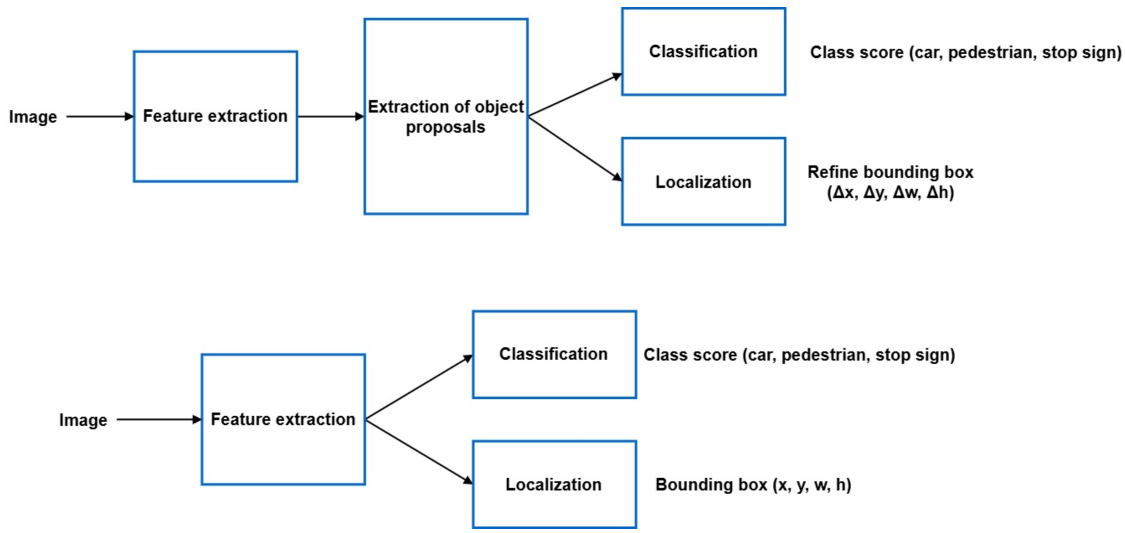


Figure 2.4: Two-Stage and One-Stage Object Detectors

In a two-stage detector, the input image undergoes feature extraction, producing a feature map. Based on this feature map, the detector generates multiple regions of interest (ROIs). Subsequently, it predicts the category of each ROI, such as a car, pedestrian, or stop sign, while refining the location and size using bounding box regression. A well-known example of a two-stage detector is Faster R-CNN [16], which integrates object detection into a unified network. The Region Proposal Network (RPN) enhances detection by focusing on relevant regions. Figure 2.5 illustrates the structure of Faster R-CNN.

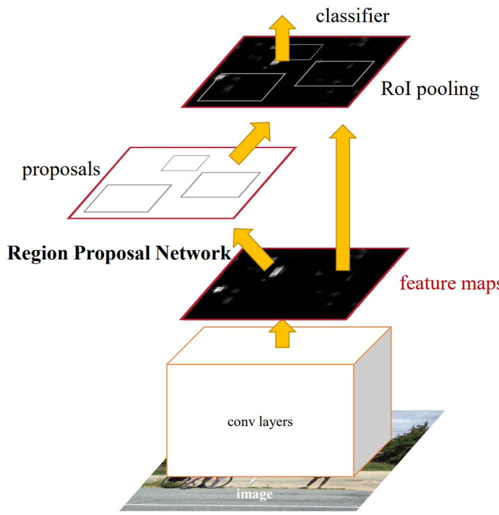


Figure 2.5: Structure of Faster R-CNN [16]

The one-stage detector processes an input image by extracting features and generating a feature map. Unlike a two-stage detector, which extracts object proposals, a one-stage detector directly predicts object categories and bounding box coordinates from the feature map. The YOLO (You Only Look Once) family represents a widely used one-stage detection approach, originally introduced by Redmon et al [9]. Figure 2.6 illustrates the YOLO detection system proposed by Redmon et al. The YOLO family has since expanded to include multiple versions, such as YOLOv3 [17], YOLOv5 [18], YOLOv6 [19], YOLOv7 [20], YOLOv8 [21], YOLOv9 [22], YOLOv10 [23], and YOLOv11 [24].

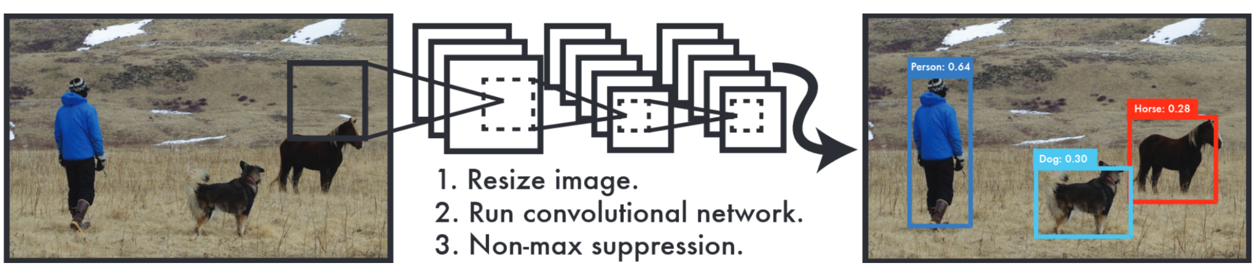


Figure 2.6: The YOLO Detection System [9]

Two-stage detectors achieve high accuracy due to their object proposal extraction but have lower real-time performance than one-stage detectors. In contrast, one-stage detectors offer faster inference but at the cost of reduced detection accuracy compared to two-stage detectors.

Two-stage and one-stage detectors represent the main categories of object detection models, addressing the research question: What are the current object detection algorithms?

2.3 The absence of an infrastructure obstacle dataset and a relevant detection system for e-scooters

Wenzelburger et al. [25] proposed a self-balancing mechanism for standard e-scooters. They modeled a combined system of an electric scooter and flywheel using reaction wheels to generate torque and designed a cascade-like controller to enable the scooter to remain upright and withstand minor disturbances. Tests demonstrated that the electric scooter equipped with this self-balancing mechanism could maintain balance, operate, and resist push pressure on the handle. In a related study, Soloperto et al. proposed a control framework for autonomous electric scooters [26]. The framework utilized a reaction wheel and braking mechanism to achieve self-balancing of the electric scooter. An RGB-Depth camera combined with the Random Sample Consensus (RANSAC) algorithm detected the surrounding environment. A vector field approach determined the scooter's movement direction to avoid obstacles. The control framework enabled the electric scooter to maintain self-stabilization, detect obstacles, and avoid collisions.

An intelligent multi-wheeled scooter was designed by Lin et al. [27]. The image sensor captures image data, and the antenna's elevation is controlled to adjust the image to the optimal viewing angle using Web Input Output Pi (WebIOPi) to adjust general-purpose input/output (GPIO) pins. The user's health status is monitored via a smartwatch, and their current location is tracked using GPS. A Raspberry Pi 2 and a wireless network card were used to construct the server and establish network connectivity, creating a smart electric scooter network for monitoring environmental information, health status, and location. Experimental results demonstrated that the system is both practical and accurate, making it suitable for remote monitoring of multi-wheeled scooter users' safety. Alai et al. [28] proposed an active perception and rear vehicle tracking collision avoidance system for electric scooters. The system employs a low-cost single-beam laser sensor with active directional control via a stepper motor to detect and track rear vehicles. A density-based spatial clustering (DBSCAN) algorithm is used for vehicle detection, and a nonlinear bicycle model is employed for vehicle motion modeling. A nonlinear observer is designed to estimate vehicle states, while a receding horizon controller dynamically adjusts the laser sensor's direction to continuously track the vehicle. To track the trajectory of target vehicles, specifically e-scooters, Alai et al. [29] proposed a multi-stage estimation algorithm. They transformed the kinematic model of a non-ego vehicle's motion into three separate models addressing velocity, direction, and position, and designed an observer for each model. The estimation algorithm outperformed previous observers in both simulations and experiments, demonstrating stability and robustness to uncertainty while maintaining low computational resource requirements. This algorithm is valuable for low-cost, sensor-based collision prediction and avoidance applications. Alai et al. [30] developed an active sensing and assessment system designed to protect electric scooter riders from collisions with cars and other scooters. The system utilized a low-cost single-beam laser sensor mounted on a stepper motor to detect vehicles behind the scooter. When a rear vehicle was detected, a nonlinear vehicle model and observer estimated the trajectory variables of the tracked vehicle. Based on these estimates, a recursive controller adjusted the laser sensor's real-time position to track the rear vehicle continuously. The entire system is integrated into a prototype electric scooter, with a Teensy microprocessor implementing all algorithms. Extensive conventional and specialized traffic experiments were conducted, confirming the system's reliability in accurately tracking the trajectory of the rear vehicle across all tested scenarios. However, the system tracks only one rear vehicle at a time without considering multiple vehicle tracking. Yousuf et al. [31] developed a self-driving scooter designed to alleviate traffic congestion on campus. The scooter is equipped with an NVIDIA Jetson, Arduino, LiDAR, motor driver, battery, LED touchscreen, GPS, and an Anker mobile power supply. Additionally, an app was created to facilitate user access to the scooter. A survey was conducted among students and the general public to better understand user needs. The system demonstrated an average error of 4.8% in reaching the final destination while achieving 100% accuracy in obstacle detection. Strässer et al. [32] proposed a collision avoidance safety filter for autonomous electric scooters. The filter is designed to mitigate the effects of noise and occasional missing distance measurements when using low-cost ultrasonic sensors to detect obstacles ahead. Additionally, a gain scheduling controller was developed to reduce the scooter's speed and prevent collisions when an imminent threat is detected based on the distance between the scooter and the obstacle. Three types of experiments were conducted: straight travel and turning with static obstacles and straight travel with dynamic obstacles. The results demonstrated that the safety filter effectively prevents collisions. Yan et al. [33] introduced a novel collision avoidance system for electric scooters. The system employs a pedestrian trajectory prediction model that integrates a Long Short-Term Memory (LSTM) network with a State Refinement (SR) Module and a motion planner based on gradient descent algorithms, incorporating artificial potential fields and energy surfaces. The pedestrian trajectory prediction model accounts for human interactions by analyzing past pedestrian locations and the behavior and positions of surrounding individuals to predict future trajectories. Using these predicted trajectories, the motion planner designs a collision-free path for the electric scooter. The pedestrian trajectory prediction model was validated on two public datasets, ETH and UCY, achieving low mean average displacement (MAD) and final average displacement (FAD) errors of 0.71 m and 1.46 m, respectively. Experimental testing of the collision avoidance system in two representative scenarios—parallel and cross-pedestrian movement—demonstrated its effectiveness in reducing potential collisions between electric scooters and pedestrians.

Several studies have explored object detection for e-scooters. Chen et al. [34] established a comprehensive benchmark evaluating 22 selected YOLO object detectors designed for e-scooter-based applications. They constructed a traffic scene dataset specifically for e-scooters and assessed YOLO models ranging from YOLOv5 to YOLOv8. The evaluation considered detection accuracy, model complexity, and inference time. The results indicated that YOLOv5s provides the best balance between efficiency and effectiveness. Furthermore, Li [35] proposed an algorithm combining depth estimates from a single camera with detection results from the YOLOv3 model to identify obstacles around the scooter. The system issues corresponding prompts based on obstacle distance, achieving a detection accuracy of approximately 70%.

Existing studies on e-scooter safety focus on stability and self-balancing, collision prevention, object detection and perception, and autonomous operation. However, no research specifically addresses infrastructure obstacle detection for e-scooters. This gap in the literature highlights the research question: What are the unresolved challenges in e-scooter safety?

Chapter 3

E-scooters infrastructure obstacle dataset

3.1 Linear vertical acceleration

The Intel RealSense Camera D435i features an RGB camera, a depth camera, and an IMU, which was mounted on the e-scooter. The mount's angle was adjusted prior to data collection. Figure 3.1 illustrates the installation of the camera on the Ninebot KickScooter MAX G30LP e-scooter.

Figure 3.1: Camera Mounted on the E-Scooter

The selection of roadway obstacle classes draws on findings from prior naturalistic e-scooter riding studies [5] and survey-based research highlighted ground conditions as a significant risk factor for riders [4, 36]. Moreover, the e-scooter's road vibrations are correlated with linear vertical acceleration, which is measured using an accelerometer in IMU [37]. Linear vertical acceleration effectively reflects road surface quality. The accelerometer first collects three-axis acceleration data, including components along the X, Y, and Z axes. Figure 3.2 illustrates the three axes of the accelerometer within the camera mounted on the e-scooter. Since gravity primarily affects the Y and Z axes, a first-order low-pass filter suppresses high-frequency fluctuations caused by motion, enabling the estimation of gravity components along these axes. The system then subtracts the estimated gravity components from the measured acceleration values to determine the scooter's linear acceleration in the Y and Z directions. To quantify the motion intensity in these directions, the Euclidean norm of the linear acceleration in the Y- Z plane is calculated, representing the scooter's vertical linear acceleration. Equation 3.1 shows the calculation of the linear vertical acceleration.

$$\text{Linear\_vertical\_accel} = \sqrt{\text{Linear\_accel}_y^2 + \text{Linear\_accel}_z^2} \tag{3.1}$$

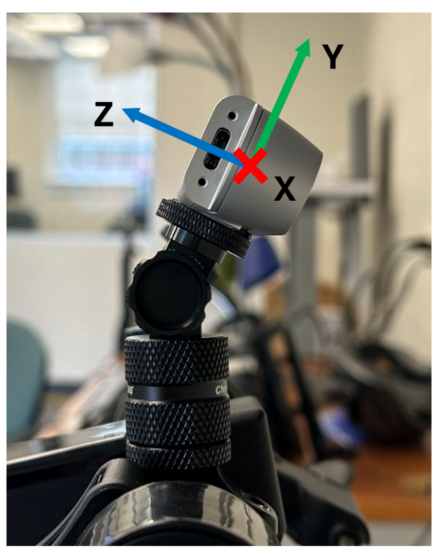


Figure 3.2: Three Axes of the Accelerometer Within the Camera Mounted on the E-Scooter

Experiments were conducted to focus on selecting road hazard classes based on the linear vertical acceleration data from the IMU. Classes that do not cause significant vibrations for e-scooters, such as weeds, were excluded. Figure 3.3 present one example of a non-directional crack and corresponding vibration. Figure 3.4 illustrates one example of a directional crack and corresponding vibration.

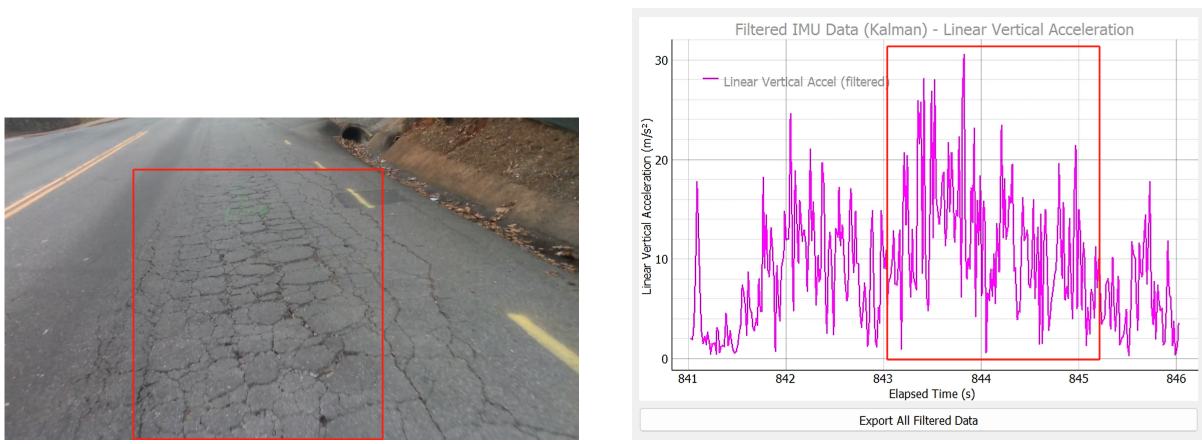


Figure 3.3: One Example of a Non-Directional Crack and the Corresponding Vibration of the E-Scooter

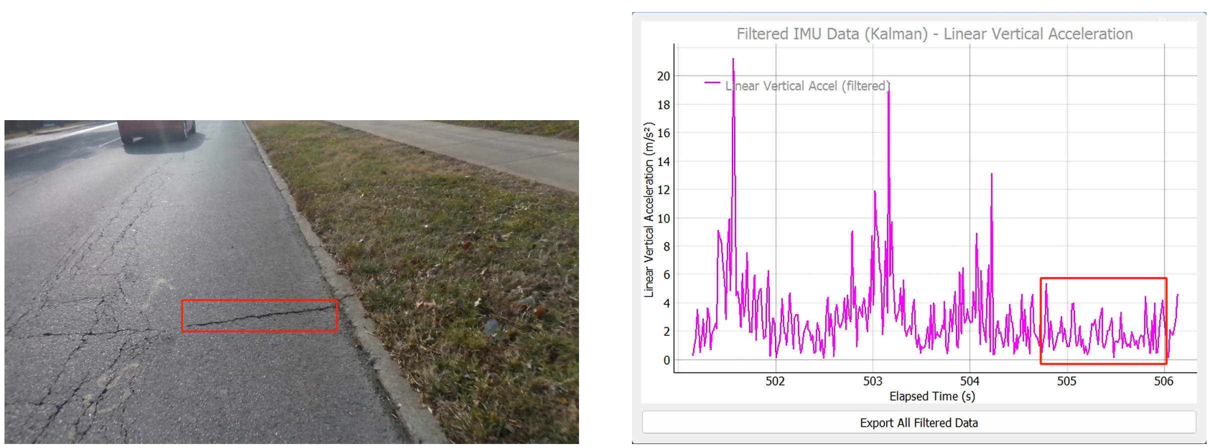


Figure 3.4: One Example of a Directional Crack and the Corresponding Vibration of the E- Scooter

Finally, the classes selected include manhole covers, non-directional cracks, pinecones, potholes, tree branches, and truncated domes. Figure 3.5 provides examples of each class.

Figure 3.5: Examples of Six Classes of Infrastructure Obstacles

Linear vertical acceleration data quantifies e-scooter vibrations on various surfaces, aiding in the classification of infrastructure obstacles that negatively impact e-scooter performance. This analysis addresses the research question: How can we classify infrastructure obstacles that affect e-scooters?

3.2 Data collection

Data collection occurred during natural riding sessions at various daytime hours near the University of Virginia, constrained by the e-scooter's limited mobility, resulting in over seven hours of recorded video footage. The videos were converted into images, and relevant frames were extracted based on classes. Label Studio (Tkachenko et al. 2020) was used to annotate bounding boxes for each class. The resulting dataset comprises 3,427 images categorized into six classes. These road hazards are characterized by highly visible objects rather than less noticeable ones. The dataset includes a total of 8,864 bounding box annotations, categorized as follows: manhole covers (1,050), non-directional cracks (1,310), pinecones (2,866), potholes (1,069), tree branches

(1,814), and truncated domes (755). Figure 3.6 presents the frequency of bounding box annotations by obstacle type, while Figure 3.7 depicts their proportion.

The e-scooter's limited mobility restricts the dataset to 3,427 images and 8,864 bounding box annotations. This scope answers the research question: How much data is needed to be collected for the dataset?

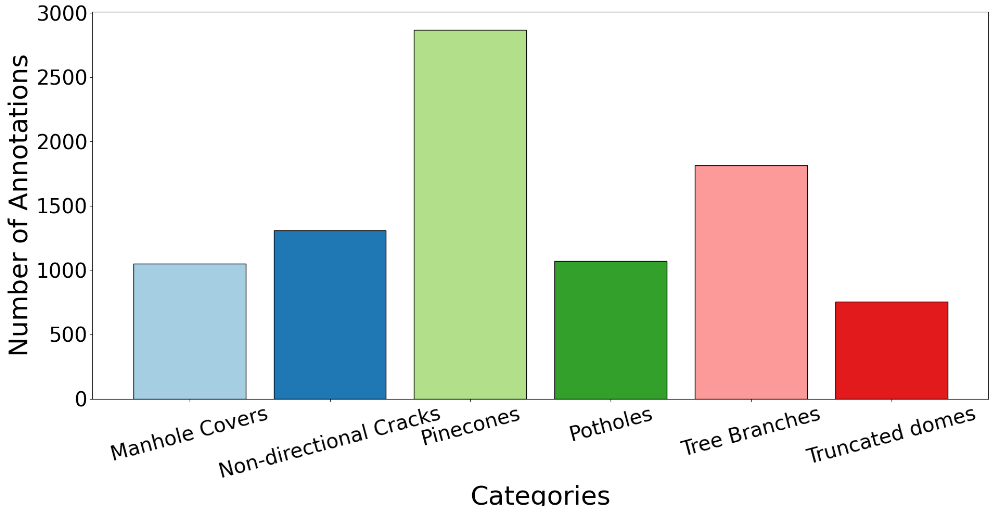


Figure 3.6: The Frequency of Bounding Box Annotations by Infrastructure Obstacle Type

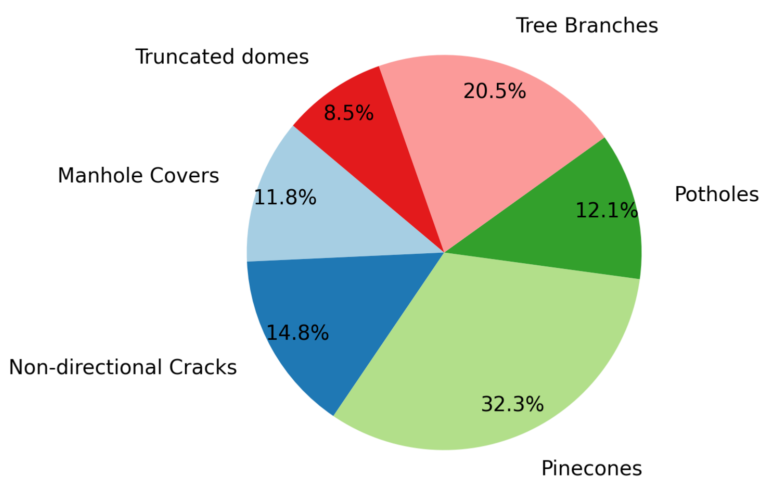


Figure 3.7: The Proportion of Bounding Box Annotations by Infrastructure Obstacle Type

Chapter 4

Object detection model for e-scooter infrastructure obstacle dataset

4.1 YOLO detectors

Due to the limited computational resources of embedded devices in e-scooters, a one-stage detector is a more suitable choice [34]. Among such detectors, the YOLO family is the most widely adopted. Since its introduction by Redmon et al., YOLO has undergone continuous development, resulting in successive versions, including YOLOv3 [17], YOLOv5 [18], YOLOv6 [19], YOLOv7 [20], YOLOv8 [21], YOLOv9 [22], YOLOv10 [23], and YOLOv11 [24]. From YOLOv3 onward, the YOLO family adopts a modular Backbone-Neck-Head architecture, as illustrated in Figure 4.1.

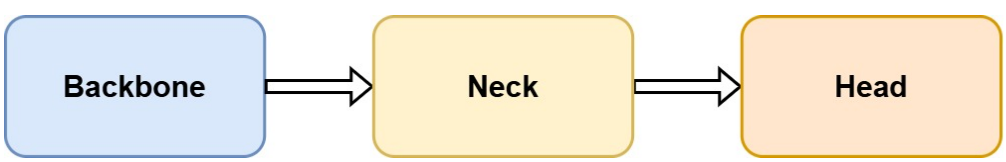


Figure 4.1: High-Level Architecture of YOLO Family

The YOLO architecture consists of three modules: the backbone, the neck, and the head, each with a distinct role. The backbone extracts features from the input image and generates a feature map. The neck enhances this feature map by integrating multi-scale features before passing it to the head. The head performs object detection by predicting bounding boxes, confidence scores, and category classifications.

Prior research by [34] established that YOLOv5s achieves optimal performance for e-scooter applications. Additionally, comparisons of newer efficient YOLO models, similar in size and complexity to YOLOv5s, are presented in the results and discussion section. This part introduces the architecture of YOLOv5s. Figure 4.2 presents the structure of the YOLOv5s network model. The Backbone forms the foundational component of the network, extracting essential features from input images. The Neck upsamples high-level feature maps and fuses them with low-level feature maps, enhancing semantic richness and capturing small object details. Additionally, low-level feature maps are down-sampled and merged with high-level feature maps, refining detailed information. This bidirectional feature fusion strengthens the network's capability to detect objects across different scales. Finally, the Head processes feature maps of high, medium, and low resolutions from the Neck, enabling the detection of small, medium, and large objects, respectively.

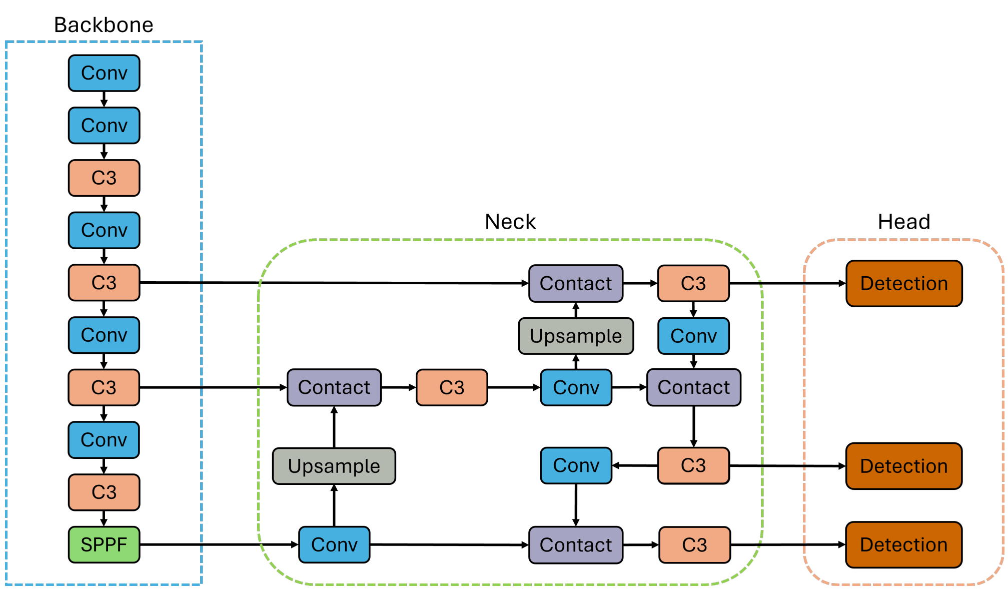


Figure 4.2: The Structure of YOLOv5s

YOLO detectors address the research question: What is the most suitable object detection model?

4.2 Precision, Recall, mAP50, Model Size, and GFLOPs

Precision (P) measures the proportion of true positives among all detected ground obstacles. Recall (R) reflects the model's ability to identify all target classes in the image. The equations for precision and recall are given in Equations 4.1 and 4.2, respectively.

$$P = \frac{TP}{TP + FP} \tag{4.1}$$

$$R = \frac{TP}{TP + FN} \tag{4.2}$$

Here, True Positives (TP) refer to correctly detected targets, False Positives (FP) represent incorrect detections, and False Negatives (FN) indicate missed detections.

A higher precision (P) signifies a greater proportion of correctly identified targets with fewer false positives, while a higher recall (R) indicates a higher detection rate with fewer missed targets.

The mean Average Precision at IoU 0.5 (mAP_50) quantifies the average precision when the intersection-over-union (IoU) threshold is set to 0.50. The IoU measures the ratio of the intersection to the union between the predicted and labeled bounding boxes. Figure 4.3 illustrates this concept.

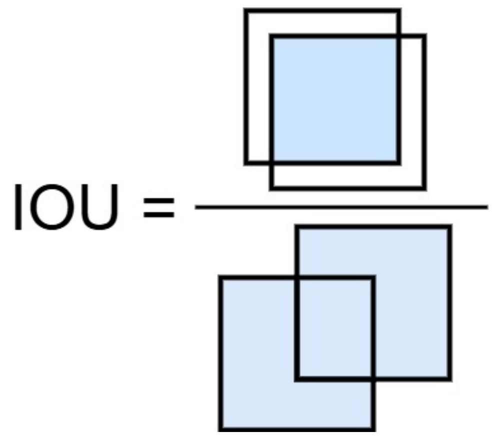


Figure 4.3: Visualization of Intersection over Union (IoU) in Object Detection

The mAP_0.5:0.95 (mAP50-95) calculates the average precision across varying IoU thresholds, from 0.50 to 0.95. These metrics range from 0 to 1, with values closer to 1 indicating superior performance. Compared to mAP50-95, mAP50 is more practical as it focuses more on detecting target objects [38]. Furthermore, mAP50 thoroughly evaluates a model's performance by integrating both P and R.

Model size measures the number of parameters in a model; a smaller model size requires less storage. Giga Floating-point Operations (GFLOPs) serve as a critical metric for evaluating the computational resources required by YOLO models. Lower GFLOPs count signifies reduced computational demands and improved real-time performance.

Precision, recall, mAP50, model size, and GFLOPs define the evaluation metrics, addressing the research question: What are the evaluation metrics?

Chapter 5

Real-time infrastructure obstacle detection system for e-scooters

Figure 5.1 illustrates the complete framework of the infrastructure obstacle detection system for e-scooters. The system utilizes the YOLOv5s model to detect ground obstacles and combines RGB and depth images from the Intel RealSense Camera D435i to estimate obstacle distances in real-time.

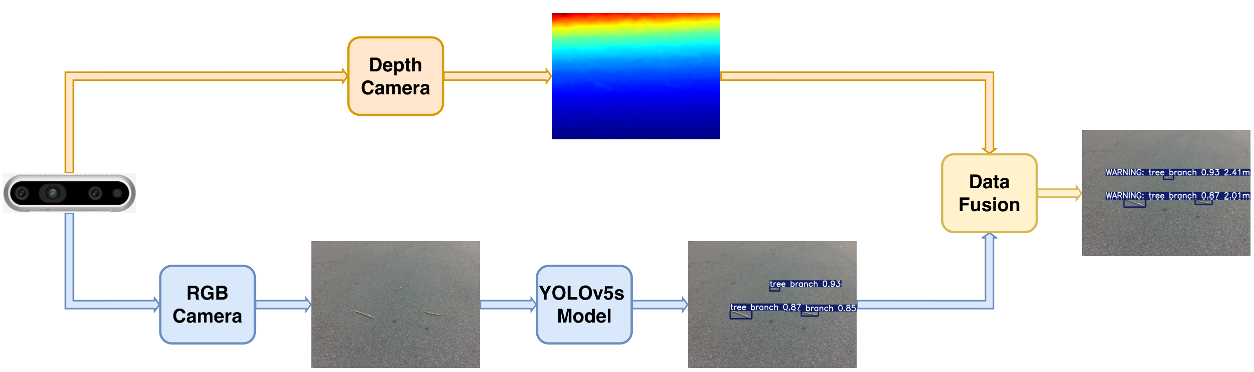


Figure 5.1: Framework of infrastructure obstacle detection system for e-scooters

RGB and depth images are initially aligned using coordinate transformations to ensure that color and depth information for each pixel resides in the same coordinate space. This alignment ensures precise mapping of object depth values to their corresponding color data. After performing target detection on the aligned RGB image using YOLOv5s, the system extracts the bounding box and category information of the detected object. The bounding box provides the coordinates of the object's center point. Due to the presence of noise in the depth data, the system performs random multipoint sampling around the center point to obtain multiple depth values. These values are sorted, and the middle section is selected to calculate the average depth value. This average depth value is then integrated into the detection result, which includes the object category and confidence level. The system demonstrates its ability to notify the rider by using this distance to display real-time warning texts on the detection image. The system outputs a warning text signal on the detection result when the distance falls to 4 meters or less, accounting for the e-scooter's speed and the camera's position on the e-scooter.

5.1 Inference time

Inference time is a critical metric for evaluating the real-time performance of this system. Multiple factors influence inference time, including camera data acquisition, YOLOv5s model inference, and data fusion. Among these, YOLOv5s model inference is significant, as it demands substantial computational resources.

Inference time addresses the research question: what is the real-time evaluation metric for this system?

5.2 Capable of detecting objects within 10 meters

The Intel RealSense D435i depth camera has a maximum detection range of 10 meters. Given a camera angle of approximately 30°, the effective detection range of the system is about 8.66 meters. Figure 5.2 illustrates this maximum detection distance.

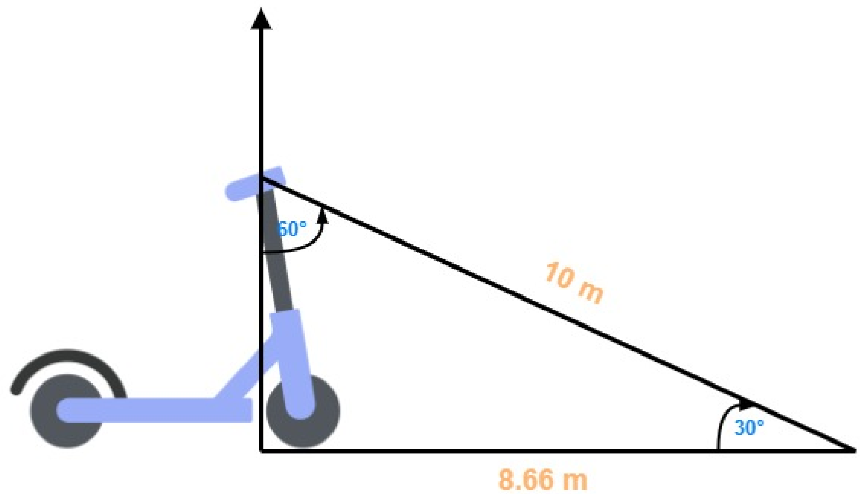


Figure 5.2: Illustration of the Maximum Detection Distance of the Intel RealSense D435i Camera

Detecting infrastructure obstacles within a 10-meter range addresses the research question: What is the system's detection distance?

Chapter 6

Result

To train the YOLOv5s model using the collected data, the input image resolution is set to 1280x1280, batch size is set to 64, and the model is trained for 100 epochs. The dataset was split into training, validation, and test sets in a 7:2:1 ratio. Training took place on a multi-GPU server provided by the University of Virginia Research Computing. The batch size and number of epochs are dynamically adjusted if GPU memory constraints arise. The same trajectory is used to train same-level models from YOLOv6 to YOLOv11 to compare the performance and efficiency.

Figure 6.1 presents the YOLOv5s loss function curves for training and validation phases, alongside the performance metrics on the validation dataset across successive training epochs, adhering to the page limit. Box loss (box_loss) quantifies the disparity between the predicted and actual bounding boxes. Object loss (obj_loss) measures how well the predicted bounding box overlaps the target object. Classification loss (cls_loss) evaluates the model's accuracy in recognizing target classes within the image. The lower value of these losses indicates better performance. The loss curves demonstrate an overall downward trend for the training and validation datasets, while the performance metrics for the validation dataset show steady improvement throughout the training period.

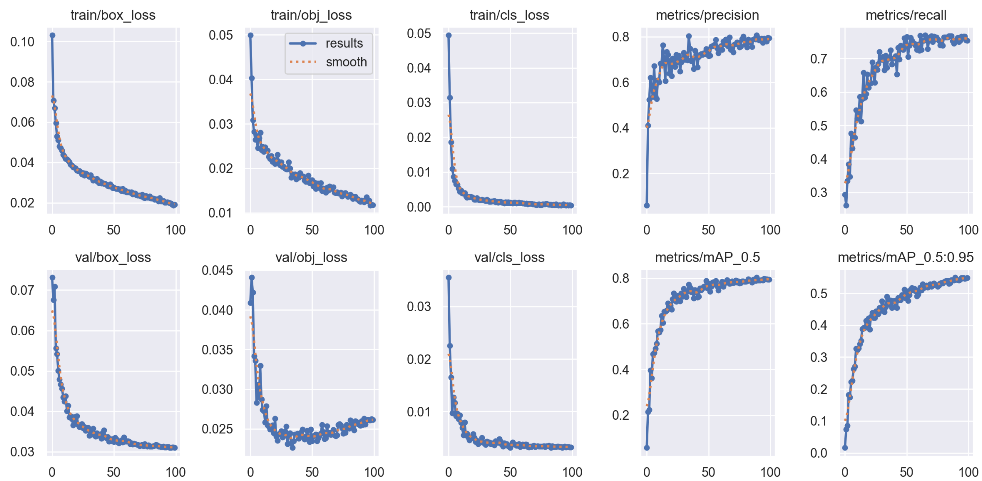


Figure 6.1: Training and validation results of YOLOv5s

Table 6.1 compares the performance metrics, model sizes, and GFLOPs of various YOLO models at the same level on the testing dataset. Model size measures the number of parameters in a model; a smaller model size requires less storage. Giga Floating-point Operations (GFLOPs) serve as a critical metric for evaluating the computational resources required by YOLO models. Lower GFLOPs count signifies reduced computational demands and improved real-time performance. Table 1 shows that the YOLO models demonstrate comparable performance. YOLOv11s achieves the highest precision (P) of 0.838, YOLOv5s records the highest recall (R) of 0.799, and YOLOv9s attains the highest mAP50 of 0.84. Additionally, YOLOv5s has the smallest model size (14 MB) and the lowest computational requirement at 15.8 GFLOPs, which is approximately 74% of the second-lowest model (YOLOv11s) and only 8% of the highest (YOLOv7x).

Table 6.2 highlights the performance of each class detected by the YOLOv5s model on the testing dataset. The truncated dome achieves the highest metrics (P: 0.952, R: 0.983, and mAP50: 0.985). In contrast, the tree branch records the lowest P (0.732), while the non-directional crack exhibits the lowest R (0.556) and mAP50 (0.649).

Table 6.1: Performance comparison of different YOLO models

YOLO models	P	R	mAP50	Model size (MB)	GFLOPs
YOLOv5s	0.834	**0.799**	0.827	**14**	**15.8**
YOLOv6s	0.835	0.756	0.816	38.8	45.3
YOLOv7x	0.809	0.778	0.807	136	188.1
YOLOv8s	0.819	0.771	0.820	21.5	28.4
YOLOv9s	0.834	0.788	**0.840**	14.6	26.7
YOLOv10s	0.817	0.776	0.826	15.8	24.5
YOLOv11s	**0.838**	0.781	0.831	18.4	21.3

Table 6.2: Performance by class of YOLOv5s on the testing Dataset

Class	Images	Instances	P	R	mAP50
All	343	831	0.834	0.799	0.827
Manhole cover	99	119	0.891	0.866	0.917
Non-directional crack	69	143	0.776	0.556	0.649
Pine cone	65	220	0.837	0.884	0.894
Pothole	70	124	0.816	0.694	0.725
Tree branch	93	165	0.732	0.812	0.792
Truncated dome	52	60	0.952	0.983	0.985

A personal laptop with an Intel i5-13500H processor, NVIDIA GeForce RTX 3050 GPU (6GB), 16GB of RAM, and Windows 11 serves as the test device. Inference size is configured to 640×480 to better focus on the road ahead of e-scooters. Real-world experiments show that YOLOv5s achieves an inference time between 9ms and 10 ms. The total inference time of the system is between 10 to 20 ms. In comparison, human drivers require approximately 1600–2300 ms to avoid obstacles [39]. Figure 6.2 shows the system's detection results for ground obstacles positioned beyond and within 4 meters of the electric scooter.

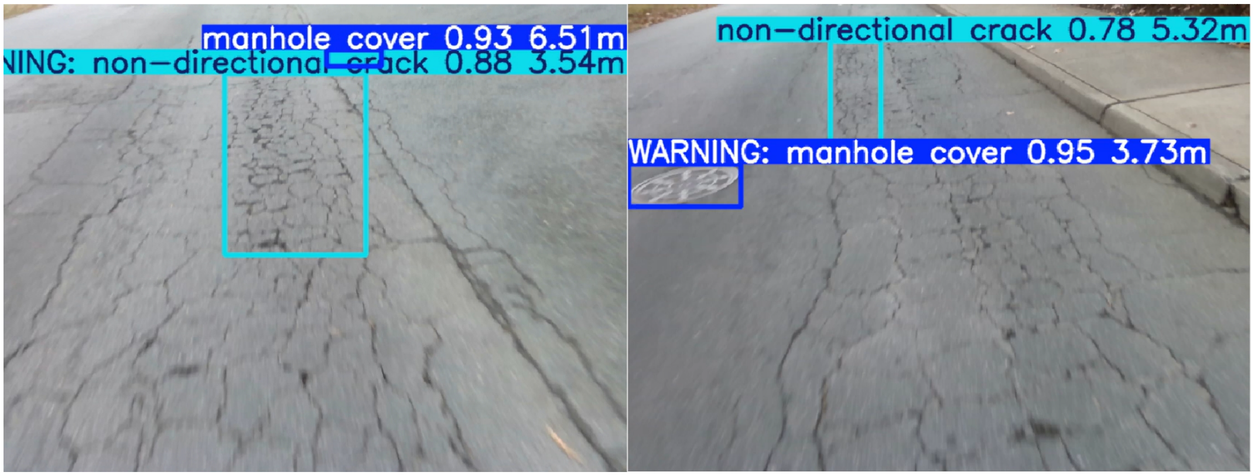


Figure 6.2: Examples of the system detection results

Chapter 7

Discussion

Compared to other same-level YOLO models, YOLOv5s achieves superior computational and storage efficiency while maintaining similar performance. These results show that YOLOv5s is better suited for e-scooters, which are devices with limited computing resources. In the future, deploying the trained YOLOv5s model on an embedded device in e-scooters will be both feasible and efficient.

A comprehensive analysis of YOLOv5s metrics for each category on the test dataset reveals notable performance variations. The manhole covers and truncated dome categories achieve excellent mAP50 scores of 0.917 and 0.985, respectively, likely due to their large sizes and distinctive features, which make them easily recognizable. The pinecone and tree branch categories contain more instances than others, as the dataset emphasizes collecting smaller target objects. However, the tree branch's mAP50 (0.792) is significantly lower than that of the pinecone (0.894). The pothole category records a mAP50 of 0.725, while the non-directional crack has the lowest mAP50 of 0.649. For the tree branch, pothole, and non-directional crack categories, the low mAP50 values are likely due to varying shadows and light reflections, which make them harder to recognize [40]. Improving detection performance may involve collecting more diverse samples of these categories and refining the training methodology to address these challenges effectively. Additionally, expanding the dataset to include other categories, such as stones, could further benefit e-scooter riders.

The system demonstrates a robust ability to measure distances to detected objects and explores its potential to warn e-scooter riders by displaying real-time warning messages on data fusion images. It lays the groundwork for future studies on delivering alerts to e-scooter riders.

Chapter 8

Future Work

Collecting additional obstacle categories and incorporating diverse environmental conditions, particularly nighttime scenarios, is essential for improving model generalization. Collaboration with other universities can enhance data collection and further strengthen the dataset's robustness.

The system's infrastructure obstacle detection accuracy primarily depends on the YOLOv5s model, which achieves a mAP50 of 0.827. Although the system demonstrates high accuracy, further improvements are possible through model enhancement, advanced data augmentation, or the development of a new model. However, maintaining real-time performance remains crucial when optimizing detection accuracy.

Currently, the system runs on a personal desktop, making it impractical for deployment on an e-scooter. To enable real-world application, the model should be deployed on an AI-embedded board, such as the NVIDIA Jetson Orin Nano Developer Kit. Given the resource limitations of e-scooters, an embedded board provides a more suitable solution.

Enhancing this system also requires improving user alerts through multiple methods, such as auditory cues, smart augmented reality (AR) glasses, and haptic feedback. For instance, an auditory alert, such as a beeping sound, can notify users of detected obstacles. When using smart AR glasses, users can receive visual alerts about detected infrastructure obstacles within their field of view. However, these enhancements must balance effectiveness with user experience, as frequent alerts in real traffic conditions may cause discomfort or distraction.

Chapter 9

Conclusion

The increasing adoption of e-scooters in urban environments has led to a rise in e-scooter-related crashes and injuries. Due to design limitations, e-scooters are particularly susceptible to vibrations caused by road hazards, making rider safety on uneven roadways a pressing concern. To address this issue, this paper proposes a real-time ground obstacle detection system for e-scooters. The system integrates an RGB camera and a depth camera to accurately and efficiently detect six types of road obstacles, achieving an overall high mAP50 of 0.827 while maintaining computational cost-efficiency. The IMU measures vibrations through linear vertical acceleration, which informs the selection of these six obstacle types. These obstacles include manhole covers, non-directional cracks, and potholes, identified using YOLO object detection models. By fusing RGB data and depth data, the system facilitates precise distance estimation. However, further inclusion of additional road obstacle classes harmful to e-scooters is necessary, and the system's performance still has room for improvement. Future work could focus on deploying the system on embedded devices with real-time notification systems, such as voice alerts or dashboard displays. This research presents an effective real-time solution for detecting ground obstacles encountered by e-scooters, paving the way for advancements in smart e-scooter safety and contributing to the broader development of smart mobility.

Appendix A

Structure of YOLOv6 to YOLOv11

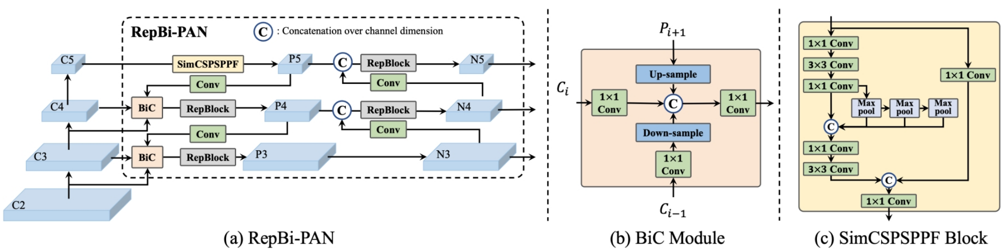


Figure A.1: (a) The neck of YOLOv6 (N and S are shown). Note for M/L, RepBlocks is replaced with CSPStackRep. (b) The structure of a BiC module. (c) A SimCSPSPPF block [19].

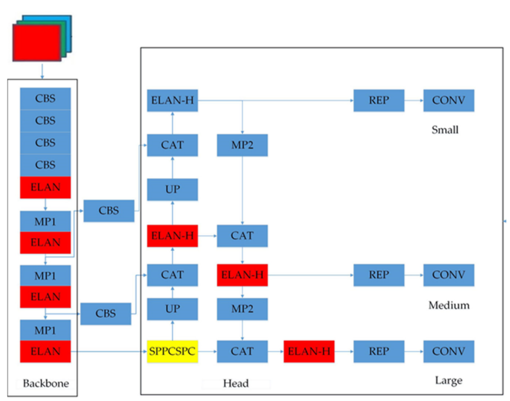


Figure A.2: The structure of YOLOv7 [41]

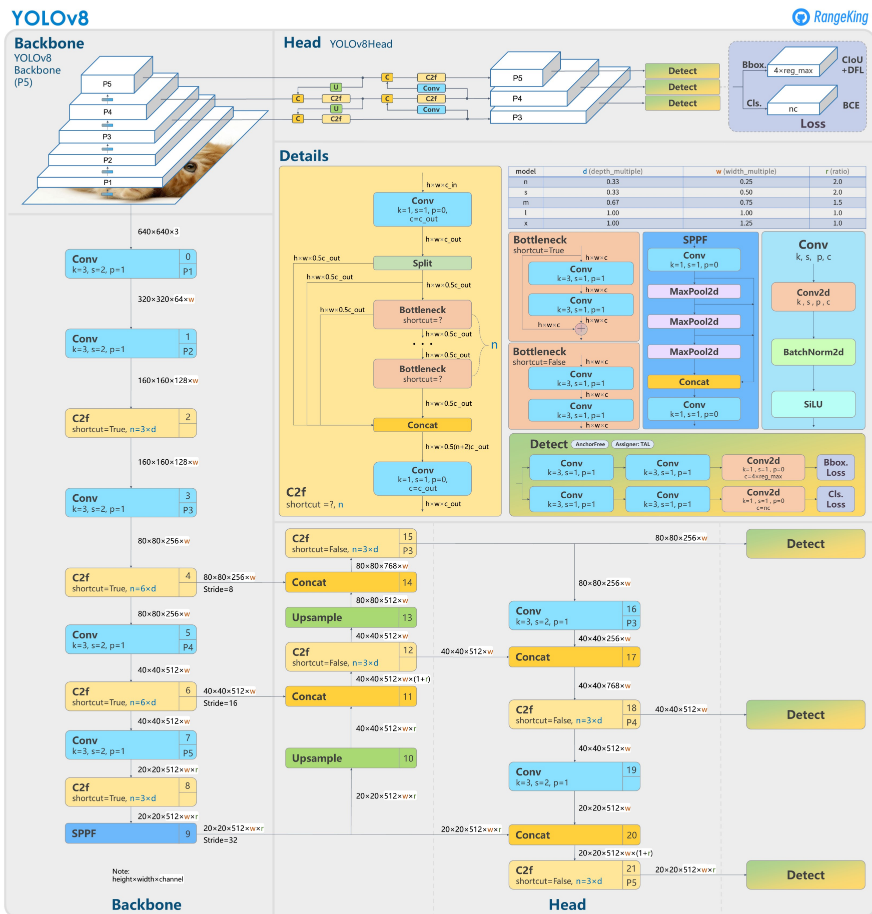


Figure A.3: The structure of YOLOv8 [42]

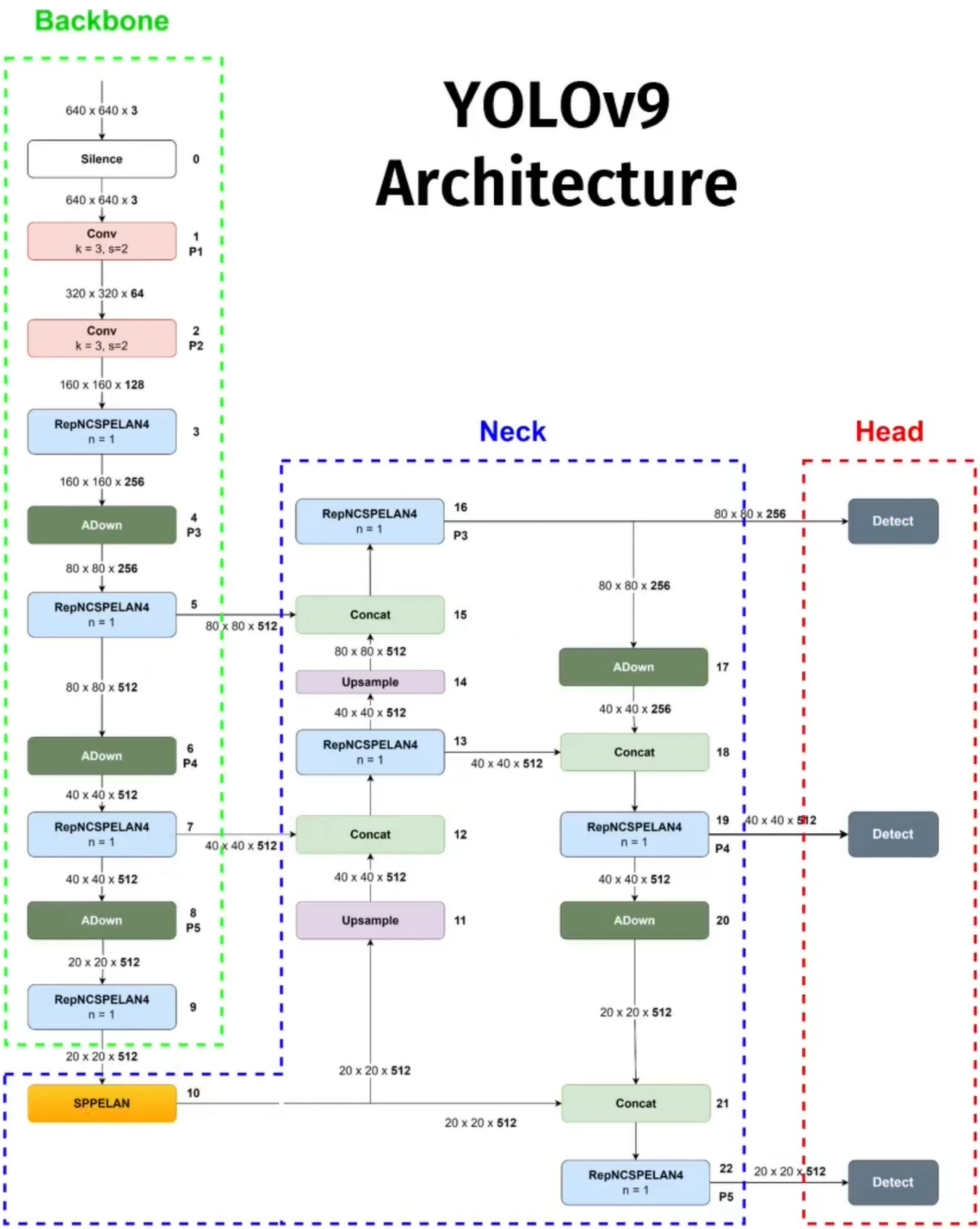


Figure A.4: The structure of YOLOv9 [22]

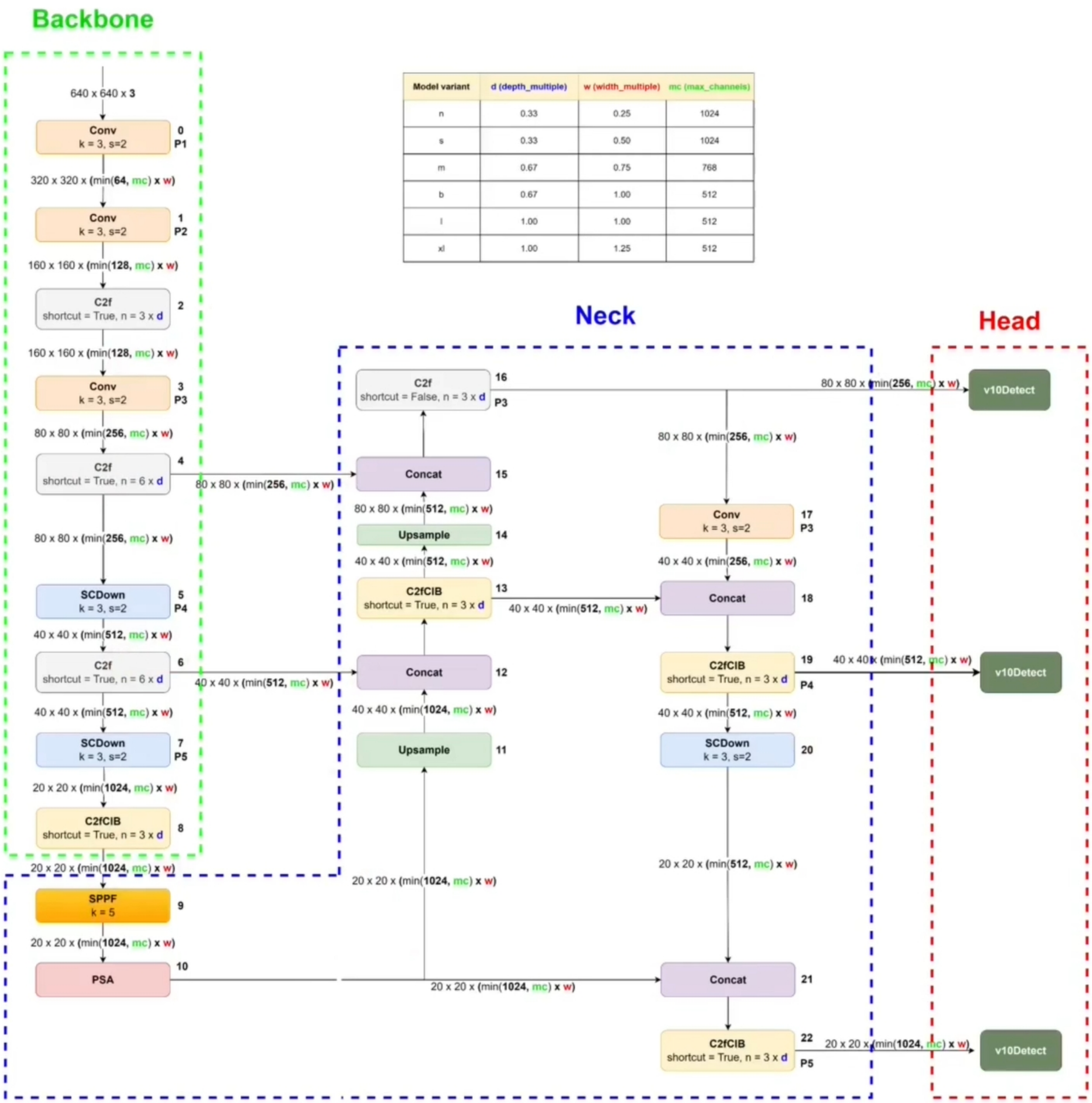


Figure A.5: The structure of YOLOv10 [43]

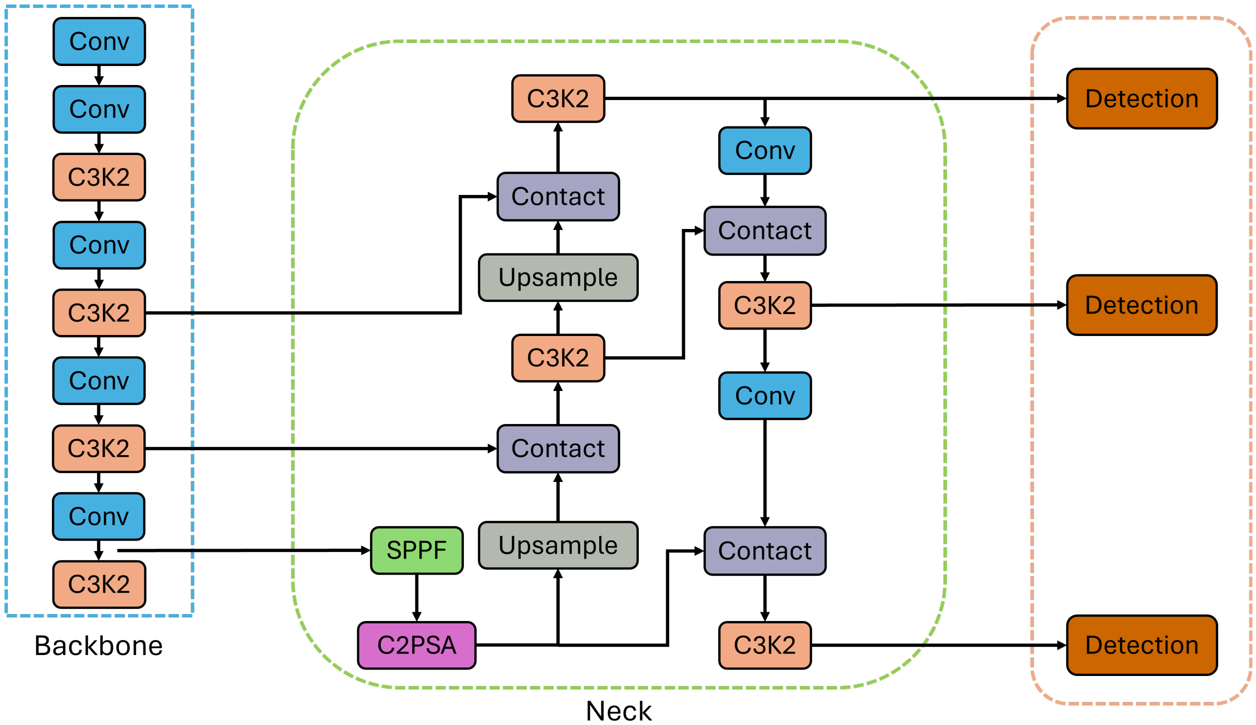


Figure A.6: The structure of YOLOv11

Glossary

Acronyms and Abbreviations	
CAGR	Compound Annual Growth Rate
ADAS	Advanced Driver Assistance System
LiDAR	Light Detection and Ranging
YOLO	You Only Look Once
IMU	Inertial Measurement Unit
RGB	Red, Green, Blue
RDD	Road Damage Dataset
SVRDD	Street View Image Dataset for Automated Road Damage Detection
ROIs	Regions of Interest
RPN	Region Proposal Network
RANSAC	Random Sample Consensus
WebIOPi	Web Input Output Pi
GPIO	General Purpose Input/Output
GPS	Global Positioning System
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
LED	Light Emitting Diode
LSTM	Long Short-Term Memory
SR	State Refinement
ETH	Federal Institute of Technology

UCY	University of Cyprus
MAD	Mean Average Displacement
FAD	Final Average Displacement
TP	True Positives
FP	False Positives
FN	False Negatives
P	Precision
R	Recall
IoU	Intersection over Union
GFLOPs	Giga Floating-point Operations
GPU	Graphics Processing Unit
UVA	University of Virginia

Bibliography

[1] "Electric Scooters Market Size & Share Analysis Report, 2030." [Online]. Available: https://www.grandviewresearch.com/industry-analysis/electric-scooters-market

[2] A. N. Fernandez, K. D. Li, H. V. Patel, I. E. Allen, U. Ghaffar, N. Hakam, and B. N. Breyer, "Injuries With Electric vs Conventional Scooters and Bicycles," *JAMA Network Open*, vol. 7, no. 7, p. e2424131, Jul. 2024. [Online]. Available: https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2821387

[3] M. Paudel, F. F. Yap, and A. K. Bastola, "Safety assessment of personal mobility devices with different wheel size based on their dynamic stability performance," *International Journal of Sustainable Design*, vol. 3, no. 4, p. 227, 2020. [Online]. Available: http://www.inderscience.com/link.php?id=112094

[4] K. Sievert, M. Roen, C. M. Craig, and N. L. Morris, "A Survey of Electric-Scooter Riders' Route Choice, Safety Perception, and Helmet Use," *Sustainability*, vol. 15, no. 8, p. 6609, Jan. 2023, number: 8 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2071-1050/15/8/6609

[5] D. Chen, A. Hosseini, A. Smith, D. Xiang, A. Heydarian, O. Shoghli, and B. Campbell, "Impact of Different Infrastructures and Traffic Scenarios on Behavioral and Physiological Responses of E-scooter Users," May 2024, arXiv:2407.10310 [cs]. [Online]. Available: http://arxiv.org/abs/2407.10310

[6] J. D. Cano-Moreno, J. M. Arenas Reina, V. d. C. Parra Lanillos, and M. E. Islán Marcos, "Sensitivity of Mass Geometry Parameters on E-Scooter Comfort: Design Guide," *Sensors*, vol. 24, no. 2, p. 399, Jan. 2024, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/24/2/399

[7] S. Cafiso, A. Di Graziano, V. Marchetta, and G. Pappalardo, "Urban road pavements monitoring and assessment using bike and e-scooter as probe vehicles," *Case Studies in Construction Materials*, vol. 16, p. e00889, Jun. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214509522000213

[8] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, Jul. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2590005621000059

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. [Online]. Available: http://ieeexplore.ieee.org/document/7780460/

[10] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester, "Lost and Found: detecting small road hazards for self-driving vehicles," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 1099–1106, iSSN: 2153-0866. [Online]. Available: https://ieeexplore.ieee.org/document/7759186/?arnumber=7759186

[11] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: A multi-national image dataset for automatic road damage detection," *Geoscience Data Journal*, vol. 11, no. 4, pp. 846–862, 2024, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/gdj3.260. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/gdj3.260

[12] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12387. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12387

[13] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 1, pp. 47–60, 2021, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12561. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12561

[14] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2020: An annotated image dataset for automatic road damage detection using deep learning," *Data in Brief*, vol. 36, p. 107133, Jun. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352340921004170

[15] M. Ren, X. Zhang, X. Zhi, Y. Wei, and Z. Feng, "An annotated street view image dataset for automated road damage detection," *Scientific Data*, vol. 11, no. 1, p. 407, Apr. 2024. [Online]. Available: https://www.nature.com/articles/s41597-024-03263-7

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. [Online]. Available: https://ieeexplore.ieee.org/document/7485869/?arnumber=7485869

[17] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018, arXiv:1804.02767 [cs]. [Online]. Available: http://arxiv.org/abs/1804.02767

[18] G. Jocher, "YOLOv5 by Ultralytics," May 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[19] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications,” Sep. 2022, arXiv:2209.02976 [cs]. [Online]. Available: http://arxiv.org/abs/2209.02976

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 7464–7475. [Online]. Available: https://ieeexplore.ieee.org/document/10204762/

[21] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[22] C.-Y. Wang and H.-Y. M. Liao, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information,” *arXiv preprint arXiv:2402.13616*, 2024.

[23] L. L. e. a. Ao Wang, Hui Chen, “YOLOv10: Real-Time End-to-End Object Detection,” *arXiv preprint arXiv:2405.14458*, 2024, publisher: Tsinghua University.

[24] G. Jocher and J. Qiu, “Ultralytics YOLO11,” 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[25] P. Wenzelburger and F. Allgower, “A first step towards an autonomously driving E-Scooter.”

[26] R. Soloperto, P. Wenzelburger, D. Meister, D. Scheuble, V. S. Breidohr, and F. Allgöwer, “A control framework for autonomous e-scooters,” *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 252–258, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896321004675

[27] C.-H. Lin and S.-Y. Ye, “Design of Intelligent Electric Scooter with a Cloud Monitoring System,” in *2018 International Symposium on Computer, Consumer and Control (IS3C)*, Dec. 2018, pp. 161–164. [Online]. Available: https://ieeexplore.ieee.org/document/8645005/?arnumber=8645005

[28] H. Alai, W. Jeon, L. Alexander, and R. Rajamani, “Rear Vehicle Tracking on a Smart E-Scooter,” in *2023 American Control Conference (ACC)*. San Diego, CA, USA: IEEE, May 2023, pp. 1735–1740. [Online]. Available: https://ieeexplore.ieee.org/document/10156621/

[29] H. Alai and R. Rajamani, “Multi-Stage Estimation Algorithm for Target Vehicle Trajectory Tracking With Applications to E-Scooter Protection,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 3022–3033, Jan. 2024, conference Name: IEEE Transactions on Intelligent Vehicles. [Online]. Available: https://ieeexplore.ieee.org/document/10230854/?arnumber=10230854

[30] H. Alai, W. Jeon, L. Alexander, and R. Rajamani, “A smart e-scooter with embedded estimation of rear vehicle trajectories for rider protection,” *Mechanical Systems and Signal Processing*, vol. 222, p. 111786, Jan. 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0888327024006848

[31] S. Yousuf, R. Al-Mannai, B. Al-Naemi, S. Al-Maadeed, N. Nawaz, and M. Chaari, “Smart System for a Self-Driving Scooter Prototype,” in *2023 International Symposium on Networks, Computers and Communications (ISNCC)*, Oct. 2023, pp. 1–5, iSSN: 2768-0940. [Online]. Available: https://ieeexplore.ieee.org/document/10323714/?arnumber=10323714

[32] R. Strässer, M. Seidel, F. Brändle, D. Meister, R. Soloperto, D. H. Ferrer, and F. Allgöwer, “Collision Avoidance Safety Filter for an Autonomous E-Scooter using Ultrasonic Sensors,” May 2024, arXiv:2403.15116. [Online]. Available: http://arxiv.org/abs/2403.15116

[33] X. Yan and D. Shen, “Active Collision Avoidance System for E-Scooters in Pedestrian Environment,” Nov. 2023, arXiv:2311.04383. [Online]. Available: http://arxiv.org/abs/2311.04383

[34] D. Chen, A. Hosseini, A. Smith, A. F. Nikkhah, A. Heydarian, O. Shoghli, and B. Campbell, “Performance Evaluation of Real-Time Object Detection for Electric Scooters,” May 2024, arXiv:2405.03039 [cs]. [Online]. Available: http://arxiv.org/abs/2405.03039

[35] A. Li, "Finding a safety feature algorithm for electric-scooter by detecting the closest obstacles."

[36] D. Tian, A. D. Ryan, C. M. Craig, K. Sievert, and N. L. Morris, "Characteristics and Risk Factors for Electric Scooter-Related Crashes and Injury Crashes among Scooter Riders: A Two-Phase Survey Study," *International Journal of Environmental Research and Public Health*, vol. 19, no. 16, p. 10129, Jan. 2022, number: 16 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1660-4601/19/16/10129

[37] M. Trefzger, W. Titov, and T. Schlegel, "Analysis and Comparison of the Gaze Behavior of E-Scooter Drivers and Cyclists." Gesellschaft für Informatik, Bonn, 2021, pp. 1163–1180. [Online]. Available: https://dl.gi.de/handle/20.500.12116/34696

[38] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Jul. 2020, pp. 237–242, iSSN: 2157-8702. [Online]. Available: https://ieeexplore.ieee.org/document/9145130/?arnumber=9145130

[39] D. V. McGehee, E. N. Mazzae, and G. S. Baldwin, "Driver Reaction Time in Crash Avoidance Research: Validation of a Driving Simulator Study on a Test Track," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 44, no. 20, pp. 3–320, Jul. 2000, num Pages: 3-323 Publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.1177/154193120004402026

[40] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023. [Online]. Available: https://doi.org/10.1007/s11042-022-13644-y

[41] D. Yang, C. Miao, X. Li, Y. Liu, Y. Wang, and Y. Zheng, "Improved YOLOv7 Network Model for Gangue Selection Robot for Gangue and Foreign Matter Detection in Coal,"

Sensors, vol. 23, no. 11, p. 5140, Jan. 2023, number: 11 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/23/11/5140

[42] "Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics." [Online]. Available: https://github.com/ultralytics/ultralytics/issues/189

[43] P. Hidayatullah, N. Syakrani, M. R. Sholahuddin, T. Gelar, and R. Tubagus, "YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review," Jan. 2025, arXiv:2501.13400 [cs]. [Online]. Available: http://arxiv.org/abs/2501.13400