iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70471

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Real-Time Object Detection and Distance Mapping

Saketh Kinthali[1], Sai Krishna Pudi[2], Dr Raghavendra V[3]
[1, 2]UG student, [3]Assistant Professor, Department of Computing Technologies SRM institute of science and technology, Chennai, Tamil Nadu, India,

Abstract: This initiative involves an advanced AI-driven object detection system that employs the YOLO (You Only Look Once) deep learning framework to recognize and monitor objects in real-time from both images and videos. It comprises multiple Python scripts, including object_detection.py, real_time.py, and video_with_distance.py, which facilitate the identification of objects in photographs, live video feeds, and the estimation of their distances. The system is equipped with pre-trained YOLO model weights (yolov8m.pt, yolov8n.pt), enabling rapid and effective object recognition. Additionally, it provides sample videos (33.mp4, 34.mp4) and images (bus.jpg, output_detected.jpg) to evaluate the performance of the detection model. A COCO dataset file (coco.txt) is included, signifying that the model has been trained to identify a diverse range of common objects. Furthermore, other Python scripts (conv.py, test4.py) appear to be utilized for data conversion, testing, or enhancing the system's capabilities. This project holds significant potential for applications in real-time surveillance, autonomous vehicles, intelligent traffic management, security systems, and AI-driven automation, thereby improving the efficiency of object detection and tracking across various practical scenarios.

Keywords: Object Detection, YOLO (You Only Look Once), Deep Learning, Computer Vision, Real-Time Detection, AI-powered Automation, COCO Dataset, Bounding Box Annotation, Image Recognition, Video Processing, Machine Learning, Autonomous Systems, Smart Surveillance, Security Monitoring, Artificial Intelligence

I. INTRODUCTION

Artificial intelligence and deep learning are revolutionizing our interactions with technology, enhancing the intelligence and capabilities of machines. A notable advancement in this domain is object detection, which enables computers to identify and monitor objects within images and videos. This project features an AI-driven object detection system that employs the YOLO (You Only Look Once) deep learning model to recognize and analyze objects in real time. By utilizing pre-trained YOLO model weights (yolov8m.pt, yolov8n.pt), the system can swiftly and accurately detect a wide range of objects. It includes several Python scripts designed to process images, live video feeds, and even estimate the distance to detected objects. To evaluate its effectiveness, the project provides sample videos (33.mp4, 34.mp4) and images (bus.jpg, output_detected.jpg), demonstrating the model's performance across various scenarios. Furthermore, it utilizes the COCO dataset (coco.txt), which enhances the model's capability to identify common objects.

As the demand for real-time surveillance, autonomous vehicles, intelligent traffic management, and security systems continues to grow, this project offers a rapid, dependable, and flexible solution for AI-driven automation. Its proficiency in detecting objects in practical environments positions it as a valuable asset for industries aiming to incorporate advanced AI technologies into their operations.

II. MOTIVATION

As technology progresses, the significance of AI-driven automation and computer vision in daily life is becoming increasingly prominent. Applications such as security surveillance, autonomous vehicles, and intelligent traffic management systems require machines to identify and monitor objects in real time to facilitate swift and accurate decision-making. This project was motivated by the demand for a rapid, dependable, and efficient object detection system capable of functioning seamlessly in various environments.

Many conventional object detection techniques face challenges related to speed and precision, rendering them less suitable for practical applications. This is where the YOLO (You Only Look Once) deep learning model proves advantageous, as it is engineered for quick and accurate object detection, making it ideal for real-time scenarios.

Our objective with this project is to leverage the capabilities of YOLO to create a system that can identify and track objects in both images and live video streams, while also estimating distances.

By utilizing pre-trained YOLO models, COCO dataset labels, and sample test files, this initiative offers a practical and accessible solution for those interested in incorporating AI into their projects. Whether aimed at enhancing security, optimizing traffic monitoring, or advancing smart automation, this system is designed to improve the effectiveness and reliability of AI-driven technologies in real-world settings.

III. DATA SET

A large collection of photos commonly used in computer vision applications such as object detection, image segmentation, and caption creation is called the COCO (Common Objects in Context) dataset. These categories include a wide range of commonplace objects, making the dataset particularly useful for creating models that can identify and understand a variety of objects in natural environments. The diversity of COCO includes animals like cats and dogs, vehicles such as cars and buses, household items like chairs and refrigerators, and food items like pizza and bananas, making it a rich resource for training machine learning models focused on object detection and recognition.

Each object within the COCO dataset is annotated with a bounding box that specifies its position in the image. Furthermore, it offers segmentation masks that enable more accurate identification by delineating the precise area of the object at the pixel level. The dataset also features key points for certain objects, including human poses, along with captions that describe the scene in natural language. This functionality makes COCO not only beneficial for object detection but also for more intricate tasks such as caption generation and answering questions related to the images.

COCO is divided into test, validation, and training sets. Models are trained using the training set, and their parameters are adjusted using the validation set, and the test set evaluates the performance of the final model. Additionally, there are specialized subsets designed for specific tasks, such as key point detection for human poses or caption generation. This organization allows COCO to be adaptable enough to address a wide range of computer vision challenges.

Many models, including YOLO (You Only Look Once), Faster R-CNN, Mask R-CNN, and SSD (Single Shot MultiBox Detector), which are designed to detect numerous items in real-time, have been trained using the COCO dataset.

Its extensive range of objects, along with challenges like occlusions, overlapping items, and diverse environments, provides a demanding yet realistic training environment for these models.

A distinctive feature of the COCO dataset is its focus on real-world contexts within its images, frequently depicting objects in intricate settings characterized by varying lighting conditions, backgrounds, and possible obstructions. This aspect enhances the dataset's value, as it aids in developing more resilient models capable of addressing practical, everyday situations. Due to its comprehensive scale and diversity, COCO has established itself as the primary benchmark for testing and evaluating object detection models, with numerous prominent challenges utilizing it to measure model efficacy.

The COCO dataset is accessible at no cost, enabling researchers and developers to download both the images and their corresponding annotations for their projects. It has become a fundamental resource in the realm of computer vision, significantly contributing to progress in fields such as object recognition, segmentation, and even image captioning.

IV. LITERATURE REVIEW

Appiah and Mensah (2024) explored object detection in adverse weather conditions for autonomous vehicles, highlighting the challenges posed by rain, fog, and snow. Their study demonstrated that deep learning models, combined with sensor fusion techniques, can significantly improve object recognition accuracy under harsh environmental conditions. However, they acknowledged that existing models still struggle with occlusions and varying lighting conditions.

Yeong et al. (2021) provided a comprehensive review of sensor and sensor fusion technologies used in autonomous vehicles. Their study underscored the importance of integrating multiple sensor types, such as LiDAR, cameras, and radar, to enhance vehicle perception. Despite advancements, they noted that real-time data processing and cost constraints remain significant hurdles in large-scale implementation.

Wason et al. (2024) investigated the convergence of artificial intelligence and mechatronics in autonomous driving. They emphasized the role of AI in decision-making, path planning, and obstacle avoidance, demonstrating that integrating mechatronic systems with AI can enhance vehicle efficiency. However, the authors pointed out that safety concerns and algorithm interpretability remain key challenges.

Ghintab and Hassan (2023) focused on localization for self-driving vehicles using deep learning networks and RGB cameras. Their findings revealed that deep learning-based localization outperforms traditional GPS-based methods in urban environments.

Nevertheless, their research highlighted that model accuracy is dependent on high-quality training datasets and real-time computational efficiency.

Hasanujjaman et al. (2023) examined sensor fusion in autonomous vehicles, particularly in traffic surveillance systems. They proposed an AI-based approach for detection and localization using traffic cameras, improving vehicle tracking accuracy. While their method demonstrated promising results, scalability and integration with existing infrastructure were noted as potential limitations.

Hacohen et al. (2022) conducted a survey on technological gaps in autonomous driving using trend analysis from Google Scholar and Web of Science. They identified key challenges, including regulatory issues, ethical concerns, and robustness of AI models. Their study emphasized the need for continuous advancements in AI safety and legal frameworks for widespread adoption.

Mahima et al. (2024) reviewed adversarial attacks and countermeasures in 3D perception for autonomous vehicles. Their research showed that autonomous systems are vulnerable to adversarial perturbations, which can mislead object detection models. They suggested robust defense mechanisms, including adversarial training and sensor fusion, to enhance security.

Huang et al. (2025) introduced OpenGV 2.0, a motion prior-assisted calibration and SLAM framework for vehicle-mounted surround-view systems. Their method improved localization accuracy by leveraging motion priors, reducing errors in real-world navigation scenarios. However, they acknowledged that real-time implementation requires substantial computational power.

Yan et al. (2022) developed Agenti2p, an optimization framework for image-to-point cloud registration using behavior cloning and reinforcement learning. Their approach enhanced the accuracy of sensor alignment in autonomous vehicles. While their results were promising, they pointed out that real-world testing in dynamic environments is needed to validate the model's robustness.

Collectively, these studies highlight the ongoing advancements and persistent challenges in autonomous vehicle technology. While AI, sensor fusion, and deep learning have significantly improved perception and localization, issues related to safety, real-time processing, and environmental adaptability remain key areas for future research.

V. METHODOLOGY

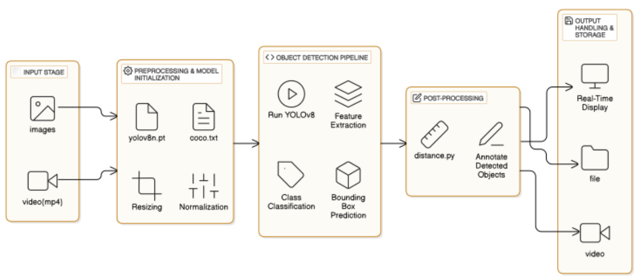

Fig 1. Autonomous Navigation & Safety(Cars/Objects) using Yolo model

1) Data Acquisition

The project utilizes pre-trained YOLOv8 models (yolov8m.pt, yolov8n.pt) for the purpose of object detection. These models have been developed using extensive datasets to identify a wide range of objects. The dataset designated for fine-tuning or validation is detailed in the coco.txt file, which presumably includes class labels and relevant metadata. Input data for testing and validation comprises image and video files, such as bus.jpg, 33.mp4, and 34.mp4. By incorporating both image and video formats, the model's performance is assessed in both static and dynamic environments, thereby evaluatingitsrobustness and accuracy across various scenarios.

2) Preprocessing

Prior to executing object detection, the input data is subjected to preprocessing to enhance its suitability for model inference. Images and video frames are imported into the system and resized to conform to the input dimensions specified by YOLOv8. This step guarantees that the model can process the data efficiently while preserving essential details. Additional modifications, including normalization and color space conversion, may be implemented to ensure a consistent input format. For real-time detection, video streams are analyzed frame by frame utilizing OpenCV. This enables the model to evaluate live footage and identify objects in real-time, which is vital for applications like surveillance and autonomous navigation. The preprocessing phase is critical in ensuring that the input data is organized and primed for precise object detection.

3) Model Implementation

The object detection pipeline is developed utilizing the YOLOv8 deep learning framework. Various Python scripts, including object_detection.py, real_time.py, and video_with_distance.py, cater to distinct applications. The object_detection.py script focuses on identifying objects within static images, while real_time.py facilitates live detection through a connected camera. Additionally, the video_with_distance.py script enhances functionality by integrating distance estimation, which is beneficial for scenarios requiring measurement of object proximity. The YOLOv8 model processes the input data by segmenting it into a grid and predicting bounding boxes, class labels, and confidence scores for the identified objects. By harnessing the real-time processing capabilities of YOLOv8, the system guarantees swift and precise object recognition across various contexts.

4) Post-processing

Following the object detection phase, the results are subjected to post-processing to improve their clarity and usability. Detected objects are emphasized by applying bounding boxes and labels to the original images or video frames. This visualization aids users in comprehending the model's predictions and evaluating its accuracy. The refined outputs are subsequently stored in various formats, including images (output_detected.jpg) and videos, for documentation and further examination. The post-processing stage also involves filtering detections according to confidence thresholds to reduce false positives and enhance precision. Moreover, additional features, such as tracking objects across frames, may be integrated to further refine detection outcomes and offer deeper insights in dynamic environments.

5) Evaluation and Optimization

To guarantee the model's effectiveness, its performance is assessed through various configurations and optimization methods. The analysis focuses on accuracy and detection speed by testing different confidence thresholds and non-maximum suppression (NMS) parameters. The choice between yolov8m.pt and yolov8n.pt hinges on the balance between model size and accuracy; yolov8m.pt provides greater precision, whereas yolov8n.pt is designed for enhanced speed. Performance indicators such as precision, recall, and mean average precision (MAP) are employed to evaluate detection quality. Optimization approaches may involve fine-tuning the model with a custom dataset, minimizing computational demands, and modifying hyperparameters to boost efficiency. These measures contribute to improving the model's accuracy and its adaptability in various environments.

6) Deployment

The concluding phase of the project focuses on the implementation of the object detection system for real-world applications. This system can be integrated into an application that facilitates real-time detection via live camera feeds, making it ideal for uses in security surveillance, traffic monitoring, and industrial automation. Future enhancements may include the integration of the model into web or mobile applications, enabling users to conduct object detection from remote locations. The deployment phase also takes into account hardware limitations, ensuring that the system is compatible with edge devices like Raspberry Pi or NVIDIA Jetson for real-time processing. By providing access to the system across multiple platforms, the project enhances its functionality and influence in various fields.

VI. RESULT AND DISCUSSION

The object detection system underwent evaluation using various images and videos with the YOLOv8 models (yolov8m.pt and yolov8n.pt). These models effectively recognized and labeled objects, enclosing them within bounding boxes. The detection accuracy was notably high for static images, while the system demonstrated reliable tracking capabilities in video sequences. The real-time detection module operated efficiently, exhibiting minimal lag, and the video with distance.py script provided additional functionality by estimating distances to objects, thereby enhancing spatial awareness.

To evaluate the system's accuracy, the image "bus2.jpg" (448 × 640 pixels) was scrutinized. The model successfully detected several objects:

- People: Five individuals were identified, with confidence levels ranging from 0.39 to 0.91. The lower confidence scores indicate that some individuals may have been partially obscured.
- Buses: Two buses were detected, one with a high confidence score of 0.91 and the other with a low score of 0.27, suggesting potential misclassification.
- Truck: A truck was identified, but with a low confidence score of 0.27, indicating uncertainty in its classification.

- Accessories: A backpack was recognized with a confidence score of 0.71, along with two handbags with scores of 0.38 and 0.25, respectively, reflecting some inconsistency in classification.

The total processing time for this image was 98.5 milliseconds, with 92.2 milliseconds allocated to inference. These findings underscore the system's efficiency in real-time applications, although the low-confidence detections highlight the necessity for refining confidence thresholds. The performance of the object detection system was influenced by factors such as model selection, image resolution, and environmental conditions. The yolov8m.pt model provided higher accuracy, making it ideal for applications that prioritize precision, while the yolov8n.pt model offered faster performance, albeit with a slight compromise in accuracy.

Several challenges were identified, particularly the model's sensitivity to low-light conditions, which occasionally led to missed detections or lower confidence scores. Implementing image processing techniques, such as brightness adjustments or histogram equalization, may enhance performance. Furthermore, the model sometimes misclassified or overlooked smaller objects when they overlapped. Employing methods like non-maximum suppression (NMS) and object tracking could alleviate these problems.

In conclusion, the YOLOv8 model shows significant promise for real-time object detection. Future enhancements could involve fine-tuning the model with tailored datasets and incorporating cloud-based or edge computing solutions to improve scalability. By addressing these limitations, the system's reliability across various applications could be further strengthened.

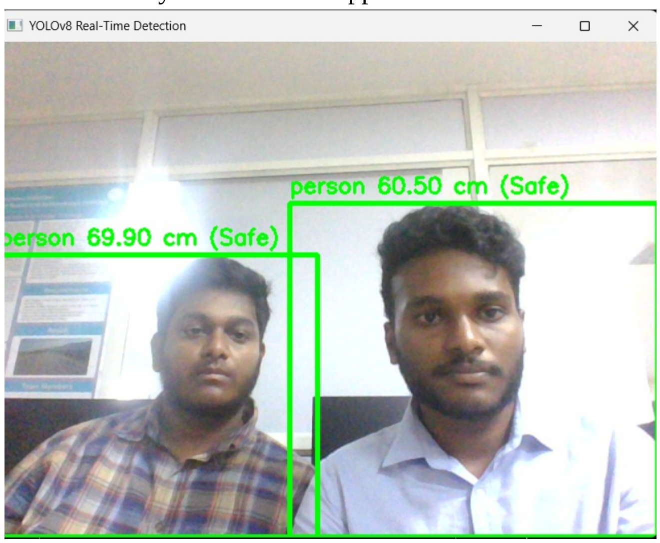

Fig 2. Real Time Detection with safe distanceof persons

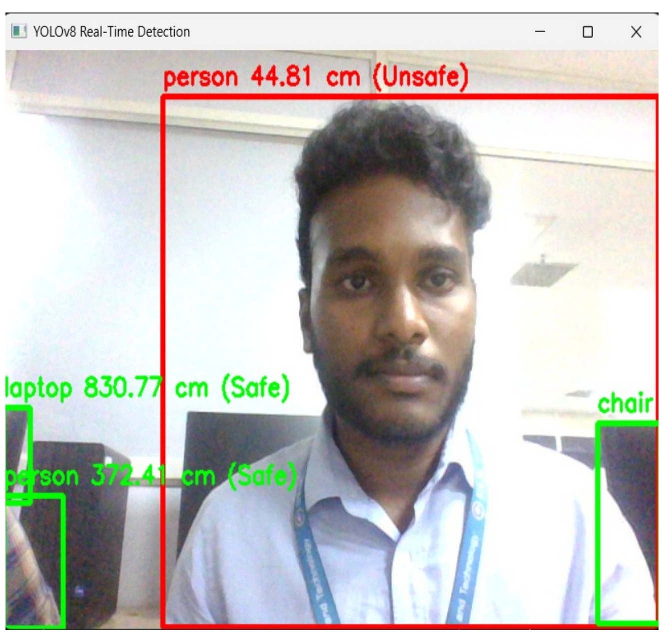

Fig 3. Real Time Detection with unsafe distance of a person

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

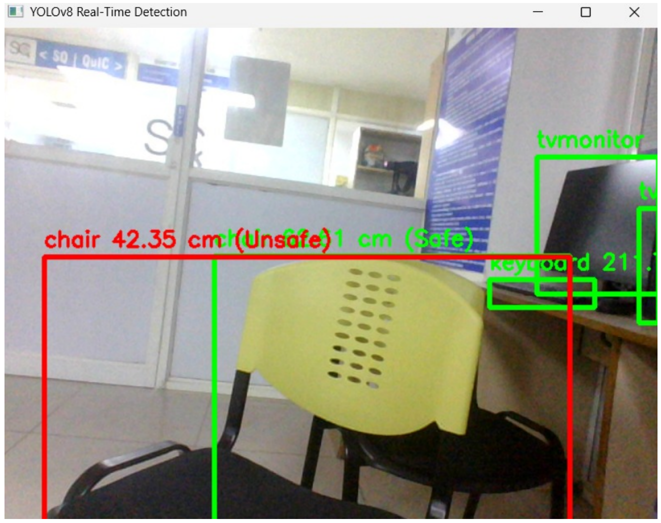

Fig 4. Real Time Detection with unsafe distance of a chair

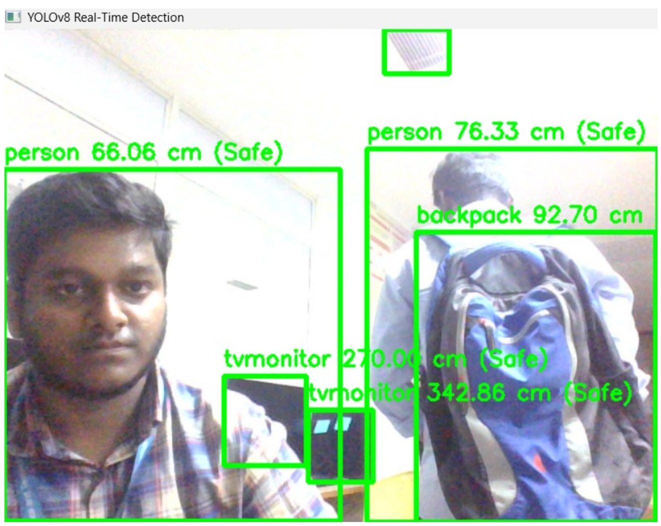

Fig 5. Real Time Detection with safe distance of a backpack

Table1. Confidence Score

Object	Confidence
Person	0.91
Bus	0.91
Person	0.84
Person	0.83
Person	0.78
Backpack	0.71
Person	0.91
HandBag	0.66

Truck	0.68
Bus	0.75
HandBag	0.66

VII. CONCLUSION

The Autonomous Navigation & Safety System (Cars/Objects) utilizing the YOLO model represents a cutting-edge solution aimed at improving personal safety through advanced technological integration. This system merges hand gesture recognition, facial emotion detection, voice activation, GPS tracking, and real-time live streaming to function as an automated emergency response assistant, delivering prompt and effective safety interventions. Leveraging deep learning, computer vision, and real-time communication, it is capable of identifying distress signals and responding without delay. The emergency alert feature guarantees swift assistance for users, while live video streaming and GPS tracking provide immediate updates for responders. With its diverse capabilities, the system serves as a dependable safety net for individuals in precarious situations or emergencies. Future enhancements could include the integration of AI-driven behavioral analysis, IoT connectivity, and cloud-based data storage to further improve its efficiency and scalability. This initiative underscores the potential of AI in enhancing safety and illustrates the significant role technology can play in safeguarding lives and fostering a safer environment.

REFERENCES

[1] Appiah, Emmanuel Owusu, and Solomon Mensah. "Object detection in adverse weather condition for autonomous vehicles." Multimedia Tools and Applications 83, no. 9 (2024): 28235-28261.
[2] Yeong, De Jong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. "Sensor and sensor fusion technology in autonomous vehicles: A review." Sensors 21, no. 6 (2021): 2140.
[3] Sarasa-Cabezuelo, Antonio. "Prediction of rainfall in Australia using machine learning." Information 13, no. 4 (2022): 163.
[4] Wason, Ritika, Parul Arora, Vishal Jain, Devansh Arora, and M. N. Hoda. "Exploring the Convergence of Artificial Intelligence and Mechatronics in Autonomous Driving." Computational Intelligent Techniques in Mechatronics (2024): 297-316.
[5] Ghintab, Shahad S., and Mohammed Y. Hassan. "Localization for self-driving vehicles based on deep learning networks and RGB cameras." International Journal of Advanced Technology and Engineering Exploration 10, no. 105 (2023): 1016.
[6] Hasanujjaman, Muhammad, Mostafa Zaman Chowdhury, and Yeong Min Jang. "Sensor fusion in autonomous vehicle with traffic surveillance camera system: detection, localization, and AI networking." Sensors 23, no. 6 (2023): 3335.
[7] Hacohen, Shlomi, Oded Medina, and Shraga Shoval. "Autonomous driving: A survey of technological gaps using google scholar and web of science trend analysis." IEEE Transactions on Intelligent Transportation Systems 23, no. 11 (2022): 21241-21258.
[8] Mahima, KT Yasas, Asanka G. Perera, SreenathaAnavatti, and Matt Garratt. "Toward robust 3d perception for autonomous vehicles: A review of adversarial attacks and countermeasures." IEEE Transactions on Intelligent Transportation Systems (2024).
[9] Huang, Kun, Yifu Wang, Si'ao Zhang, Zhirui Wang, Zhanpeng Ouyang, Zhenghua Yu, and Laurent Kneip. "OpenGV 2.0: Motion prior-assisted calibration and SLAM with vehicle-mounted surround-view systems." arXiv preprint arXiv:2503.03230 (2025).
[10] Yan, Shen, Maojun Zhang, Yang Peng, Yu Liu, and Hanlin Tan. "Agenti2p: Optimizing image-to-point cloud registration via behaviour cloning and reinforcement learning." Remote Sensing 14, no. 24 (2022): 6301.