Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 10, No. 1, 2024

Multi-Platform Electric Vehicle Detection System in Elevators

Yichen Li, Meina Zhang *, Jiayu Wang, Yuxuan Liu, Yutong Xing, Xinyuan Wang

School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan Liaoning, 114001, China
*** Corresponding author:** Meina Zhang (Email: zhangmeina@163.com)

Abstract: This project addresses elevator safety by proposing a solution based on an improved lightweight YOLOv3 model. The system is trained on a custom-built dataset of electric vehicles inside elevators, achieving efficient and accurate object detection suitable for edge computing environments. It demonstrates excellent performance through transfer learning and comparative experiments. The user interface, developed with PyQt5 and Streamlit, supports image, video, and real-time detection, along with result-saving capabilities. Tests on elevator videos show outstanding accuracy and practicality, making this solution highly applicable to smart elevators and public safety management.

Keywords: Object Detection System; Edge Computing; Elevator Safety.

1. Introduction

In recent years, safety incidents within elevators have surged, primarily due to the rising popularity of electric scooters as a mode of urban transportation. With more individuals bringing electric scooters into elevators, a range of safety hazards has emerged, underscoring the urgent need for effective technological solutions to prevent such practices. This project addresses this challenge by integrating a lightweight YOLOv3 model with multi-platform interfaces built using PyQt5 and Streamlit, resulting in a highly efficient and user-friendly target detection solution. The model is optimized to deliver high detection accuracy while being compact enough for deployment on edge devices, such as elevator monitoring systems. This optimization enhances the practicality and convenience of safety monitoring in elevators, offering a scalable solution to address emerging safety risks in urban environments.

2. The Analysis of Current Issues Related to Electric Vehicle Detection in Elevators

(1) Problem Background

In recent years, elevator safety has become an increasingly important area of focus, especially as the use of electric vehicles, such as e-bikes, in elevators has introduced new safety risks. Despite this growing concern, there remains a substantial gap in research on the interactions between e-bikes and elevator environments, resulting in a shortage of publicly available datasets specific to this issue. This lack of data presents a significant barrier to understanding and mitigating the associated risks.

In addition, elevator surveillance systems—often operating as edge devices with limited processing power—face the complex challenge of balancing high detection accuracy with the need for lightweight models. These systems must be able to accurately detect e-bikes within elevators in real time without compromising performance or overloading the device's computational resources.

Given these constraints, the primary technical objectives of this project are twofold. First, to bridge the data gap by acquiring and curating comprehensive datasets that capture various scenarios involving e-bikes inside elevators. Second, to develop a highly efficient and lightweight detection model that can reliably identify e-bikes within confined elevator spaces with high accuracy. By addressing these goals, the project aims to enhance overall elevator safety, proactively tackling a rising safety risk within urban environments where e-bike usage continues to grow.

(2) Existing Solutions

In 2019, Hua Zhichao integrated a batch normalization algorithm into YOLOv1 and improved the loss function to detect prohibited objects inside elevators[1]. In 2020, Cen Siyang used the GIOU loss function in YOLOv3 and improved the generation of bounding boxes to achieve facial detection inside elevators[2]. In 2021, Zhang Yuan and colleagues proposed an e-bike detection algorithm based on YOLOv3, enabling real-time detection of e-bikes in elevators[3]. In 2023, Yang Xianyu improved the backbone networks of YOLOv3 and YOLOv4 to make the models more lightweight, allowing them to be deployed on edge devices[4][5]. Although these algorithms have improved detection performance, current applications still exhibit a high rate of false positives, such as misidentifying strollers, pets, or children as e-bikes, which disrupts daily life.

(3) Solution Approach

In this project, we gathered images from a variety of sources, including web searches, video frame extraction, and web scraping, to build a comprehensive dataset. To simulate diverse real-world scenarios and enhance the robustness of our model, we applied three types of data augmentation techniques to these collected images. The images were then annotated using the open-source tool LabelImg, allowing us to generate a dataset in YOLO format specifically tailored for object detection tasks.

Once the dataset was prepared, we conducted multiple rounds of training to fine-tune the model and optimize its weight file for the highest possible accuracy. This refined model was subsequently used for inference testing to evaluate its performance. To make the detection process accessible and user-friendly, we developed a visual detection interface, which was packaged as a standalone application.

In the final stage, we designed and deployed a web-based

platform hosted on the cloud, providing users with an easy-to-access link to interact with the model. This platform allows users to experience the model's capabilities in real time, making advanced visual detection technology more accessible to a broader audience. The development process of the system is shown in Figure 1.

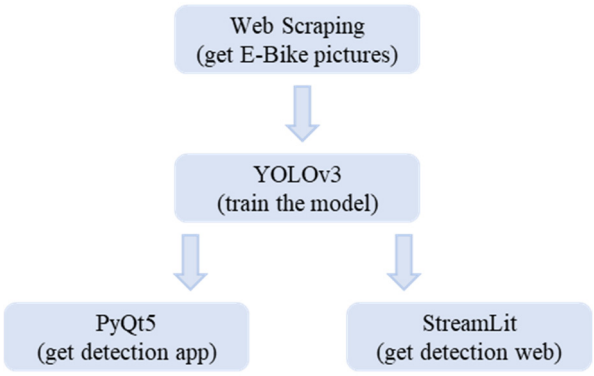

Figure 1. The Development Process of the System

3. The Solutions for Detection System Development

(1) Data Acquisition and Annotation

The dataset was collected and annotated using several Python libraries and tools to streamline the data acquisition and preparation process. First, the YOU-GET library was employed to automatically download all videos from specified URLs, providing an efficient way to gather video data in bulk. For images, the BeautifulSoup library was used to parse HTML content from Baidu image search results based on targeted keywords. This allowed for automated downloading and local storage of relevant images, significantly accelerating the data collection process.

To enhance the diversity and robustness of the dataset, various data augmentation techniques were applied, including Gaussian blur to simulate different levels of focus, horizontal flipping to introduce variability in orientation, and random rotation to account for different angles of view. These augmentations help improve model generalization by providing a wider range of examples during training.

Finally, for annotation, the open-source tool LabelImg was used to manually label the data, ensuring that each image and video frame was accurately annotated with bounding boxes. This comprehensive approach to dataset preparation establishes a strong foundation for model training and validation, ultimately contributing to the overall success of the project.

(2) Data Training and Improvement Process

The experiments are run on the Ubuntu 20.04 operating system with Python 3.8, using PyTorch version 2.0.0. The hardware configuration includes an NVIDIA RTX 4090 GPU with 24GB of memory and a single Intel(R) Xeon(R) Platinum 8352V CPU with 12 virtual CPUs, operating at a frequency of 2.10GHz for computations. During the experiments, the network input is set to 640x640, utilizing the ADAM optimizer along with transfer learning to obtain pre-trained weights. The initial learning rate for the model is set to 0.001, with a batch size of 16, and the training and testing dataset ratio is 8:2. A total of 200 iterations are conducted, and the training process is illustrated in Figure 2.

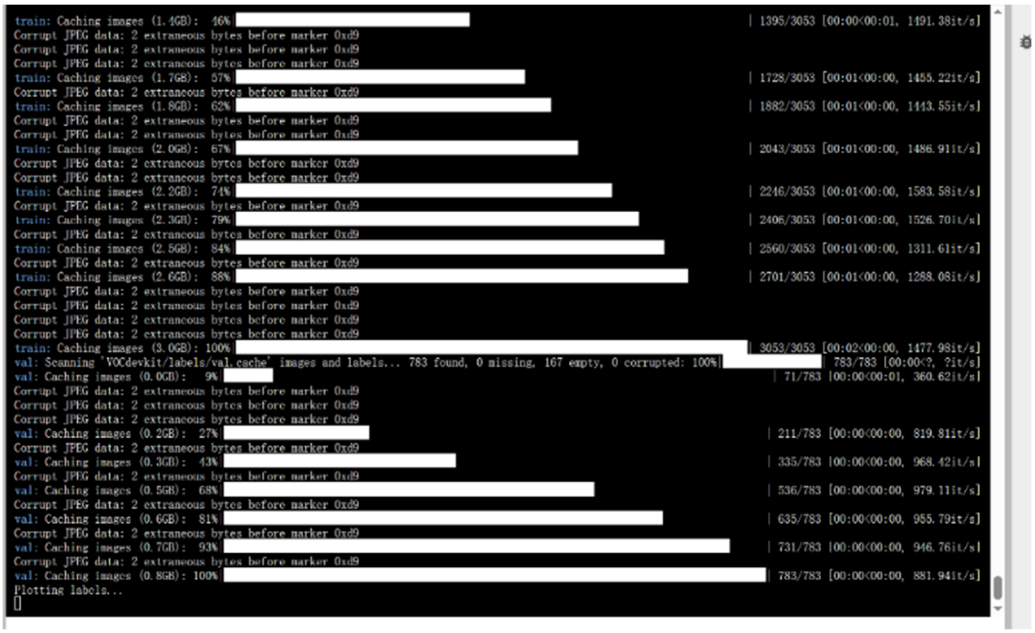

Figure 2. The Process of Training Model

(3) System Development and Deployment

The application, built using PyQt5, is designed to operate on Windows and features a fully functional graphical user interface (GUI) for an intuitive user experience. After development is complete, the Python script is converted into a standalone executable using the auto-py-to-exe tool, making it easy to distribute and run independently without requiring a Python environment.

In parallel, an interactive web application is developed using Streamlit, an open-source Python framework that streamlines web app creation. The Streamlit app includes several essential features to improve user interaction and functionality: a threshold slider to adjust detection sensitivity, an interactive interface for real-time detection, options to select different model files, and the capability to upload images or videos for analysis.

Upon finalizing development, the entire codebase is uploaded to GitHub, allowing for version control, easy collaboration, and transparency. The web application is then deployed on Streamlit's cloud platform, creating a public link

that allows users to access and interact with the app directly from their browsers, as shown in Figure 3. This deployment strategy not only simplifies usage for end-users but also enhances accessibility, making it easier to gather user feedback and conduct wider testing.

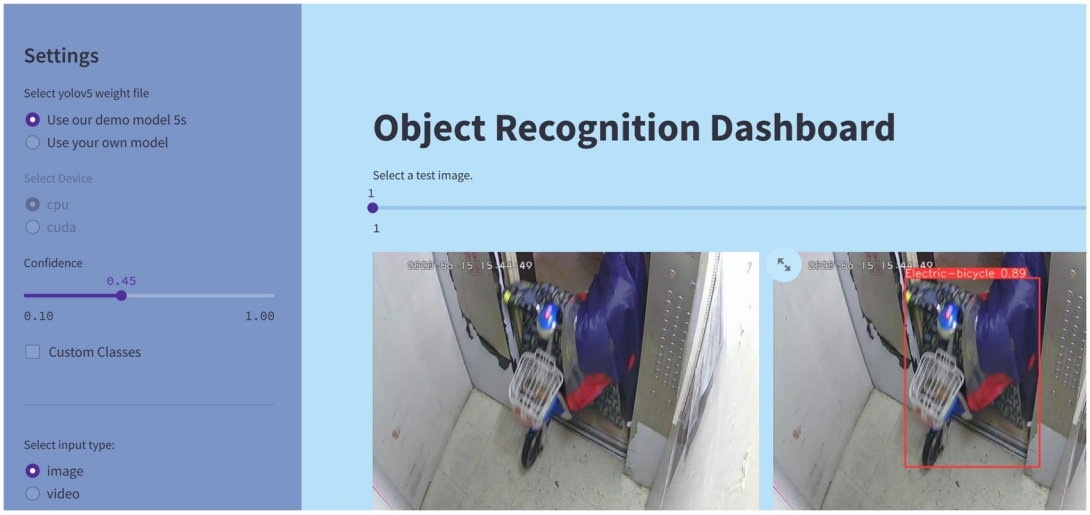

Figure 3. The Detection System

4. The Testing Analysis of the Detection System

(1) Model Performance and Inference Results

In this project, we conducted a series of rigorous comparative experiments using both the original and augmented datasets to thoroughly assess the model's performance. Our findings demonstrated a marked improvement in mean Average Precision (mAP) across both datasets, with the model trained on the augmented dataset achieving significantly higher accuracy compared to the model trained solely on the original data. This boost in performance underscores the effectiveness of data augmentation in enhancing the model's generalization capabilities, enabling it to maintain high accuracy across a broader array of real-world scenarios.

As a result, the detection model implemented in our system was trained on the augmented dataset, achieving high recognition rates and exceptional precision. These results not only meet but, in several cases, surpass the project's initial performance targets, underscoring the pivotal role of data augmentation in strengthening the model's robustness, accuracy, and overall reliability. This approach provides a solid foundation for deploying the model in complex, real-world applications where precision and consistency are paramount.

(2) Application Testing

The application features an intuitive graphical interface developed using PyQt5, making it user-friendly and easily deployable as a standalone executable. Through this interface, users can seamlessly upload model weight files along with images or videos for analysis. By simply clicking the "Object Detection" button, the system initiates the detection process, providing quick and accurate results.

For real-time monitoring needs, the application also includes a "Camera Detection" feature, which allows users to access live detection directly through the interface. This is particularly useful for applications such as elevator surveillance, where continuous monitoring is essential. Detected videos can be exported and saved locally, enabling users to archive footage for further analysis or record-keeping purposes.

During testing, the system exhibited a high level of accuracy, especially in distinguishing between electric vehicles (e-bikes) and other objects, such as strollers. The model consistently detected e-bikes without mistakenly identifying strollers, demonstrating its robustness and precision in complex, confined environments. This capability offers a reliable tool for enhancing safety monitoring in spaces where misclassification could lead to safety risks. The system's accuracy and real-time functionality make it an invaluable asset for improving safety protocols in enclosed areas like elevators.

(3) Web Testing

The electric vehicle detection system has been designed as a web-based application built with Streamlit, offering users a convenient, cloud-hosted solution accessible through a single link. This online platform enables users to effortlessly upload videos or images for electric vehicle detection, harnessing advanced pre-trained models to provide accurate and reliable results. The system also offers a high degree of customization, allowing users to fine-tune detection sensitivity by adjusting threshold settings. This flexibility ensures that the detection system can be tailored to meet diverse operational needs and requirements, making it adaptable to various environments and use cases. With its user-friendly interface and robust performance, this platform simplifies the process of electric vehicle detection, making it accessible to both technical and non-technical users alike.

Since the system runs entirely within a web browser, it removes the need for any software installation, making it easily accessible to users regardless of their technical expertise. This streamlined, hassle-free interface is designed to accommodate both quick, on-the-go checks and more in-depth monitoring tasks. With a thoughtful blend of flexibility, ease of use, and dependable performance, the platform provides an efficient, adaptable solution for monitoring electric vehicles in various settings. It meets the demands of a wide range of applications without the complications associated with traditional software, offering a robust tool for users who need powerful yet accessible detection capabilities. Whether for occasional checks or continuous monitoring, this platform empowers users to perform sophisticated detection with minimal effort.

5. Conclusion

The project effectively compresses the model size by adopting the lightweight version of YOLOv3—YOLOv3-Tiny—while maintaining high precision and recall rates, making it suitable for deployment on resource-limited edge devices, such as elevator monitoring systems. This optimization not only enhances the efficiency and reliability of the monitoring system but also significantly reduces hardware costs and computational resource requirements.

Additionally, the project has developed a desktop application based on PyQt5 and a web application interface using Streamlit, providing a user-friendly and highly interactive operational interface that supports various detection methods (including images, videos, and real-time cameras), further expanding the system's applicability and flexibility.

The innovation of this project lies in its integration of deep learning models with client-side and web applications, optimized for edge computing environments, and creatively applying intelligent monitoring technology to elevator monitoring, offering new solutions for safety management.

The project has significant potential for future development. With the continuous advancement of deep learning technology and the ongoing enhancement of edge computing capabilities, the project can further optimize the efficiency and accuracy of the YOLOv3-Tiny model to achieve more effective and precise object detection. At the same time, by continually improving the user interfaces based on PyQt5 and Streamlit, the project can offer richer functionalities and a better user experience, catering to the needs of diverse user groups.

Acknowledgments

Fund: 2024 Innovation and entrepreneurship training program for College Students.

References

[1] Hua Zhichao. Research and Implementation of Intrusion Detection Algorithm for Elevator Cab Based on Surveillance Video [D]. Master's Thesis, Southeast University, 2019.

[2] Cen Siyang. Research and Application of Target Detection in Elevators Based on Deep Learning [D]. Master's Thesis, Anhui University of Technology, 2020.

[3] Zhang Yuan, Feng Yu. Design of Electric Vehicle Detection System in Elevators Based on Raspberry Pi and YOLOv3 [J]. Information Technology and Informatization, 2022, 263(2): 105-108.

[4] Yang Xianyu. Elevator Electric Vehicle Detection Algorithm Based on Improved YOLOv3 [J]. Computer Era, 2023, (07): 61-65. DOI: 10.16644/j.cnki.cn33-1094/tp.2023.07.014.

[5] Yang Xianyu. Elevator Electric Vehicle Detection Algorithm Based on Improved YOLOv4 [J]. Computer Era, 2023, (10): 54-58. DOI: 10.16644/j.cnki.cn33-1094/tp.2023.10.