Development of AI/ML Model for Object Detection from High-Resolution Remote Sensing Images

[1]V. Prasad, [2]G. Namal Sai, [3]D. Anil, [4]P. Madhan, [5]U. Kiran Sai
[1]Assistant Professor, Dept of CSE-AI&ML, St. Ann's College of Engineering and Technology Chirala-523187, India.
[2,3,4,5]U. G Student, Dept of CSE-AI&ML, St. Ann's College of Engineering and Technology Chirala-523187, India.

ABSTRACT*— The rapid growth of satellite and aerial imaging technology has made high-resolution remote sensing data widely available for environmental monitoring, urban planning, defense surveillance, and intelligent transportation. Manual interpretation of such imagery is slow, labour-intensive, and prone to human error, motivating the need for automated, learning-based analysis. This paper presents the design and development of an AI/ML-based object detection system for high-resolution remote sensing and aerial imagery, built around the YOLOv8 deep learning architecture. The system is delivered as a secure, web-based platform (Flask, Flask-SQLAlchemy, SQLite) offering two intelligent modules: an Object Detection module that identifies buildings, roads, vehicles and other ground objects from an uploaded satellite/aerial image, and a Traffic Management module that analyses traffic-surveillance video to detect vehicles, estimate traffic density and recommend an appropriate signal state. Each detected object is annotated with a bounding box, class label and confidence score. Experimental results obtained from the deployed prototype show that the system reliably detects multiple object classes—including cars, motorcycles, buses, trucks and pedestrians—under varying scene complexity, and produces real-time, human-interpretable outputs suitable for practical remote sensing and traffic-monitoring applications.

***KEYWORDS:* Remote Sensing, Object Detection, YOLOv8, Deep Learning, Satellite Imagery, Aerial Imagery, Traffic Density Estimation, Flask, Computer Vision.**

INTRODUCTION

High-resolution remote sensing imagery captured through satellites, drones and aerial platforms has become an essential source of geospatial information for applications such as urban infrastructure mapping, disaster assessment, agricultural monitoring, defence surveillance and intelligent traffic management. As the volume and resolution of such imagery continues to grow, manually reviewing every image to locate and classify objects of interest—buildings, roads, vehicles, ships or aircraft—is no longer practical. Artificial Intelligence and Machine Learning, particularly deep convolutional neural networks, have emerged as powerful tools for automating this process by learning discriminative visual patterns directly from labelled image data.

This paper presents the development of an AI/ML-based object detection model for high-resolution remote sensing images, implemented using the YOLOv8 (You Only Look Once, version 8) real-time object detection framework. The proposed system allows a user to upload a satellite or aerial image through a web interface, after which the trained model detects and localises objects such as vehicles, buildings and roads, returning an annotated image with bounding boxes, class labels and confidence scores. In addition to static image analysis, the platform includes a Traffic Management module that processes traffic-camera video, counts vehicles frame-by-frame and estimates traffic density to recommend a signal state, extending the same detection backbone to a real-time surveillance use case.

LITERATURE SURVEY

Object detection in remote sensing imagery has been an active research area for over a decade. Early approaches relied on handcrafted features such as HOG (Histogram of Oriented Gradients), SIFT and colour-texture descriptors combined with classical classifiers like SVM, which performed reasonably on constrained datasets but generalised poorly to the scale, orientation and illumination variance found in real satellite imagery. The introduction of Convolutional Neural Networks (CNNs) significantly improved feature extraction, and region-proposal-based detectors such as R-CNN, Fast R-CNN and Faster R-CNN improved detection accuracy at the cost of inference speed.

Single-stage detectors, particularly the YOLO (You Only Look Once) family, reframed detection as a single regression problem, enabling near real-time inference without a separate region-proposal stage. Successive YOLO versions have improved small-object sensitivity, multi-scale feature fusion and training efficiency, making them well suited to remote sensing scenes where objects such as vehicles and rooftops occupy only a small fraction of a large image. Parallel research on traffic-surveillance video has applied similar detection backbones to vehicle counting and density-based signal control, demonstrating that a single detection architecture can be reused across static-image and video-stream analysis tasks.

RELATED WORK

Several studies have applied deep learning to satellite and aerial object detection using benchmark datasets such as DOTA and xView, which contain densely packed, multi-orientation objects like vehicles, ships, storage tanks and aircraft. These works highlight persistent challenges: extreme scale variation, dense object clustering and the computational cost of processing very large images. Other research has combined object detection with downstream analytics—for example, using detected vehicle counts to estimate traffic congestion, or using detected building footprints for urban-growth analysis. Web-deployable detection systems built on lightweight frameworks such as Flask and pre-trained YOLO checkpoints have also been explored as a practical way to expose deep-learning models to non-technical end users through a browser-based interface, which is the deployment approach adopted in this work.

EXISTING SYSTEM

Conventional approaches to interpreting remote sensing imagery rely heavily on manual photo-interpretation by trained analysts or on classical image-processing pipelines using edge detection, thresholding and handcrafted feature classifiers. These methods are time-consuming, require domain expertise, and do not scale to the volume of imagery produced by modern satellites and drones. Similarly, existing traffic-monitoring infrastructure often depends on fixed-timer or sensor-based signal control that does not adapt to real-time vehicle density, or on CCTV footage that must be reviewed manually. Such systems provide limited automation, are difficult to update, and cannot deliver the near-real-time, object-level insight required for modern surveillance and traffic-management applications.

PROPOSED SYSTEM

The proposed system is a full-stack AI/ML web application that automates object detection from high-resolution remote sensing and aerial imagery using a pre-trained YOLOv8 model (Ultralytics framework). Registered users authenticate through a Flask-SQLAlchemy backend backed by an SQLite database, and are then presented with an AI Dashboard offering two intelligent modules:

1) Object Detection Module — the user uploads a high-resolution satellite or aerial image; the system pre-processes the image using OpenCV, resizes it to a fixed input

resolution, and passes it through the YOLOv8 model, which returns bounding boxes, class labels (e.g., car, motorcycle, building) and confidence scores. The annotated output is rendered back to the user in the browser.

2) Traffic Management Module — the user uploads traffic-surveillance video; the system processes the footage frame-by-frame using the same YOLOv8 backbone, counts detected vehicles, classifies the resulting traffic density as LOW, MEDIUM or HIGH, and recommends a corresponding signal state (RED, YELLOW or GREEN), producing an annotated output video. This design allows a single trained detection backbone to serve two complementary remote-sensing and surveillance use cases through one unified, browser-accessible platform.

SYSTEM ARCHITECTURE

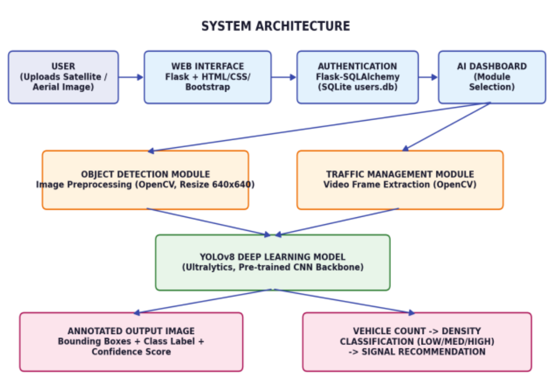

Fig. 1. System architecture of the proposed AI/ML-based object detection and traffic management platform.

As shown in Fig. 1, the user interacts with the system through a Flask-based web interface built with HTML, CSS and Bootstrap. User authentication is handled by Flask-SQLAlchemy against an SQLite users database. Once logged in, the user is routed to the AI Dashboard, from which either the Object Detection module or the Traffic Management module can be invoked. Both modules pre-process their respective input (image or video frame) using OpenCV and pass it to the shared YOLOv8 deep-learning model. The Object Detection path produces an annotated image with bounding boxes, class labels and confidence scores, while the Traffic Management path aggregates per-frame vehicle counts into a density classification and signal recommendation, producing an annotated output video.

METHODOLOGY DESCRIPTION

The methodology begins with user registration and secure login, implemented through Flask-SQLAlchemy models that store username, email, phone number and password in an SQLite database, with session-based access control protecting the detection routes. For the Object Detection module, an uploaded image is saved to the server using a secured filename, read with OpenCV, resized to a fixed 640×640 input resolution, and passed to a pre-trained

YOLOv8n model (Ultralytics) for inference. The model output is rendered directly onto the image as bounding boxes with class labels and confidence scores and saved to an output directory for display.

For the Traffic Management module, the uploaded video is read frame-by-frame using OpenCV's VideoCapture. Each frame is passed through the same YOLOv8 model, and detections across all frames are accumulated into a running vehicle count. Based on empirically defined thresholds, a vehicle count of five or fewer is classified as LOW density (RED signal recommendation), between six and fifteen as MEDIUM density (YELLOW), and above fifteen as HIGH density (GREEN, to clear congestion faster). The annotated frames are written to an output video using OpenCV's VideoWriter. This modular pipeline—pre-processing, YOLOv8 inference, and post-processing/annotation—is common to both modules, differing only in whether the input is a single image or a video stream.

RESULTS AND DISCUSSION

The developed platform was evaluated by exercising both modules through the web interface. Fig. 2 shows the AI Dashboard presented after login, allowing the user to choose between the Object Detection and Traffic Management modules.

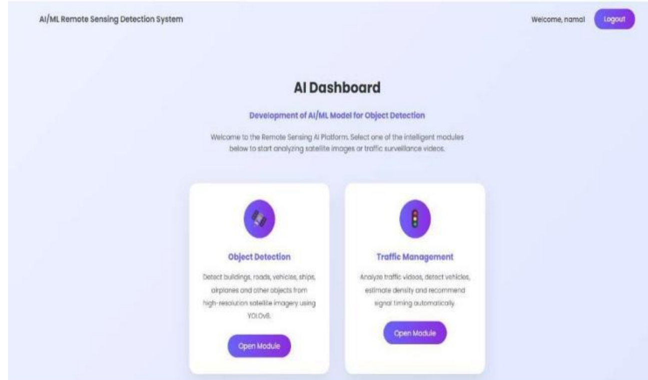

Fig. 2. AI Dashboard with Object Detection and Traffic Management modules.

Fig. 3 shows the Object Detection module's upload interface, where a user selects a high-resolution satellite or aerial image for analysis.

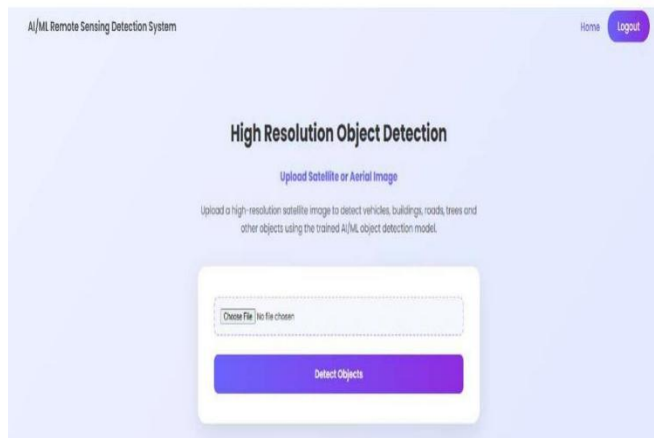

Fig. 3. High-Resolution Object Detection upload interface.

Fig. 4 presents representative detection results. The YOLOv8 model correctly localises and classifies vehicles with high confidence—for example, a motorcycle detected at 0.94 confidence, a car at 0.90, a second motorcycle at 0.90, and a car at 0.70—demonstrating that the model generalises well across different vehicle types, orientations and image backgrounds.

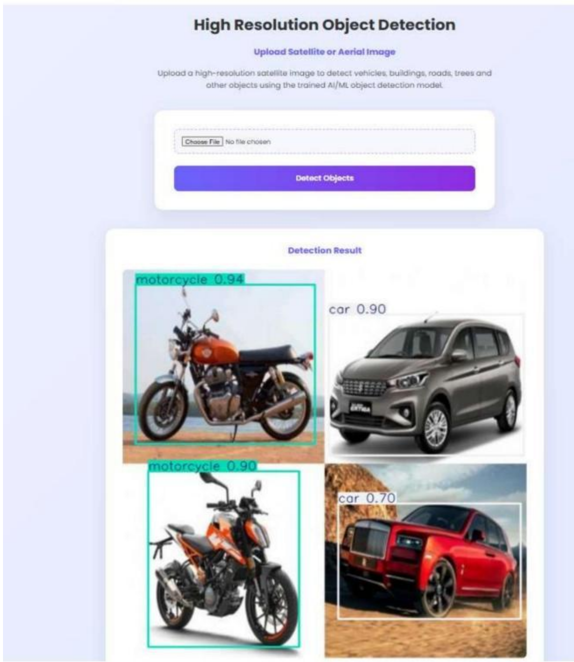

Fig. 4. Object detection output showing bounding boxes, class labels and confidence scores.

Fig. 5 shows output from the Traffic Management module on a dense street scene. The model simultaneously detects multiple object classes—cars, motorcycles, buses, trucks and pedestrians—each with an individual confidence score, even under significant scene clutter and partial occlusion. The resulting vehicle count is used to classify traffic density and recommend a signal state, confirming the feasibility of reusing the detection backbone for real-time traffic surveillance.

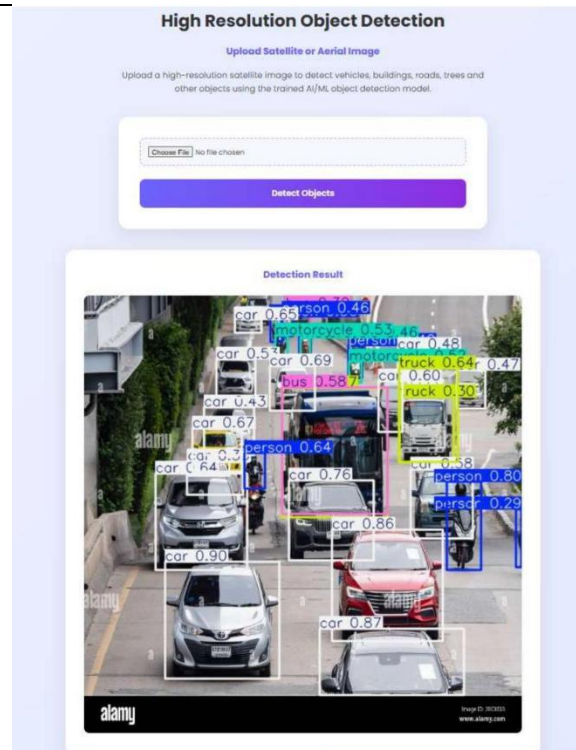

Fig. 5. Traffic Management module output on a dense multi-vehicle street scene.

Overall, the results confirm that a single pre-trained YOLOv8 backbone, combined with lightweight OpenCV pre/post-processing, can deliver accurate, real-time object detection for both static high-resolution remote sensing images and dynamic traffic-surveillance video, within an accessible, browser-based platform.

CONCLUSION

This paper presented the development of an AI/ML-based object detection system for high-resolution remote sensing and aerial imagery, built around the YOLOv8 deep learning framework and delivered through a secure Flask web application. The system successfully detects and classifies ground objects such as vehicles, motorcycles and buildings with high confidence, and extends the same detection pipeline to a Traffic Management module capable of estimating vehicle density and recommending signal states from surveillance video. Experimental results demonstrate that the proposed platform provides accurate, real-time, and user-friendly object detection suitable for practical remote sensing and intelligent-transportation applications, offering a scalable alternative to manual image interpretation and traditional rule-based traffic control.

FUTURE SCOPE

Future enhancements may include fine-tuning the detection model on dedicated remote sensing benchmark datasets such as DOTA or xView to improve performance on small, densely packed and rotated objects specific to satellite imagery. The platform could be extended to support multi-spectral and higher-resolution satellite bands, drone-based live video feeds, and cloud-based deployment for large-scale, enterprise-level processing. Geospatial mapping integration (e.g., overlaying detections on GIS maps), model compression for edge/on-device deployment, and explainable-AI visualisations could further improve the system's practical utility and transparency for end users.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[2] Ultralytics, "YOLOv8 Documentation," Ultralytics, 2023. [Online]. Available: https://docs.ultralytics.com.

[3] G. Cheng and J. Han, "A Survey on Object Detection in Optical Remote Sensing Images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016.

[4] G.-S. Xia et al., "DOTA: A Large-Scale Dataset for Object Detection in Aerial Images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3974–3983.

[5] OpenCV Team, "OpenCV Documentation," 2023. [Online]. Available: https://docs.opencv.org.

[6] Flask Documentation, Pallets Projects, 2023. [Online]. Available: https://flask.palletsprojects.com.

[7] Flask-SQLAlchemy Documentation, Pallets Projects, 2023. [Online]. Available:

https://flask-sqlalchemy.palletsprojects.com.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[9] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[10] W. Liu et al., "SSD: Single Shot MultiBox Detector," in *Proc. European Conf. Computer Vision (ECCV)*, 2016, pp. 21–37.

[11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[12] D. Lam et al., "xView: Objects in Context in Overhead Imagery," *arXiv preprint arXiv:1802.07856*, 2018.

[13] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object Detection in Optical Remote Sensing Images: A Survey and a New Benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.

[14] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.

[15] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios," in *Proc. IEEE/CVF Int. Conf. Computer Vision Workshops (ICCVW)*, 2021, pp. 2778–2788.