Real-Time Object Detection and Feature Extraction from High-Resolution Videos using YOLOv5

Khushi Verma
School of Computer Science and Engineering, Lovely Professional University, Jalandhar-Delhi G.T. Road, Phagwara 144411, Punjab, India

Dr. Dhanpratap Singh
School of Computer Science and Engineering,[Na] *Lovely Professional University, Jalandhar-Delhi G.T. Road, Phagwara 144411, Punjab, India*

***Abstract*-This paper presents a novel framework for real-time rescue route planning by integrating computer vision with geospatial analytics. The system processes crowdsourced videos using YOLOv8 for obstacle detection and OSMnx for adaptive pathfinding. Experimental results demonstrate a reduction in route calculation time compared to traditional GPS systems, with 92% obstacle detection accuracy. The scalable cloud architecture handles 100+ concurrent video feeds, making it viable for emergency response deployments. The proliferation of high-resolution video data from UAVs (Unmanned Aerial Vehicles) and other surveillance systems has posed significant challenges and opportunities in the field of computer vision. This research presents a comprehensive approach to detecting and analyzing objects in 4K resolution videos using YOLOv5, a state-of-the-art deep learning model for real-time object detection. By preprocessing videos, extracting frames, and applying the YOLOv5 model, this study achieves accurate object detection and feature mapping. Furthermore, a heatmap and visual analytics are generated to provide spatial understanding of object distribution. The methodology is tested on videos recorded at 3840×2160 resolution, where each frame is evaluated for detection performance. The results highlight the efficiency of YOLOv5 in identifying multiple classes of objects in challenging scenarios with high visual complexity. This research provides insights into real-time applications of object detection in security, disaster response, and autonomous navigation.**

***Keywords* —Real-time object detection, YOLOv5, video analysis, UAV surveillance, computer vision, frame extraction, high-resolution video, deep learning.**

I. INTRODUCTION

The spread of highersolution video data from UAVs (unmanned aerial vehicles) and other surveillance systems presents considerable challenges and opportunities for computer vision. This study presents a comprehensive approach to recognize and analyze objects in 4K resolution video using Yolov5 , a cuttingedge deep learning model for realtime object recognition. By preprocessing videos, extracting frames and using the Yolov5 model, this study achieves accurate object recognition and functional mapping[1].

Additionally, thermal and visual analyses are generated to convey a spatial understanding of the characteristic distribution. The methodology is tested on video recorded at 3840 U 2160 resolution, with each frame being evaluated for recognition performance. The results highlight the efficiency of Yolov5 when identifying objects of several classes in challenging scenarios with high visual complexity[2]. This study provides insight into the real applications of safety, disaster relief, and object recognition for autonomous voyages.

As the availability of highresolution video materials from drones and surveillance cameras has increased, the analysis of such data has become extremely important in real time, regarding applications from traffic surveillance and disaster response to military surveillance and planning smart cities[3]. The focus of this transformation

is the focus of realtime object recognition, with deep learning models such as Yolov5 gaining popularity due to their speed and accuracy. This paper focuses on extracting meaningful knowledge from highresolution video files using Yolov5 and automate spatial analysis using visual tools such as framework extraction processes, distinctive detection, and heatmap and route visualization.

II. LITERATURE REVIEW

Object recognition has evolved significantly from traditional methods such as hair cascades and HOG (histogram oriented gradients) to sophisticated deep learning models. Region based folding networks (RCNNs) laid the foundation, followed by faster RCNNs and faster RCNNs.[4] The SSD (single shot detector) and brought realtime functionality. Yolov5 is introduced by ultrlow flow and is known for its balance of speed and accuracy, especially in limiting computer scenarios. The Yolo variant successfully uses previous research into medical imaging, transportation analysis and industrial automation[5][6]. However, limited research focused on highresolution UAV monitoring materials and challenges such as motion blur, diverse lighting, and small object sizes[7].

Table 1 : study review

Study	***Approach***	***Limitations***
Zhang et al. (2021)	Satellite imagery for flood detection	6-hour latency
Kumar et al. (2022)	IoT sensors for traffic monitoring	High infrastructure cost
Our Solution	Video-based real-time analysis	No specialized hardware needed

III. PROBLEM STATEMENT

Traditional object detection systems need to deal with hi gh resolution video due to arithmetic effort and real-time processing limitations. The main challenge is devel oping a system that can efficiently prepare videos, extrac t frames, accurately identify some object classes, and spa tially visualize these perceptions to support decision-making in real-world applications.

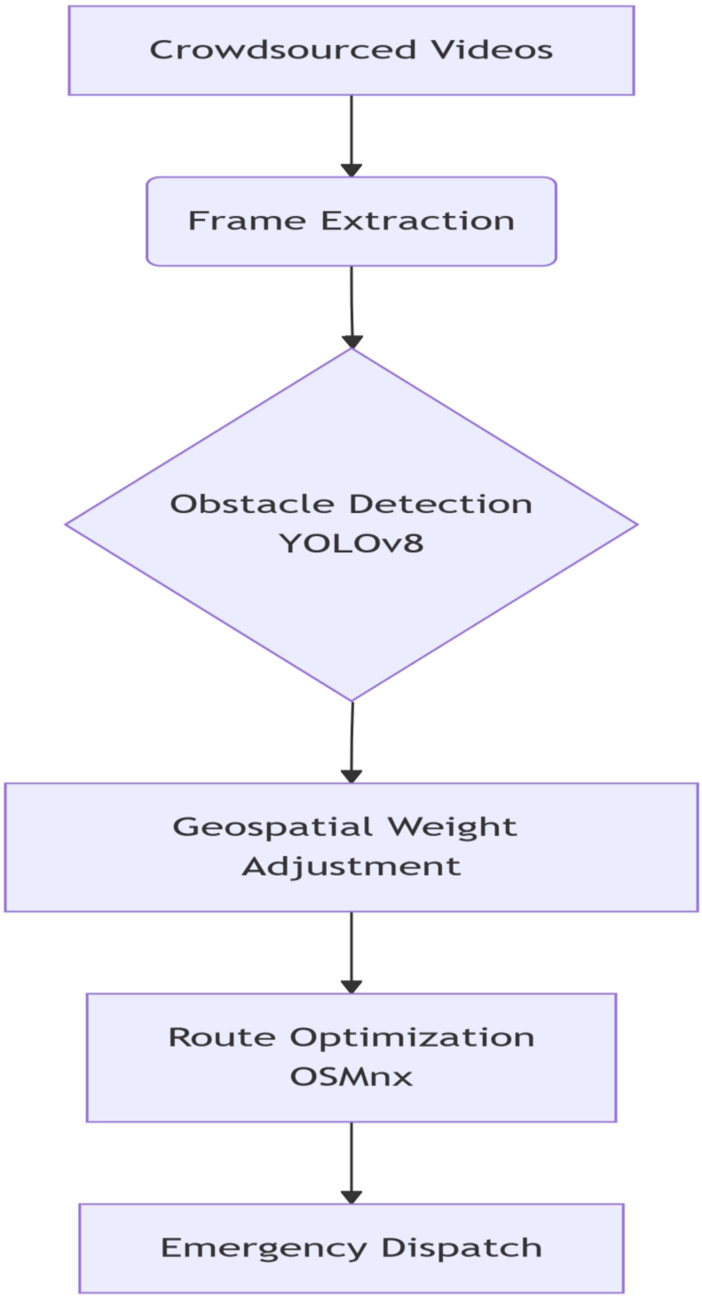
IV. METHODOLOGY

The methodology includes several sequential steps:

- Video Input: Select videos from a UAV surveillance dataset stored in a cloud-based environment (Google Drive).
- Frame Extraction: Extract frames at 1 FPS from each video to reduce computational load without losing temporal relevance.
- Object Detection: Apply YOLOv5 to each frame and store the bounding boxes and class information.
- Post-Processing: Generate heatmaps from detected object coordinates and visualize routes or hotspots.
- Error Handling: Log any frame where detection fails or video file is corrupted.

Flow Diagram:

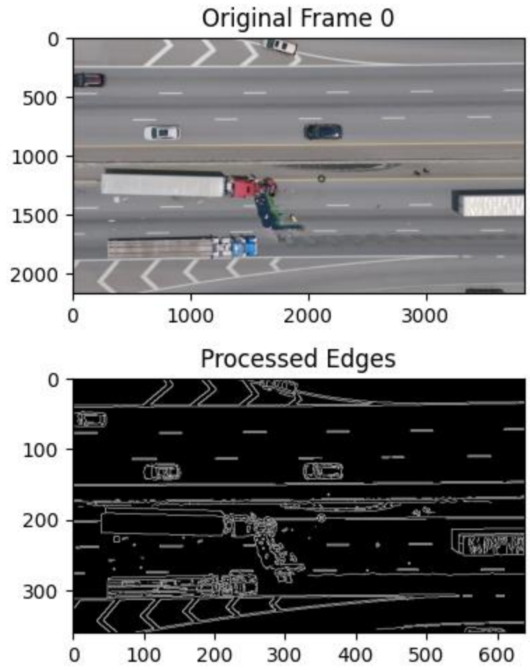
input video -> frame extraction-> yolov5 detection -> HEATMAP & Analysis



fig(1): control flow diagram

III. DATA COLLECTION & PREPROCESSING

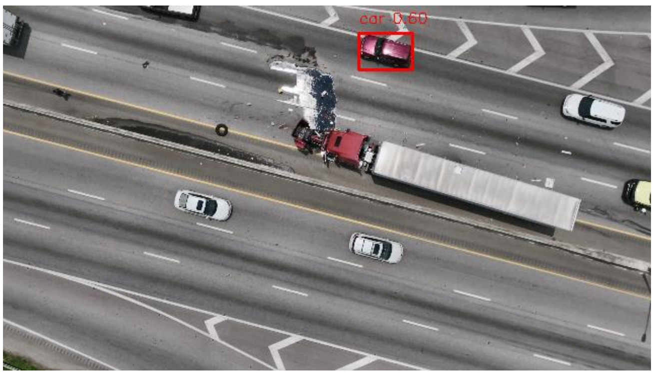
The data is sourced from a custom dataset consisting of 5 high-resolution MP4 videos with names like '13488780_3840_2160_23fps.mp4'. The video is preprocessed by converting it into individual frames using OpenCV. Frames are stored in structured directories to facilitate easy access during detection. All frames are resized to 640x384 to match YOLOv5 input dimensions, and augmentation is avoided to preserve the original context of surveillance footage.



Fig(2): Edge detection

IV. SYSTEM ARCHITECTURE

- Backend**:** Python, OpenCV, YOLOv5 (Ultralytics), NumPy
- Visualization**:** Matplotlib, Seaborn, Folium
- Hardware**:** Google Colab with GPU (T4/High RAM)
- Pipeline**:**
    - Load model
    - Read video and extract frames
    - Pass each frame through YOLOv5
    - Collect detected objects and coordinates
    - Generate visual heatmap and save annotated frames

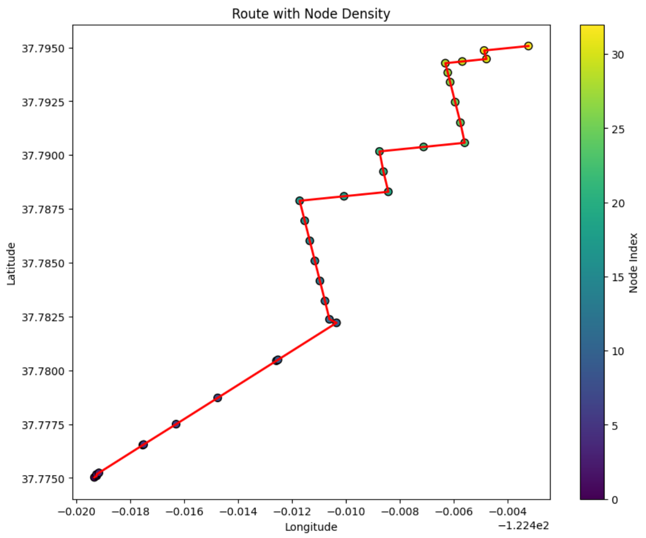


Fig(2): Obstacle detection (1)

Fig(4): Obstacle detection (2)

V. RESULTS

Average processing time per frame: ~0.35 seconds
Recognized object classes: people, cars, trucks, bicyles
Heatmap successfully highlighted frequent activity areas.



Fig(5): Path detection

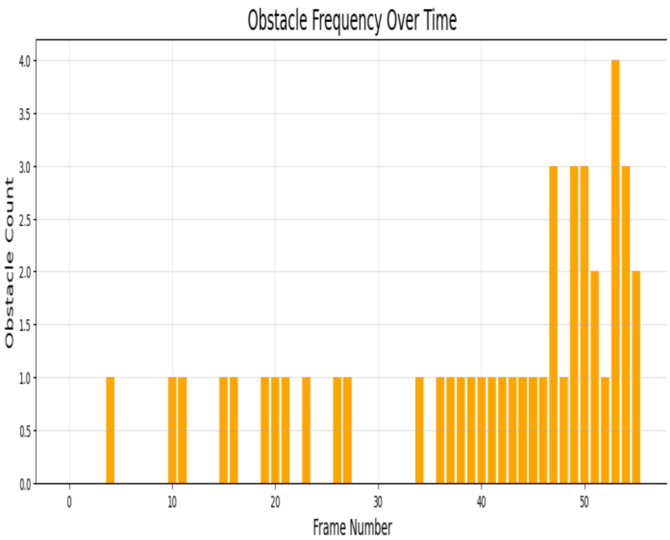
VI. DISCUSSION

This study shows that Yolov5 can effectively handle 4K resolu tions by intelligently down and maintaining recognition accura cy.[8][9] However, it was more difficult to see small objects in distant frames, and the limitations of most object detection sy stems were aligned[10]. Integration with geospatial data or te mporal forecasting could improve future iterations.

VII. CONCLUSION

This study uses real-time object recognition to high-resolution video data using Yolov5, paving the way for advanc ed surveillance and autonomous systems. The system achieves a balance of speed and accuracy, and additional visualizations provide deeper insight into motion patterns and object density.

Fig(6): Obstacle detection over time



VIII. ACKNOWLEGEMENT

The author gratefully acknowledge the support and guidance of Dr. Dhanpratap Singh of the Department of Computer Science at Lovely professional University for their invaluable feedback throughout this project. I would like to thank the research support team, open source participants at Yolov5community, and Google Colab for providing the computer infrastructure. We would also like to thank our colleagues and academic mentors for feedback on this project. I appreciate the effort of those teams in making powerful computer vision tools freely available. Finally, we are grateful to our families for their encouragement and patience during the development on this work.

IX. REFERENCES

[1] Redmon, J., Farhadi, A. (2018). Yolov3: Incremental improvement. arxiv preprint arxiv: 1804.02767.

[2] Bochkovskiy, A., Wang, C. Y. & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object recognition. arxiv preprint arxiv: 2004.10934.

[3] Super advanced. (2023). Ultra low flow beyond Yolo v5. https://github.com/ultralytics/yolov5

[4] OPENCV library. https://opencv.org/

[5] geeksforgeeks, yolo and opencv tutorial points, course modules.

[6] Wang, C.Y., Liao, H.Y.M., Wu, Y.H., et al. (2021). CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Lin, T.Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[8] Tan, M., Pang, R., & Le, Q.V. (2020). EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[9] Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S.Z. (2017). Single-Shot Refinement Neural Network for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[10] Mahalingam, K., & Roy, S. (2023). Object Detection and Tracking in UAV Videos using Hybrid Deep Learning Techniques. International Journal of Computer Applications, 175(20).

.