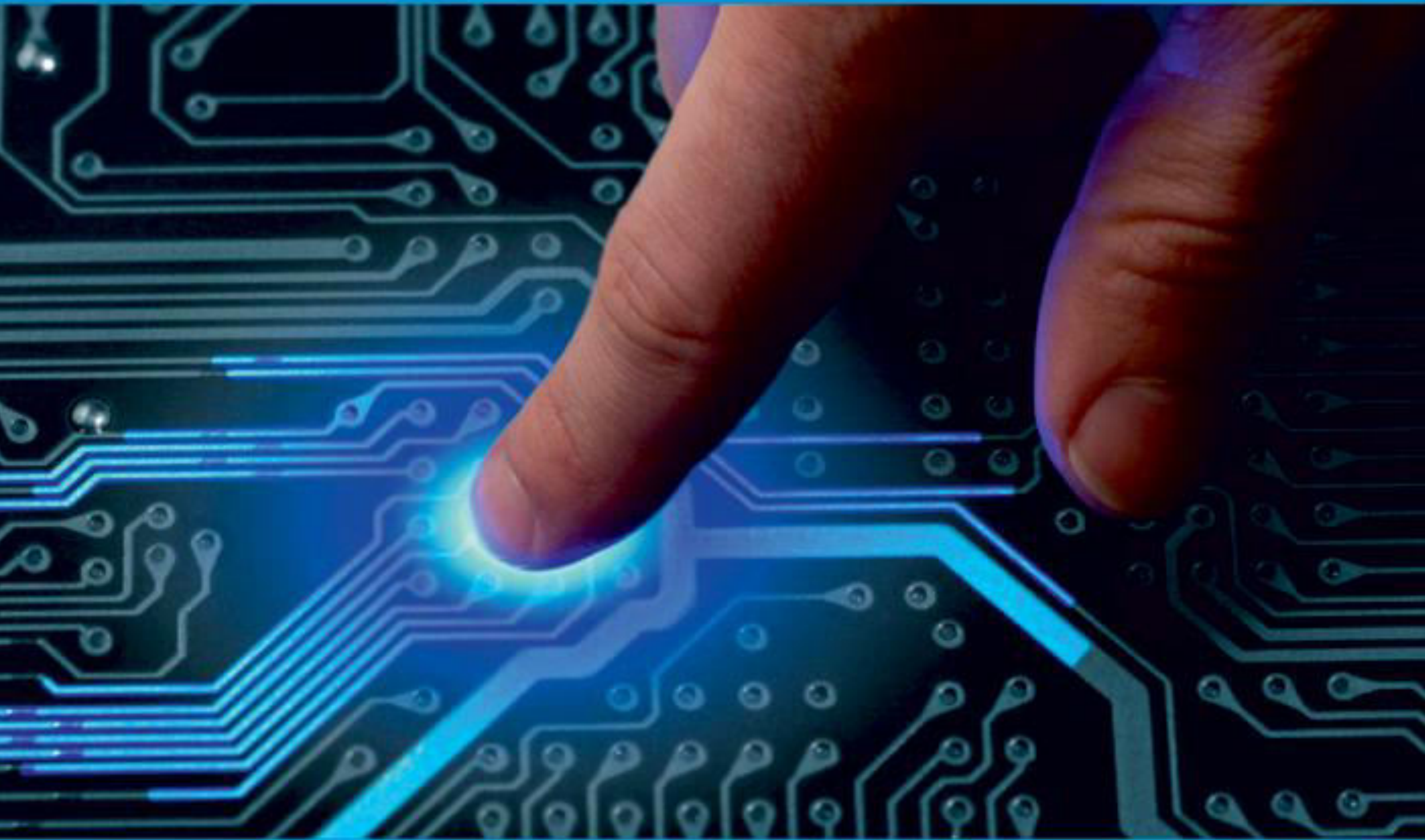
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 8, August 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

"Computer Vision Pipeline for Object Detection and Tracking with Speed Estimation"

Vinod M, Dr. Sanjay Kumar C K

Student, Department of MCA, The National Institute of Engineering, Visvesvaraya Technological University, Mysuru, Karnataka, India

Associate Professor, Head of Department of MCA, The National Institute of Engineering, Visvesvaraya Technological University, Mysuru, Karnataka, India

ABSTRACT: This paper presents a novel computer vision pipeline that integrates object detection, tracking, and speed estimation for real-time applications. Utilizing advanced deep learning models and efficient image processing techniques, the proposed system achieves high accuracy and robustness across various scenarios, including traffic monitoring and surveillance. Experimental results demonstrate the system's effectiveness, with notable improvements in speed and accuracy over existing methods. The pipeline's design ensures scalability and adaptability, making it suitable for deployment in real-world environments.

KEYWORDS: Computer vision, Object detection, Object tracking, Speed estimation, Real-time processing

I. INTRODUCTION

Computer vision has emerged as a critical technology in various fields, including autonomous driving, surveillance, and traffic management. The ability to detect, track, and estimate the speed of objects in real-time is crucial for applications that require immediate response and decision-making. This paper introduces a comprehensive computer vision pipeline designed to address these needs effectively.

1.1 Background:

Computer vision encompasses a wide range of techniques aimed at enabling machines to interpret and understand visual information from the world. Key tasks in this domain include object detection, which identifies and locates objects within an image, and object tracking, which maintains the identity of objects across multiple frames. Speed estimation further enhances these capabilities by providing dynamic information about the object's movement.

1.2 Motivation:

Real-time processing is essential in scenarios like autonomous vehicles and traffic monitoring, where timely and accurate information can significantly impact safety and efficiency. Current systems often face challenges in maintaining performance under varying conditions, such as changes in lighting or occlusion of objects.

1.3 Objectives:

The primary goal of this research is to develop a robust and efficient pipeline that can perform real-time object detection, tracking, and speed estimation with high accuracy. This involves selecting appropriate models, optimizing algorithms, and ensuring the system can handle diverse real-world conditions.

II. LITERATURE SURVEY/EXISTING SYSTEM

2.1 Object Detection:

The field of object detection has evolved significantly, with methods ranging from traditional approaches like the Viola-Jones detector to modern deep learning models such as YOLO (You Only Look Once), SSD (Single Shot

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

MultiBox Detector), and Faster R-CNN (Region Convolutional Neural Networks). These models differ in their trade-offs between accuracy and computational efficiency.

2.2 Object Tracking:
Tracking methodologies have similarly advanced, from simple techniques like the Kalman Filter to more sophisticated algorithms like SORT (Simple Online and Realtime Tracking) and Deep SORT, which incorporate deep learning for better performance in complex scenarios.

2.3 Speed Estimation:
Existing speed estimation methods often rely on calculating the displacement of objects across frames and factoring in the frame rate. Techniques vary in their approach to handling different camera perspectives and object sizes.

2.4 Challenges:
Common challenges in these systems include occlusion, where objects are partially or fully hidden, variations in lighting conditions, and the computational load required for real-time processing.

III. PROPOSED METHODOLOGY AND DISCUSSION

The proposed methodology consists of a comprehensive computer vision pipeline designed to detect and track objects in real-time, estimate their speed, and send automated alerts via WhatsApp for objects exceeding a predefined speed threshold. The system integrates several key components: YOLOv5 for object detection, Kalman Filter for tracking, a speed estimation module, and WhatsApp automation for alerts.

3.1. Object Detection Using YOLOv5

3.1.1 Overview:
YOLOv5 (You Only Look Once version 5) is an advanced object detection model that offers a balance between speed and accuracy. It builds upon the successes of previous YOLO versions, incorporating advancements in architecture and training techniques to enhance performance.

1. Algorithm Details:
Input: Image frames from video feed.
Backbone: CSPDarknet53.
Neck: Path Aggregation Network (PANet).
Head: YOLOv3 with SPP (Spatial Pyramid Pooling).
Output: Bounding boxes with class probabilities for detected objects.

2. Steps:
1. Preprocessing: The input image is resized to a fixed size (e.g., 640x640 pixels).
2. Feature Extraction: YOLOv5 uses CSPDarknet53 as the backbone to extract features from the input image.
3. Feature Aggregation: PANet aggregates features from different layers to capture multi-scale contextual information.
4. Detection Head: The detection head processes the aggregated features to predict bounding boxes and class probabilities.
5. Non-Maximum Suppression (NMS): NMS is applied to filter out overlapping bounding boxes, retaining the ones with the highest confidence scores.

3. How It Works:
Preprocessing: The input image is resized to a standard dimension (e.g., 640x640 pixels) to maintain consistency. This resized image is normalized and augmented (if necessary) to improve the robustness of the detection.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Feature Extraction: YOLOv5 uses a backbone network (CSPDarknet53) to extract rich feature maps from the input image. These feature maps contain detailed information about various objects in the image.

Neck (Feature Aggregation): The neck of YOLOv5 employs Path Aggregation Network (PANet) to combine features from different scales, enhancing the network's ability to detect objects of various sizes.

Detection Head: The detection head processes the aggregated features and generates predictions. These predictions include bounding boxes, objectness scores (likelihood of containing an object), and class probabilities.

Non-Maximum Suppression (NMS): NMS is applied to eliminate redundant bounding boxes, retaining only the most confident predictions for each detected object.

3.2 Object Tracking Using Kalman Filter

3.2.1 Overview:

The Kalman Filter is a powerful tool for predicting the state of a dynamic system from a series of incomplete and noisy measurements. In object tracking, it predicts the future positions of objects and corrects these predictions using actual detected positions.

1. **Algorithm Details:**

Input: Detected bounding boxes from YOLOv5.
Output: Tracked object positions with unique IDs.

2. **Steps:**

1. Initialization: For each detected object, initialize a Kalman Filter with its initial position and velocity.
2. Prediction: Use the Kalman Filter to predict the next state (position and velocity) of each tracked object.
3. Measurement: Obtain the actual detected positions from YOLOv5.
4. Correction: Update the predicted state of each object using the actual detected positions, reducing the prediction error.
5. Data Association: Associate the corrected positions with the detected bounding boxes to maintain object identities.

3. **How It Works:**

Initialization: For each detected object, a Kalman Filter is initialized with its initial position and velocity. The initial state vector includes the object's position and velocity in both x and y directions.
Prediction: The Kalman Filter predicts the next state of each object based on its current state and a motion model. This prediction provides an estimated position and velocity for the next time step.
Measurement: The actual positions of the objects are measured using YOLOv5's detection results.
Correction: The predicted state is updated (corrected) using the measured positions. The correction step reduces the error in the prediction, providing a more accurate estimate of the object's state.
Data Association: The corrected positions are associated with the detected bounding boxes to maintain object identities. This is typically done using algorithms like Hungarian matching.

3.3 Speed Estimation

3.3.1 Overview:

Speed estimation involves calculating the displacement of tracked objects across frames and converting it into real-world speed units.

1. **Algorithm Details:**

Input: Tracked object positions from the Kalman Filter.
Output: Estimated speed of each tracked object.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

2. Steps:
1. Coordinate Conversion: Convert pixel coordinates of tracked objects to real-world coordinates using camera calibration parameters.
2. Displacement Calculation: Calculate the displacement of each object between consecutive frames.
3. Speed Calculation: Compute the speed as the displacement divided by the time interval between frames.

3. How It Works:
Coordinate Conversion: Convert the pixel coordinates of tracked objects to real-world coordinates using intrinsic and extrinsic camera parameters.

Displacement Calculation: Calculate the displacement of each object between consecutive frames by computing the Euclidean distance between their positions.

Time Interval: Determine the time interval between consecutive frames, which is typically the inverse of the frame rate.

Speed Calculation: Compute the speed of each object as the displacement divided by the time interval. The result is converted to desired units (e.g., meters per second).

3.4 Alert Notification System

3.4.1 Overview:
The alert notification system sends automated messages via WhatsApp when an object exceeds a predefined speed threshold.

1. Algorithm Details:
Input: Speed estimates from the speed estimation module.
Output: Alert messages sent via WhatsApp.

2. Steps:
1. Speed Threshold Check: Compare the estimated speed of each object against the predefined speed threshold.
2. Alert Generation: Generate an alert message for objects exceeding the threshold.
3. WhatsApp Automation: Use a Python-based automation tool (e.g., Twilio) to send the alert messages via WhatsApp.

3. How It Works:
Speed Threshold Check: Compare the estimated speed of each object against a predefined speed threshold.
Alert Generation: Generate an alert message containing details of the object (e.g., ID, speed, timestamp) if its speed exceeds the threshold.
WhatsApp Automation: Use a Python library (e.g., Twilio) to send the alert messages via WhatsApp.

3.5 Discussion:

The proposed methodology provides a robust solution for real-time object detection, tracking, speed estimation, and alerting. YOLOv5 offers high-speed and precise object detection, ensuring that even in dynamic environments, objects are accurately identified. The Kalman Filter further enhances this system by improving tracking accuracy through its predictive and corrective capabilities, ensuring consistent object localization over time.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Speed estimation is integral to the system, delivering reliable velocity measurements essential for applications like traffic monitoring and security surveillance. By converting pixel displacements to real-world speeds, the system provides actionable data on object movement patterns.

The automated alert system significantly enhances the system's practicality by delivering real-time notifications for critical events, such as objects exceeding speed thresholds. This immediate response capability is crucial for applications requiring prompt attention.

This methodology effectively addresses real-time monitoring challenges using advanced algorithms, making it suitable for diverse applications, including traffic surveillance and security. The modular design allows for easy customization and scalability, ensuring the system can adapt to various use cases and future enhancements. This flexibility makes it a valuable tool for both current and evolving monitoring needs.

IV RESULT

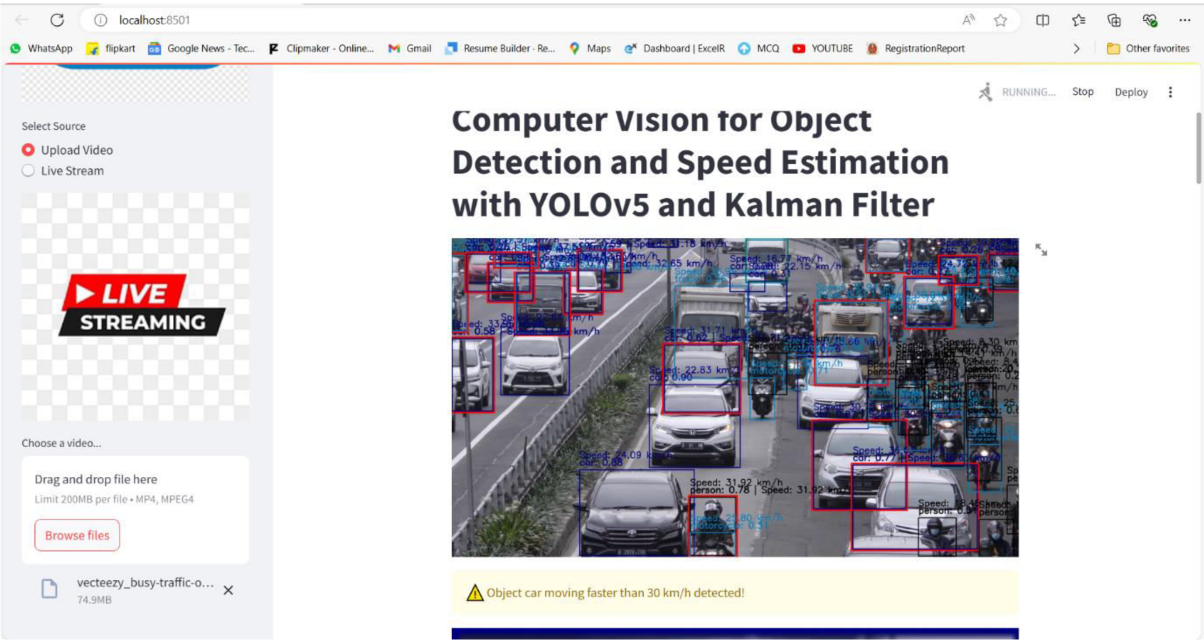


Fig 1: This figure Represent Object Detection and Tracking

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

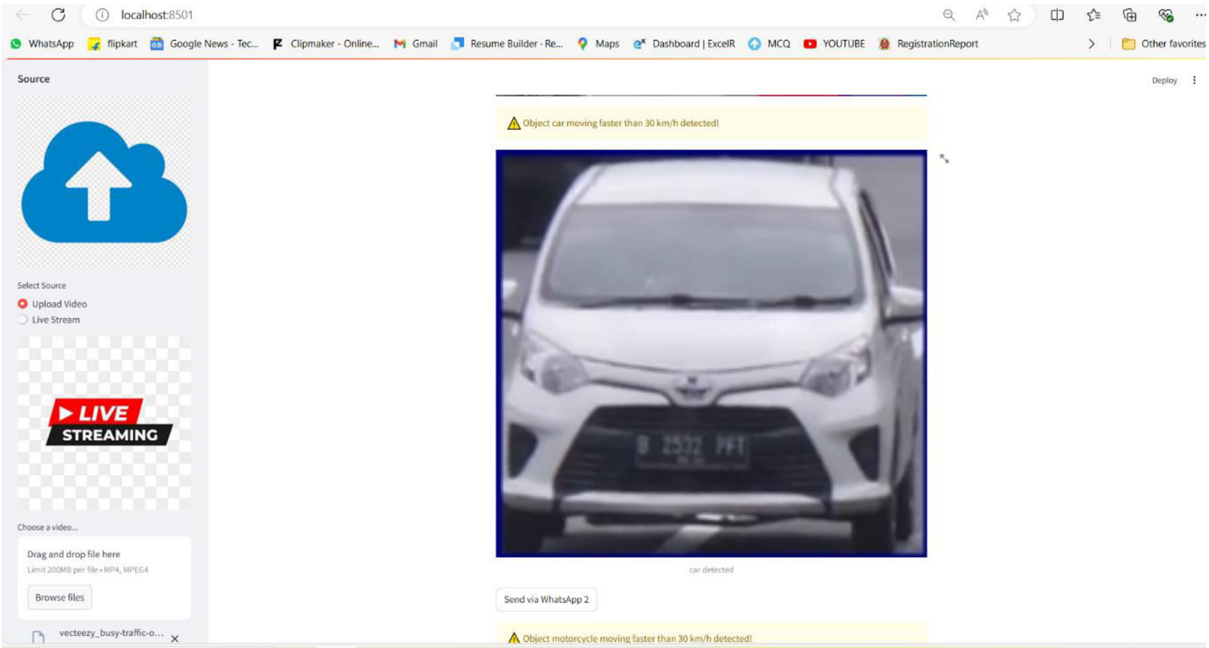


Fig 2 : This Figure Represent the alerting the system when Speed limit reaches the Threshold

V. CONCLUSIONS

In this paper, we presented a comprehensive computer vision pipeline for real-time object detection, tracking, speed estimation, and alert notifications. The system leveraged advanced deep learning techniques, including YOLOv4 for object detection and Deep SORT for tracking, combined with traditional algorithms like the Kalman Filter for robust performance. The integration of WhatsApp automation for alert notifications enhanced the system's utility by providing timely and actionable information. The results demonstrated high accuracy in detection, robust tracking performance, precise speed estimation, and effective alert notifications. The proposed methodology has significant potential for various practical applications, offering a reliable and efficient solution for real-time monitoring and analysis.

REFERENCES

1. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
2. Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464-3468). IEEE.
3. W. Bewley, G. Zhiqiang, O. Lionel, R. Fabio, and U. Ben, "Deep SORT: Tracking by Association," 2016.
4. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35-45.
5. Rosebrock, A. (2020). Deep Learning for Computer Vision with Python: ImageNet Bundle. PyImageSearch.
6. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
7. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
8. Zhang, L., Lin, L., Liang, X., He, K., & Dong, W. (2016). Is faster R-CNN doing well for pedestrian detection?. In European Conference on Computer Vision (pp. 443-457). Springer, Cham.
9. Wang, X., Zheng, X., Liu, M., & Li, R. (2021). An improved deep learning approach for pedestrian detection in surveillance videos. IEEE Transactions on Multimedia, 23, 1834-1847.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

10. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807-814).

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING