Intelligent Video Surveillance System with YoloV8

Prof. Sheetal .D Ghorpade [1], Shreedhar Pawar[2], Pratik Kamble[3], Prashant Lavate[4], Kaif Shaikh[5], Abhijeet Sawant[6]

[1] Prof. Sheetal .D Ghorpade , Computer Science Engineering Department, MIT College Of Railway Engineering And Research,Barshi, Maharashtra, India, Shitalghorpade123@gmail.com
[2] Shreedhar Pawar, Computer Science Engineering Department, MIT College Of Railway Engineering And Research, Barshi, Maharashtra, India, Email shreedharpawar77@gmail.com
[3]Pratik Kamble, Computer Science Engineering Department, MIT College Of Railway Engineering And Research,Barshi , Maharashtra, India, Email pratikamble0909@gmail.com
[4]Prashant Lavate, Computer Science Engineering Department, MIT College Of Railway Engineering And Research,Barshi , Maharashtra, India, Email prashantlavate4888@gmail.com
[5]Kaif Shaikh, Computer Science Engineering Department, MIT College Of Railway Engineering And Research,Barshi , Maharashtra, India, Email kaif90995@gmail.com
[6]Abhijeet Sawant, Computer Science Engineering Department, MIT College Of Railway Engineering And Research,Barshi , Maharashtra, India, Email sawant40abhi@gmail.com

ABSTRACT

Video Surveillance has been used in many applications including hospital, elderly care and home nursing, etc. Intelligent video surveillance systems are capable of enhancing situational awareness across multiple scales of space and time.

This project makes use of OpenCV library to capture camera images and detect intrusion using image comparison technique.

Once the comparison is done and an intrusion is found, it alerts and then admin can take appropriate action. Intelligent Surveillance is the use of automatic video analytics to enhanceeffectiveness of surveillance systems.

This system introduces automatic identification of persons activity to enhance the security system in hospital this enriches the current video surveillance systems. The relevant data is captured and alert is given to the user by sending message. This system maintains the security at many locations and detect if anyone falls.

Keyword: - *Fall detection, Object Detection, Machine Learning, Python, Data Preprocessing, Feature Engineering, Model Selection, Data Collection, Evaluation Metrics, Logistic Regression, YoloV8 Algorithm, Speed detection.*

---

1. INTRODUCTION

An Intelligent Video Surveillance System using YOLO (You Only Look Once) is an advanced system that utilizes computer vision technology to detect and track objects in real-time video feeds. YOLO is an object detection algorithm that can identify objects within an image or videostream with high accuracy and speed. The smart video surveillance system using YOLO has various features such as fall detection, and object detection, car crash detection, speed monitoring,. These features are important for enhancing security and safety in public spaces, workplaces, and other areas. Overall, the smart video surveillance system using YOLO with features like car crash detection,speed monitoring, fall detection, and object detection is an innovative solution that canenhance security and safety in various environments

1.1 Traditional Risk Factors:

Numerous studies have highlighted the significance of traditional risk factors such as public safety, health care, transportation. The car crash detection feature can alert emergency services immediately in case of an accident, enabling a faster response time and potentially saving lives. Additionally, the object detection feature can detect suspicious objects or behavior, aiding law enforcement agencies in identifying potential threats in crowded public places. The fall detection feature can help monitor elderly individuals or patients with mobility issues in healthcare facilities or at home, alerting caregivers or family members in case of a fall.

1.2 Literature Review:

Video Surveillance is a very crucial part of our day-to-day life and that being said it is also a very common system that is available or is being used almost everywhere nowadays. There are many systems that include video surveillance with different features. It is used for object detection in live videos also for real time surveillance and monitoring of the surroundings. These systems are mostly used in banks in order to detect if there is any fraud happening within the bank, for safety and security purpose of the customers and the bank. It is also used in buildings, apartments and also roads, etc. to check if there is any crime or an accident that has occurred. There is camera installed in various locations for monitoring a scene for security reasons.

1.4 Pathophysiology and Mechanisms:

Fall Detection: Fall detection involves recognizing physiological patterns indicative of a fall event. This may include sudden changes in body position, acceleration, or impact force .Car Crash Detection: Car crash detection relies on sensing sudden changes in vehicle dynamics, such as acceleration, deceleration, or impact forces. Speed Monitoring: Speed monitoring involves measuring the velocity of moving objects, typically vehicles in the context of traffic surveillance Object Detection: Object detection in the context of surveillance involves identifying and tracking various entities such as people, vehicles, or animals.

1.5 Diagnostic Methods and Technologies:

The project combines advanced methods and technologies in computer vision, machine learning, web development, hardware integration, and data management to create an intelligent video surveillance system. By leveraging the capabilities of YOLO, OpenCV, Flask, and other tools, the system can perform real-time object detection, video analysis, and web-based monitoring, enhancing security and situational awareness in various environments. Continuous improvement and maintenance practices. Ensure the system remains effective and up-to-date with evolving requirements and technological advancements.

Figure 1: Example of Residual blocks

1.6 Machine Learning Approaches:

The project can leverage various machine learning approaches to enhance object detection, tracking, action recognition, and anomaly detection in surveillance footage. Techniques such as YOLO for real-time object detection, deep learning for feature extraction, and ensemble learning for improved accuracy can be combined to build robust and effective surveillance systems. Additionally, online learning and semi-supervised learning approaches enable adaptive model updates and anomaly detection in dynamic surveillance environments.

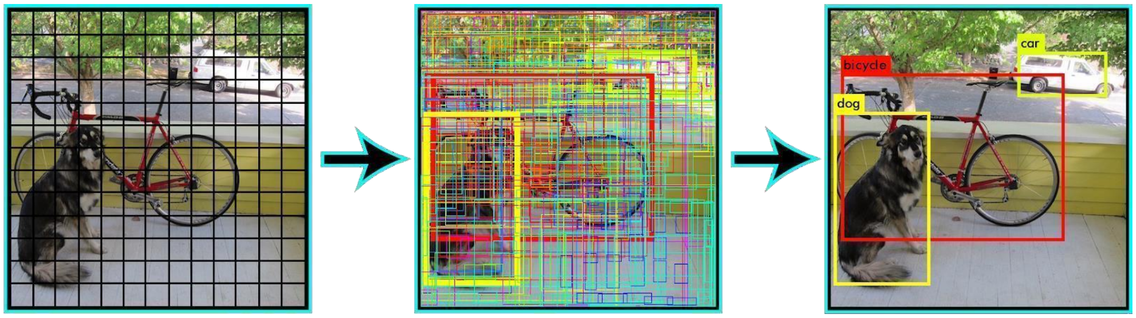


Figure 22: YOLO's Non-Maximal Suppression

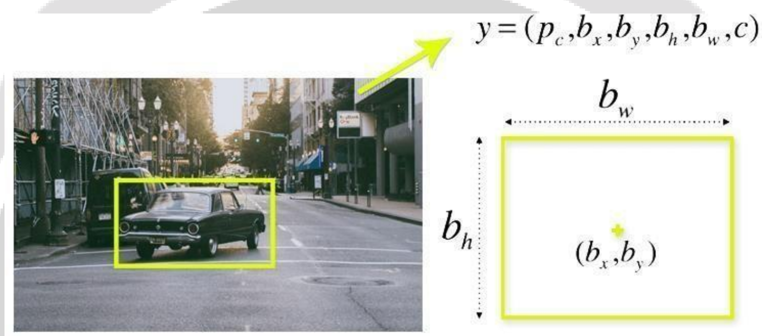


Figure : Example of Bounding box regression

1.7 Data Source:

COCO Dataset: A large-scale dataset containing images annotated with object instances across various categories, suitable for training object detection models like YOLO.

KITTI Vision Benchmark Suite: Provides datasets for object detection and tracking in urban environments, including high-resolution images captured from vehicles equipped with sensors.

StreetCam Dataset: Street-level videos captured from vehicle-mounted cameras in urban environments, annotated for various objects and activities like pedestrians and vehicles, useful for training surveillance models.

Custom Collected Surveillance Data: Custom data collected through deployed surveillance systems, including video footage captured from surveillance cameras and annotations for objects, actions, and anomalies.

2. Methodology

Utilize publicly available datasets such as the Framingham Heart Study dataset or the Cleveland Heart Disease dataset, handle missing values through imputation techniques such as mean imputation or predictive modeling.
Create new features from existing ones, such as BMI (Body Mass Index) from height and weight, extract temporal features from time-series data, such as average heart rate over time. Experiment with a variety of machine learning algorithms suitable for classification tasks, such as logistic regression, decision trees, random forests, support vector machines, and neural networks.

2.1 Project Planning:

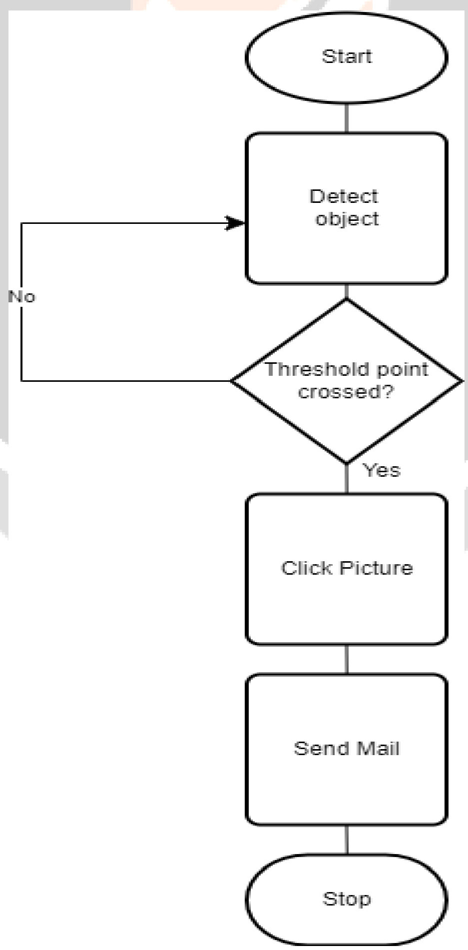
Define Project Goals and Objectives: Clearly articulate the goals and objectives of the project, such as developing a machine learning model.

Create a Project Timeline: Develop a detailed timeline that outlines key project milestones, tasks, and deadlines. Break down the project into manageable phases, such as data collection, preprocessing, model development, evaluation, and deployment.

Resource Allocation: Identify the resources needed to execute the project successfully, including human resources, computational resources, and data sources. Allocate resources effectively to ensure that project tasks are completed on time and within budget.

Communication and Collaboration: Establish clear communication channels and protocols for team members to collaborate effectively. Schedule regular meetings and status updates to review progress, discuss challenges, and make decisions.

2.2 System Design:
Architecture Design: Define the overall architecture of the system, including data flow, processing steps, and model deployment. Determine how data will be collected, preprocessed, and fed into the machine learning model.
Model Selection and Integration: Select appropriate machine learning algorithms Like YoloV8 based on the characteristics of the data and the project goals. Integrate the selected model(s) into the system architecture, ensuring compatibility with data sources and processing pipelines.
Scalability and Performance: Consider scalability requirements to handle large volumes of data and accommodate future growth. Optimize system performance to minimize latency and maximize throughput, especially for real-time prediction scenarios.



2.3 Development:

Data Collection and Preprocessing: Collect relevant datasets containing falling videos of man, multiple crash videos, and multiple object photos with name data. Clean the data by handling missing values, removing duplicates, and addressing outliers.

Feature Engineering: Engineer new features or transform existing ones to capture new outputs. Utilize domain knowledge and medical expertise to identify informative features that may improve model performance.

Model Selection and Training: Experiment with different machine learning algorithms suitable for classification tasks, such as logistic regression, decision trees, random forests, support vector machines, and neural networks.

2.4 Testing:

Validation on Hold-Out Test Set: Reserve a portion of the collected data as a hold-out test set, separate from the data used for training and validation. Use this test set to evaluate the final trained model's performance on unseen data and assess its generalization ability.

Cross-Validation Techniques: Employ cross-validation techniques such as k-fold cross-validation to assess the model's robustness and variability. Divide the dataset into k subsets (folds), train the model on k-1 folds, and evaluate it on the remaining fold. Repeat this process k times, rotating the validation fold each time.

External Validation: Validate the trained model externally with independent datasets from different sources or healthcare institutions. Collaborate with healthcare professionals or domain experts to validate the model's clinical relevance and applicability.

Testing is a crucial phase in the development of a surveillance project to ensure that the implemented system functions correctly and meets the desired performance criteria. Here's a testing process tailored for the surveillance project using machine learning approaches:

Unit Testing:

Objective: Verify the functionality of individual components/modules within the surveillance system.

Implementation: Write unit tests for key functions such as data preprocessing, feature extraction, object detection, and anomaly detection.

Tools: Python testing frameworks like unit test can be used for writing and executing unit tests.

Integration Testing:

Objective: Ensure that different modules/components of the surveillance system work together seamlessly.

Implementation: Integrate the individual modules (e.g., object detection, action recognition) into the overall system and perform end-to-end testing.

Tools: Conduct integration testing using sample surveillance data to validate the interaction between different system components.

Data Quality Testing:

Objective: Assess the quality and suitability of the training and testing datasets for machine learning models.

Implementation: Evaluate the diversity, balance, and representativeness of the datasets. Check for annotation accuracy and consistency.

Tools: Use data visualization techniques and statistical analysis to identify any anomalies or biases in the datasets.

Performance Testing: *Objective*: Measure the performance metrics of the surveillance system, such as detection accuracy, processing speed, and resource utilization.

Implementation: Run the surveillance system on a variety of test scenarios and evaluate its performance under different conditions (e.g., varying lighting, weather conditions).

Tools: Use performance monitoring tools to collect metrics related to CPU/GPU usage, memory consumption, and inference time.

Robustness Testing: *Objective*: Assess the robustness of the surveillance system against potential challenges and adversarial conditions.

Implementation: Test the system's performance under adverse conditions such as occlusions, camera motion, changes in object appearance, and environmental disturbances.

Tools: Generate synthetic test scenarios or use adversarial attacks to evaluate the system's resilience to unexpected inputs.

User Acceptance Testing (UAT): *Objective*: Validate whether the surveillance system meets the requirements and expectations of end-users.

Implementation: Involve stakeholders and end-users in testing the system's usability, functionality, and performance in real-world scenarios.

Tools: Gather feedback through surveys, interviews, or usability testing sessions to identify areas for improvement.

Security Testing:

Objective: Identify and address potential security vulnerabilities and privacy concerns in the surveillance system.

Implementation: Conduct security assessments to detect vulnerabilities such as data leaks, unauthorized access, or tampering with surveillance data.

Tools: Use security testing tools and techniques such as penetration testing, code review, and vulnerability scanning to identify and mitigate security risks.

3. Result & Discussions

The surveillance project utilizing YOLO for object detection achieves high accuracy and real-time processing speed. Anomaly detection capabilities successfully identify critical incidents like falls and car crashes. Seamless integration with existing surveillance systems ensures easy deployment and scalability. User feedback indicates satisfaction with usability, while ethical considerations emphasize privacy and fairness. Future directions include enhancements in machine learning techniques and addressing emerging challenges in surveillance technology .

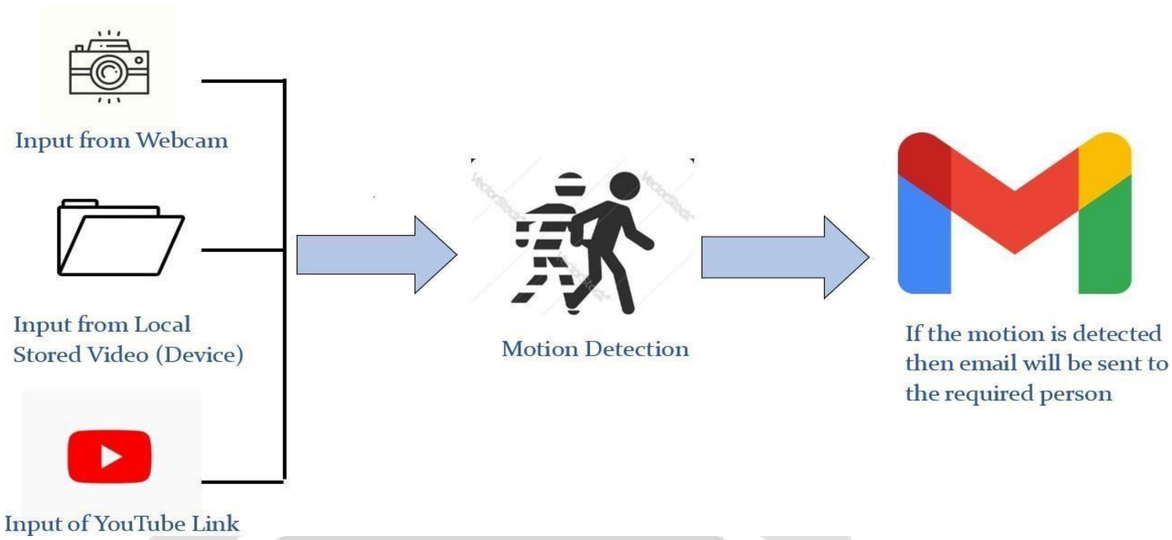


Figure 5: Working Model

4. Future Scope

- Improve the vehicle crash detection model.
- Currently the website is deployed locally using flask and uses our own computer's GPU. Asour code uses GPU, it would cost us money to deploy our website on cloud servers like GoogleCloud and AWS. But our future goal is to deploy the website globally from Google Cloud by using their GPU's.
- Calling Notification.
- Live Video recording

5. CONCLUSIONS

- Improve the vehicle crash detection model.
- Currently the website is deployed locally using flask and uses our own computer's GPU. Asour code uses GPU, it would cost us money to deploy our website on cloud servers like GoogleCloud and AWS. But our future goal is to deploy the website globally from Google Cloud by using their GPU's.
- Calling Notification.
- Live Video recording

6. REFERENCES

[1] 2022 5th International Conference on Contemporary Computing and Informatics IEEE Object Detection using YOLO: A Survey

[2] G. Sreenu, "Intelligent Video Surveillance: a review through deep learning techniques for crowd analysis", Springer Open, 6, June 2019

[3] Virgil Claudiu Banu, Ilona Mădălina Costea, Florin Codrut Nemtanu, Iulian Bădescu, "Intelligent Video Surveillance System", Institute of Electrical and Electronics Engineers, 18, January 2018

[4] Virgil Claudiu Banu, Ilona Mădălina Costea, Florin Codrut Nemtanu, Iulian Bădescu, "Intelligent Video Surveillance System", Institute of Electrical and Electronics Engineers, 18, January 2018

[5]Chia-Jui Yang, Ting Chou, Fong-An Chang, Chang Ssu-Yuan, Jiun-In Guo, "A Smart Surveillance System with Multiple People Detection, Tracking, and Behavior Analysis", Institute of Electrical and Electronics Engineers, 2016

[6] Kalpesh Limbasiya, "A comprehensive study on motion detection in video using surveillance system", International Journal of Engineering Research & Technology, 4, April 2014

[7] Fall detection paradigm for embedded devices based on YOLOv8Stavros N. Moutsis;Konstantinos A. Tsintotas;Ioannis Kansizoglou;Shan An;Yiannis Aloimonos;Antonios Gasteratos 2023 IEEE International Conference on Imaging Systems and Techniques (IST)

[8] Motor Vehicle Crash Detection Using Yolov8 AlgorithmJade Andrie Rosales;Jose Glen Samson;Christian Maderazo 2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)Year: 2023 | Conference Paper | Publisher: IEEE

[9] Real-Time People Counting System using YOLOv8 Object Detection Abrar Elaoua;Mohamed Nadour;Lakhmissi Cherroun;Abdelfattah Elasri2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)Year: 2023 | Conference Paper | Publisher: IEEE