International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 02 February 2025
Page No: 240-247
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0034

AI Based Real Time Object Detection and Analysis of Multiple Object Existence

Dhivya S[1], Dhanusya P[2], PriyaDharshini N[3], Mr. D. Vijayakumar B.E, M. S[4]
[1,2,3]Student, Department of Computer Science and Engineering, National Engineering College, Kovilpatti-India.
[4]Assistant Professor (SG), Department of Computer Science and Engineering, National Engineering College, Kovilpatti-India.
Emails: *2112045@nec.edu.in[1], 2112057@nec.edu.in[2], 2112114@nec.edu.in[3], dvk@nec.edu.in[4]*

Abstract
Accurate and efficient object detection and counting are crucial in modern computer vision, especially for real-time applications. However, existing methods often struggle with challenges like processing delays, low accuracy in complex environments, and difficulty in detecting overlapping objects. To address these drawbacks, this project proposes an AI-based solution using the YOLO (You Only Look Once) framework, integrated with OpenCV for enhanced image processing. Our system processes video streams as input, performing real-time object detection and counting with high accuracy. It ensures reliable performance under challenging conditions, such as low lighting, cluttered environments, and partial occlusion. OpenCV optimizes input data through noise reduction and image sharpening, improving detection precision.
Keywords: *Object Detection, Object Counting, Open CV*

1. Introduction
The rapid development of AI and computer vision technologies has introduced new possibilities for real-time object detection across industries. This project focuses on utilizing these advancements to build a robust system that can detect, classify, and analyze multiple objects in real-time, even under challenging conditions such as occlusions, varying lighting, and crowded environments. The need for accurate and efficient object detection has become increasingly important in industries like logistics, retail, and security. In these sectors, tracking multiple objects simultaneously while maintaining real-time analysis provides significant operational benefits, including improved decision-making, reduced human error, and enhanced safety measures. Traditional object detection methods, which rely heavily on manual processes or outdated technology, struggle to adapt to the complexities of modern environments, where dynamic interactions between objects frequently occur. Object detection plays a critical role in applications such as inventory tracking, surveillance, and theft prevention, particularly in dynamic environments like warehouses and supermarkets. In warehouse management, for instance, the ability to identify and count items automatically saves time and reduces labor costs. Similarly, in surveillance systems, recognizing abnormal behavior, such as theft or unauthorized access, is essential for maintaining security. These tasks require a detection system capable of processing large amounts of data quickly and accurately, even in suboptimal conditions. The goal of this project is to build a scalable and adaptable system that can handle multiple object types and provide accurate tracking and analysis in real-time. The system leverages the YOLO (You Only Look Once) model, which is highly regarded for its speed and accuracy in detecting objects in complex scenes. YOLO's real-time capabilities stem from its unique architecture, which divides images into a grid and predicts bounding boxes and class probabilities for multiple objects in a single pass through the network. This approach drastically reduces processing time compared to other methods, making it ideal for real-time applications. In conjunction with YOLO, OpenCV (Open Source Computer Vision Library) is

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 02 February 2025
Page No: 240-247
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0034

used for image preprocessing and manipulation. OpenCV's capabilities in handling video feeds and performing image transformations, such as resizing, edge detection, and noise reduction, help optimize the input data for YOLO's object detection tasks. By preprocessing the frames with OpenCV, the system can better cope with challenging conditions, such as poor lighting, shadows, or cluttered backgrounds. These optimizations ensure that the object detection system remains robust and reliable in various real-world scenarios. [1]

2. Related Work

Object detection technologies have significantly evolved, with models such as YOLO (You Only Look Once) leading the charge due to their speed, precision, and ability to process multiple objects simultaneously. Traditional object detection methods, such as sliding window and region-based methods, were computationally expensive and often slow in detecting objects in real-time applications. YOLO revolutionized this field by treating object detection as a regression problem and processing the entire image in one pass, rather than focusing on individual regions. This innovative approach allows YOLO to detect objects quickly without compromising accuracy, making it highly suitable for real-time applications like surveillance, autonomous vehicles, and inventory management.Several studies have explored the use of YOLO for multi-object detection in various industries. These studies demonstrate the model's versatility in addressing diverse challenges such as detecting objects in cluttered environments, handling occlusions, and maintaining accuracy in varying lighting conditions. For instance, Mohandoss and Rangaraj (2024) proposed a system that utilizes an enhanced YOLOv2 model in conjunction with LuNet for improving detection capabilities in video surveillance. Their system demonstrated an ability to detect objects in low-visibility scenarios, such as during nighttime or in poorly lit environments. By integrating the LuNet algorithm, the model was also able to detect smaller objects, which are typically more challenging to identify in crowded scenes(finalpaper1). Similarly, Shen and Liu (2024) conducted an in-depth analysis of YOLO's applications in object detection, focusing on its real-time processing capabilities. Their research highlighted YOLO's strengths in balancing speed and accuracy, particularly in dynamic environments such as autonomous driving. Shen and Liu emphasized that YOLO's ability to process multiple objects simultaneously in a single frame made it superior to traditional detection methods like R-CNN (Regions with Convolutional Neural Networks) or Fast R-CNN, which required multiple passes through the image for object detection. They also noted YOLO's efficient use of computational resources, which is critical in applications where low latency and high throughput are required, such as drone-based surveillance or traffic monitoring(finalpaper1). The real-time capabilities of YOLO have been further enhanced with newer iterations like YOLOv3 and YOLOv4. These models introduced several improvements, such as using Darknet-53, a more advanced neural network architecture, which allows for better feature extraction. Additionally, the introduction of multi-scale prediction in YOLOv3 enables the detection of both large and small objects with higher accuracy. This feature is particularly beneficial in applications like autonomous driving, where objects of varying sizes (pedestrians, vehicles, road signs) must be detected simultaneously and in real-time to ensure safety. One of the most significant challenges in object detection is handling environmental variability, such as changes in lighting, occlusion of objects, and the dynamic nature of real-world settings. In warehouse environments, for example, the lighting conditions can change drastically, from well-lit areas to darker corners where inventory is stored. Occlusion is another common challenge, where objects overlap or partially obscure one another, making it difficult to detect each object accurately. YOLO's real-time detection capabilities, coupled with techniques such as anchor boxes, have shown to be effective in dealing with these challenges. Anchor boxes allow YOLO to predict multiple bounding boxes for objects of different shapes and sizes, making it adaptable to complex scenes. In the context of autonomous vehicles, the need for rapid and accurate object detection cannot be overstated. Self-driving cars must be able to detect and classify a variety of objects, including other vehicles, pedestrians, cyclists, and obstacles, while traveling at high speeds. YOLO's speed advantage comes into play here, allowing vehicles to process images and make split-second decisions that could prevent accidents. Studies have shown that YOLO can outperform other object detection models like SSD (Single Shot Multibox Detector) in terms of processing time, making it ideal for real-time applications in the automotive industry. Moreover, the model's ability to detect objects across multiple scales and its robustness against environmental noise, such as glare or rain, further enhances its applicability in autonomous driving. Beyond traditional applications like surveillance and autonomous vehicles, YOLO has been used in innovative ways across various domains. For instance, in the field of agriculture, YOLO has been applied for real-time monitoring of crops and livestock. By integrating YOLO with drone technology, researchers have developed systems capable of detecting diseased plants, counting livestock, and even monitoring crop growth in large fields. The model's ability to process high-resolution images captured by drones allows farmers to manage large-scale agricultural operations efficiently. In retail and inventory management, object detection is critical for automating processes such as inventory tracking, customer behavior analysis, and theft prevention. Traditional methods of inventory management often involve manual counting, which is time-consuming and prone to human error. By deploying a real-time object detection system powered by YOLO, retailers can automate the inventory management process, ensuring that stock levels are monitored continuously without the need for manual intervention. This not only saves time but also improves the accuracy of stock counts. Additionally, in high-traffic environments like supermarkets, YOLO can be used to analyze customer movement and product interaction, providing valuable data for optimizing store layout and enhancing customer experience. In recent years, YOLO has also been combined with other advanced techniques like deep learning and reinforcement learning to create more intelligent systems. For instance, researchers have integrated YOLO with reinforcement learning algorithms to develop smart surveillance systems capable of not only detecting objects but also predicting potential security threats based on observed behaviors. This type of predictive analysis is crucial for proactive security measures in places like airports, public transportation hubs, and large public events. These systems can analyze the behavior of individuals in real-time, flagging suspicious activities and alerting security personnel before a potential incident occurs. Despite its many advantages, YOLO is not without its limitations. One of the challenges faced by YOLO is its relatively lower accuracy when compared to models like Faster R-CNN, especially when detecting smaller objects or objects in very crowded scenes. This limitation arises from YOLO's approach of dividing the image into a grid and assigning one object per grid cell. In crowded scenes where multiple objects fall within the same grid cell, YOLO may struggle to distinguish between them accurately. However, researchers are continuously working on enhancing the model's performance in such scenarios. For example, YOLOv4 and YOLOv5 have introduced several optimizations, such as the use of Cross Stage Partial Networks (CSPNet) to improve feature propagation and reduce computation, resulting in better accuracy for smaller objects. In conclusion, object detection technologies have come a long way, with models like YOLO leading the way in terms of speed, precision, and real-time processing capabilities. YOLO's versatility makes it suitable for a wide range of applications, from surveillance and inventory management to autonomous driving and agriculture. As the model continues to evolve, with improvements in handling small objects, occlusions, and environmental variability, it is poised to remain at the forefront of object detection technology. Future research will likely focus on integrating YOLO with emerging technologies like edge computing and 5G networks, further enhancing its real-time processing capabilities and expanding its applicability across industries. [2-5]

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 02 February 2025
Page No: 240-247
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0034

3. Proposed Methodology

3.1 Video Capture and Information Gathering

We concentrate on obtaining pertinent video footage at this point in order to train and test the YOLOv8 object detection and tracking algorithm. Video sources are gathered from publicly accessible repositories like YouTube and from specialist datasets like Open Images, KITTI, and COCO that are designed for object detection applications. These movies provide the diversity required to improve model resilience because they feature a wide range of scenarios, object types, and climatic variables. The model must be trained on this varied dataset in order to handle varying illumination, angles, item densities, and backdrops. The gathered movies have been pre-labeled for the particular purpose of object tracking and detection, offering ground truth information for supervised learning and model assessment.

3.2 Data Preprocessing

In this stage, raw video data is prepared for input into the YOLOv8 model. Initially, each video is divided into individual frames, with the frame rate selected carefully to ensure temporal consistency. Several preprocessing steps are applied to each frame:

- **Resizing:** Frames are resized to match the YOLOv8 input dimensions, typically 640x640, to ensure a uniform input size.
- **Normalization:** Pixel values are normalized to a range of 0 to 1, improving model stability and convergence during training.
- **Augmentation:** Data augmentation techniques such as random cropping, color jittering, and horizontal flipping are applied to increase dataset variability. These transformations mimic real-world variations, helping the model generalize better on unseen data.
- **Frame Extraction:** To reduce computational load, frames are extracted at regular intervals based on the video's frame rate. This approach ensures the retention of important temporal features while minimizing redundancy ensure temporal consistency several preprocessing

3.3 Model Building

The YOLOv8 model is chosen for this project because of its cutting-edge object identification capabilities and real-time inference capabilities. Because it effectively balances speed and accuracy, the design of YOLOv8 is especially well-suited for this purpose. It accomplishes this by utilizing a combination of attention mechanisms and convolutional neural networks (CNNs) to recognize objects in pictures or video frames. Large-scale datasets like COCO provide pre-trained weights that are used to initialize the model. By using previously acquired feature representations, the model can leverage transfer learning to shorten the training time required to adjust to the particular dataset. ByteTrack is implemented alongside YOLOv8 to manage the tracking of identified objects. High performance in multi-object tracking (MOT) activities is a well-known attribute of ByteTrack, linking

3.4 Training and Testing Models

Using the preprocessed video frames as input, the YOLOv8 model is trained by fine-tuning it with transfer learning. Training is done using a proprietary dataset that is unique to the application domain (traffic surveillance, retail settings, etc.). The goal is to minimize the loss function, which takes into account both localization and classification loss, to make sure the model can precisely forecast bounding box coordinates in addition to reliably detecting objects. Depending on the features of the dataset, the model is optimized during training using methods like Adam optimizer and stochastic gradient descent (SGD) with momentum. The learning rate is dynamically changed using a learning rate scheduler, and early stopping is applied to minimize overfitting. Important performance indicators, such as mean Average Precision, F1-score, recall, and precision

3.5 Real-time Prediction

Upon successfully completing the training and testing phases, the YOLOv8 model is deployed for real-time object detection and tracking in live video streams. The model processes input from recorded or live video feeds, identifies objects in each frame, and draws bounding boxes around them. Byte Track is

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 02 February 2025
Page No: 240-247
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0034

employed to assign unique IDs to each object, enabling continuous tracking across multiple frames. This tracking remains robust even when objects overlap, exit the frame and re-enter, or are partially occluded. The system can handle video inputs from various sources, including security cameras, mobile devices, and video files. Thanks to its real-time capabilities, the system provides near-instantaneous feedback with minimal latency between video input and output, making it suitable for time-sensitive applications. Figure 1 shows System Architecture

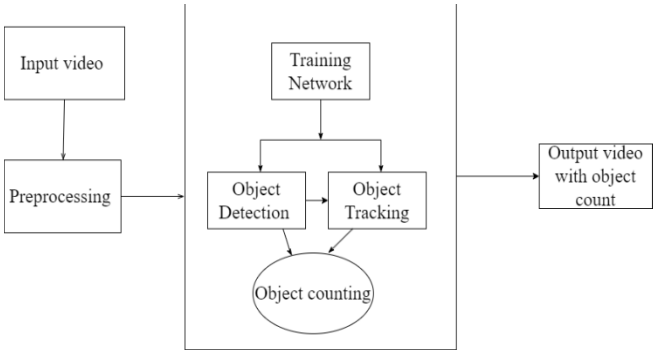

Figure 1 **System Architecture**

Figure 1 shows represents a video-based object detection and counting system, suitable for applications like traffic monitoring, surveillance, and crowd analysis. It starts with an input video that undergoes preprocessing to enhance the quality for further analysis. After preprocessing, the video is fed into a trained network, which detects objects using a model, typically a neural network trained on relevant data. The object detection stage identifies objects of interest in each frame, while the object tracking component follows these detected objects across consecutive frames. This ensures continuity and helps maintain accurate object identification. Both detection and tracking feed into the object counting module, which keeps track of how many objects are present or moving within the video. [7]

3.6 Tracking and Counting Algorithm

The system includes a counting method based on object tracking in addition to real-time detection. Real-time counting is done on items that move through certain zones, such as traffic lanes, retail entrances, and entry and exit points. The tracked object IDs produced by ByteTrack serve as the foundation for the counting method, which guarantees that each object is tallied just once—regardless of how many times it passes through the area. The layout of the counting mechanism allows it to handle various scenarios [6]

- **Crossing Line Counters:** When an object crosses a designated line (such as a doorway or lane) the count is increased.
- **Region-based Counting:** To ensure that the system manages larger areas without double counting, the algorithm keeps track of items entering and leaving preset areas for larger zones.

3.7 Performance Evaluation and Optimization

In the final stage, the performance of the object recognition, tracking, and counting system is evaluated using various metrics to ensure high accuracy and real-time efficiency. The performance assessment includes:

- **Detection Metrics:** Mean Average Precision (MAP) is used to measure the accuracy of object detection across different classes. Additional metrics such as precision, recall, and F1-score further demonstrate the system's reliability in identifying objects while minimizing false positives and false negatives.
- **Tracking Metrics:** The effectiveness of Byte Track in maintaining object identities over time is assessed using Multi-Object Tracking Precision (MOTP) and Multi-Object Tracking Accuracy (MOTA). These metrics evaluate how well the system tracks objects across frames, even in complex scenes.
- **Counting Accuracy:** The counting mechanism is evaluated by comparing the expected versus actual object counts across different scenarios. The system is designed to minimize instances of undercounting or over counting, ensuring accurate object counts in real-world applications.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 02 February 2025
Page No: 240-247
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0034

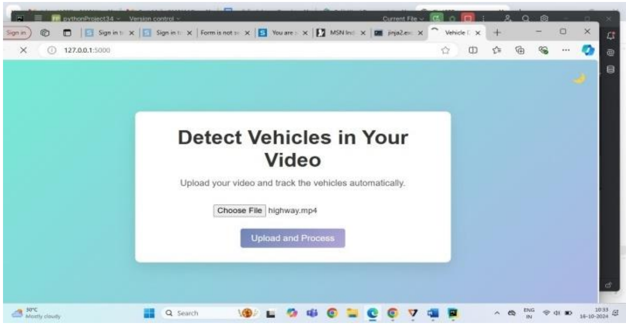

Figure 2 UI Interface for System

Figure 2 shows a user interface of a web application designed for vehicle detection and tracking in video files. Users can upload a video using the "Choose File" button and initiate processing by clicking "Upload and Process." The design is clean and modern, with a simple layout and a gradient background, ensuring ease of use. The application allows users to upload a video, which is then processed by the backend to detect and track vehicles automatically. This interface is part of your final year project, showcasing a functional and visually appealing solution for vehicle detection.

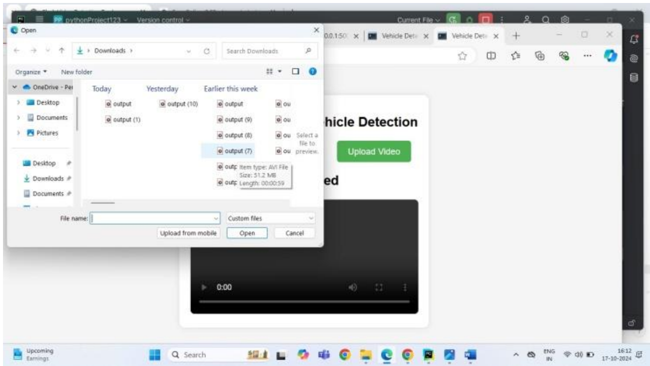

Figure 3 Choose an Input Video

Figure 3 shows an critical step in the workflow of a vehicle detection system, part of a final-year project. The interface presented is for uploading video files to analyze vehicle movements in real-time. The user is interacting with the file upload dialog, selecting an AVI video file named “output (7)” from the local system’s Downloads folder. The video file has a size of 51.2 MB and a duration of 59 seconds. Once uploaded, the system processes the video and displays the results below the upload section. The goal of this project is to detect and analyze vehicles within video streams, possibly using object detection models like YOLO to ensure accurate and efficient detection.

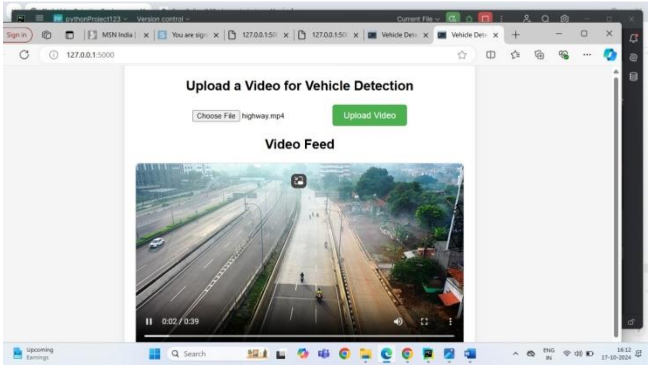

Figure 4 Input Video Displayed in the UI Interface

Figure 4 shows an user interface of a vehicle detection system, developed as part of a final-year project. The interface allows users to upload a video for processing, demonstrated here with the selection of "highway.mp4." Once uploaded, the video feed is displayed below the upload section, showing traffic activity on a highway, including vehicles and motorbikes. The system processes the video in real-time to detect and analyze vehicles, potentially using advanced object detection models such as YOLO. The interface is hosted locally on 127.0.0.1:5000, indicating it is running on a development server, likely powered by a Python web framework like Flask. This project aims to provide a practical solution for monitoring and analyzing traffic, with applications in smart transportation and traffic management systems. [8]

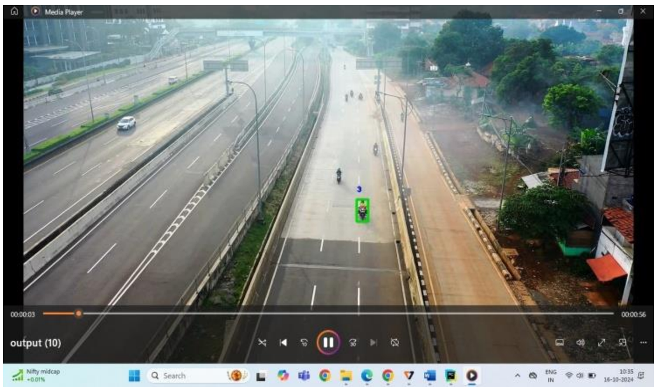

Figure 5 Object Detection and Tracking

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 02 February 2025
Page No: 240-247
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0034

Figure 5 shows an output of a vehicle detection and tracking system from my final year project. The video captures a highway where moving vehicles are automatically detected and tracked. A motorbike is highlighted with a green bounding box and labeled with the identifier "3," demonstrating the system's ability to track multiple vehicles. The project uses computer vision to monitor vehicles in real-time, contributing to traffic monitoring and intelligent transportation systems. [9-10]

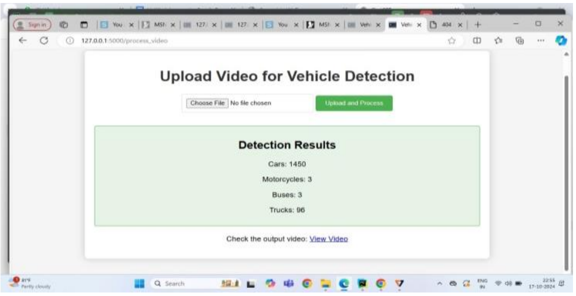

Figure 6 Detection Results

Figure 6 shows an web-based Vehicle Detection System with a simple, user-friendly interface that allows users to upload videos for automated vehicle detection and counting. The interface includes a "Choose File" button to select a video and an "Upload and Process" button to initiate detection. Once processed, the system displays the results in a green-highlighted box, categorizing vehicles into Cars (1450), Motorcycles (3), Buses (3), and Trucks (96). A link labeled "View Video" at the bottom allows users to access the output video with detected vehicles. The system is running locally on 127.0.0.1:5000, indicating it is in a development environment.

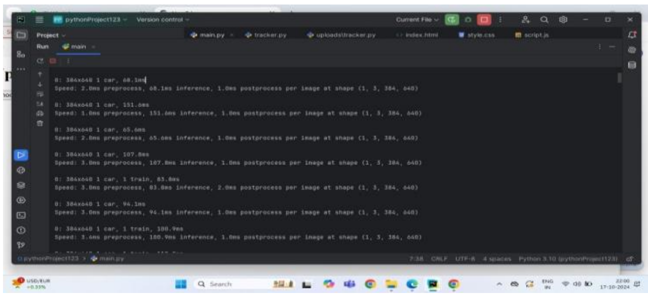

Figure 7 Python Model Inference Performance Output

Figure 7 shows a Python development environment (VS Code) with a terminal output displaying inference performance results from an object detection or image processing task. The results log inference times, preprocessing durations, and post-processing times for images processed at a resolution of 384x640 pixels. Detected objects include cars and trains, and the speed breakdown shows preprocessing, inference, and post processing times for each image.

Conclusion

The integration of YOLO for real-time object detection and OpenCV for image preprocessing results in a scalable, adaptable system that provides real-time object analysis across multiple industries. The automation of object detection not only enhances operational efficiency but also significantly reduces human errors, improving decision-making and safety measures. Whether for warehouse management, retail, or security applications, this system delivers a comprehensive solution for modern object detection challenges. The system is powered by the YOLO (You Only Look Once) model, renowned for its speed and accuracy. YOLO’s unique architecture divides the image into a grid and predicts bounding boxes and class probabilities for multiple objects in a single pass, making it ideal for real-time applications. This method significantly reduces processing time while maintaining high detection accuracy, even in complex scenes. In conjunction with YOLO, OpenCV (Open Source Computer Vision Library) plays a crucial role in image preprocessing. OpenCV’s image manipulation capabilities—such as resizing, edge detection, and noise reduction—optimize video input for YOLO’s object detection tasks. This preprocessing enables the system to handle suboptimal conditions, such as poor lighting or cluttered environments, ensuring robust and reliable performance.

References

[1]. Jin Yao, Weitao Chen ,13 June 2024, Automatic Counting Method for Centipedes Based on Deep Learning, IEEE, The paper enhances YOLOv5 into FD-YOLO for centipede detection, featuring CBAM for focus on centipedes, BiFormer for better feature extraction, and SPPFCSP for

improved multi-scale handling and robustness.

[2]. Ziming Wang b , Huixiang Liu a , 08 June 2022, Trainable Spiking-YOLO for low-latency and high-performance object detection, Science Direct .The paper introduces Tr-Spiking-YOLO, which combines Spiking Neural Networks with YOLO for enhanced object detection, featuring efficient SNN integration, a Feature Pyramid Network for better small object detection, and an optimized decoding scheme for balancing accuracy, energy, and time efficiency.

[3]. Jieyu Chen, Zhenghao Xi, 23 December 2020, Multiple Object Tracking Using Edge Multi-Channel Gradient Model With ORB Feature, IEEE, The paper focuses on Tracking by Detection in Multi-Object Tracking (MOT), emphasizing deep learning advancements for appearance similarity and motion integration, while also acknowledging the importance of non-deep learning methods like KLT and RFBNet for their robustness.

[4]. Zhihao Wu, Jie Wen, 08 June 2022, Enhanced Spatial Feature Learning for Weakly Supervised Object Detection, IEEE. The paper focuses on the SLE method, which introduces feature grouping into pooling operations to capture multiple scales without extra parameters

[5]. Yunhe1, Junjie Hu1, 8 April 2024, The Efficient Dual-Channel Bottleneck Structure YOLO Detection , IEEE, The paper presents DCGC-YOLO, an enhanced YOLOv5 model for fire detection that improves accuracy and efficiency using Dual Channel Group Convolution (DCGC). This structure optimizes feature extraction and information encoding.

[6]. Luyl-Da Quach 1 , Khang Nguyen Quoc , 9 January 2024 ,Tomato Classification, Detection, and Counting System Based on YOLOv5, IEEE ,The study presents a Tomatoes Health Check System using MobileNet models and Grad-CAM++ for detecting, classifying, and counting tomatoes. The research also integrates YOLOv5 and the SORT algorithm for real-time tracking in videos.

[7]. T. Mohandoss,J.Rangaraj . June 2024 . Multi-Object Detection using Enhanced YOLOv2 and LuNet Algorithms in Surveillance Videos . ScienceDirect . Project aims to enhance object detection capabilities by improving processing speed and incorporating advanced analysis for defect and mismatch identification in video footage.

[8]. Ange Joel NoungaNjanda, Jocelyn EdinioZackoGbadoubissa. December 2024, People counting using IR-UWB radar sensors and machine learning techniques, ScienceDirect . Proposes a high-accuracy multi-object tracking model using enhanced YOLOv2 and LuNet, with superior performance in vehicle detection and efficient FPGA implementation.

[9]. Longchao Shen,Yafeng Liu.14 September 2024. Development and challenges of object detection: A survey. ScienceDirect . The evolution of object detection technologies from traditional methods to advanced deep learning approaches, emphasizing YOLO algorithms and their applications, datasets, evaluation metrics, and future challenges.

[10]. Abrar Elaoua; Mohamed Nadour, 28-29 November 2023, Real-Time People Counting System using YOLOv2 Object Detection , IEEE , Real-time people counting system using YOLOv2, showcasing its high accuracy and robustness in tracking and counting individuals across diverse scenarios for applications in crowd management and surveillance.