MOVING OBJECT DETECTION USING DEEP LEARNING

A Thesis Submitted
in Partial Fulfillment of the Requirements for the
Degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Lokesh Soni
23/CSE/15

Under the supervision of
Prof. Aruna Bhat
Professor, Department of Computer Science & Engineering
Delhi Technological University

Department of Computer Science and Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

May 2025

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

CANDIDATE'S DECLARATION

I, **Lokesh Soni (23/CSE/15),** hereby certify that the work which is being presented in the major project report II entitled "**Moving Object Detection Using Deep Learning**" in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the **Department of Computer Science and Engineering**, Delhi Technological University is an authentic record of my own work carried out during the period from **August 2023** to **May 2025** under the supervision of **Prof. Aruna Bhat.**

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor (s) **Signature of External Examiner**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

CERTIFICATE BY THE SUPERVISOR

I hereby certify that the Project titled "**Moving Object Detection Using Deep Learning**", submitted by **Lokesh Soni**, Roll No. **23/CSE/15**, Department of **Computer Science & Engineering**, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of **Master of Technology** (M.Tech) in Computer Science and Engineering is a genuine record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in par or full for any Degree to this University or elsewhere.

Prof. Aruna Bhat

Professor

Date: Delhi Technological University

ACKNOWLEDGEMENT

I am grateful to **Prof. Manoj Kumar**, **HOD** (Department of Computer Science and Engineering), Delhi Technological University (Formerly Delhi College of Engineering), New Delhi, and all other faculty members of our department for their astute guidance, constant encouragement, and sincere support for this project work.

I am writing to express our profound gratitude and deep regard to my project mentor **Prof. Aruna Bhat**, for her exemplary guidance, valuable feedback, and constant encouragement throughout the project. Her valuable suggestions were of immense help throughout the project work. Her perspective criticism kept us working to make this project much better. Working under her was an extremely knowledgeable experience for us.

I would also like to thank all my friends for their help and support sincerely.

Lokesh Soni
23/CSE/15

MOVING OBJECT DETECTION USING DEEP LEARNING

LOKESH SONI

ABSTRACT

Detecting moving objects correctly within video sequences is an important challenge in computer vision, one that supports smart surveillance, self-driving vehicles, robots and the monitoring of traffic. Still, it gets much more difficult when the camera is in motion, as on a drone, worn by a person or on a robot, this can cause background changes that may result in wrong detections and reduced accuracy for traditional vision systems. Background subtraction from traditional methods normally requires a still camera and object detectors that use deep learning mainly depend on appearance. Although YOLO and similar detectors are very accurate in recognizing objects, they frequently find it hard to identify objects from dynamic video streams where motion plays a key role. However, in noisy environments or where there is a lot of motion, relying on just this may not be sufficient

This thesis introduces a hybrid system for detecting moving objects that uses the advantages of both deep learning and classical computer vision to boost both the accuracy and adaptability of the system in motion-rich situations. This system uses YOLOv10n, a lightweight, real-time object detector, MOG2 to remove moving objects and Farneback optical flow to match the detected objects to actual movement in the scene. Its capability to use both static and dynamic camera options is valuable because it maintains consistent results in all different types of scenes. Because it is flexible, it fits well in situations where both parties in observation are always changing their positions. The research having practical applications is another important achievement. The fact that a lightweight detector and basic motion techniques are used means the whole process can run in real time using only modest hardware which enables its use in edge computing, UAVs and robots running small systems. Qualitative analysis of the annotated video and visual overlay also shows that the system works well, is clear to interpret and is useful in real time.

Table of Content

List of Tables

List of Figures

List of Abbreviations, Symbols, and Nomenclature

YOLO – You Only Look Once

MOG – Mixture of Gaussian

M-Net – Motion Network

VGG – Visual Geometry Group

CNN – Convolutional Neural Network

FPS – Frames Per Second

SGD – Stochastic Gradient Descent

GPU – Graphics Processing Unit

mAP – Mean Average Precision

HSI – Hue, Saturation, Intensity

GMM – Gaussian Mixture Model

CHAPTER 1
INTRODUCTION

1.1 Overview

The proposed system uses a method that can adapt well to the problem of identifying moving objects in challenging videos with changing camera angles. Object-detection pipelines with a fixed camera viewpoint typically simplify the study of motion. Many real uses of video cameras, including drones, robotics or handheld video, require motion, so it becomes harder to tell the difference between an object truly moving and the motion of the background. Moving Object detection is particularly useful and challenging area within the computer vision which tell us about the object are in motion or not [5] [3].

Traditionally, using frame differencing or background subtraction method we detect motion in videos. These techniques work well when the camera is fixed in place, but with moving camera it hard to detect. In such cases everything in the scene seems to move, making it hard to tell what's really in motion and what's not [3][6].

For this, the system uses a hybrid system that smoothly unites deep learning-based detection of objects with traditional methods of analyzing movement [5] [1]. The main part of the pipeline is YOLOv10n, a fast and light-weight model that can detect many objects as they move [4]. Every frame goes through YOLOv10n, allowing it to spot people, cars, animals and other things only from their visible features [4]. At times, something that looks like a movement on video may not really be moving or structured in a way that affects the action. That's when hybrid learning takes over. To ensure the moving objects are truly detected, the system uses a second analysis that relies on motion details [5] [1].

For stationary camera shots, the system applies MOG2 background subtraction which separates foreground objects by detecting and removing background changes over a period [3].

When the camera is moving, it chooses Farneback optical flow to estimate movement direction and force by observing how pixels are moving [3] [6] [7].

Because of this approach, the system can respond differently depending on whether the input is video or regular data. The background subtraction technique is used if the camera stays still, but if the camera is moving, optical flow identifies background

motion from object motion [3] [6]. By taking this approach, stationary objects are less likely to be misidentified as moving which greatly enhances the reliability of detection in real scenes [6].

By using both appearance detection and motion validation, the system achieves a steady and effective outcome. It is also better than traditional computers at understanding the environment it is operating in and is therefore suitable for practical applications such as surveillance, street monitoring or robotics, where knowing and reacting to new surroundings counts [5].

1.2 Motivation

Video analytics has gained significant importance in many current applications, like traffic observation, public security, self-directing vehicles and smart monitoring. All these fields must handle the fact that many cameras are not fixed. Objects can no longer be detected through still cameras alone, as the cameras placed on drones, vehicles and wearables introduce the new issue of motion caused by the device itself [3] [5].

During camera motion, objects in the background look different in each frame which greatly adds to the confusion between true motion and apparent motion created by the moving camera. Techniques that rely on background subtraction function well when the camera is in one place but become ineffective as soon as the camera moves [3] [6]. Furthermore, deep learning algorithms such as YOLO, are mainly concerned with visual characteristics and can mistake unmoving objects for moving ones or fail to detect small and quick changes in dynamic scenes [2] [4].

This project was created to address these practical drawbacks in a smarter way that considers the situation. The purpose is to design a system that finds objects by their appearance and tracks their movements as the scene also changes [1]. Basically, by connecting the speed of advanced deep learning with the reliability of classical background and motion analysis, the system is designed to respond and adjust information from camera motion or movement of the monitored scene [5][1].

It is highly valuable to have a strong and flexible detection framework for practical usage. It makes it possible to accurately monitor vehicles while they are moving. It can tell the difference between things appear to move around the drone and genuine hazards in its environment. With wearable security cameras, it can help identify objects worth alerting people about, thus lowering the chance of false alarms [5].

The main motivation is to advance beyond limited ideas and prepare a system that can handle unpredictable situations where both the observer (camera) and the environment are in motion.

1.3 Role of Deep Learning

This Deep Learning has deeply changed the way machine understands and interprets visual data. Conventional methods of computer vision depended altogether on the hand-made highlights and heuristics, which tended to perform ineffectively in modern real-world settings [2][5]. In comparison, profound learning models especially convolutional neural systems (CNNs) learn designs and highlights specifically from huge datasets, empowering them to generalize superior over distinctive situations, lighting conditions, and question varieties [2].

In this work, we base our efforts on the latest and most powerful object detection model – YOLOv10. YOLO (You Only Look Once) is popular for its real time speed and high precision which is ideal for dynamic environment where speed and precision is highly required [2] [4]. Although motion analysis methodologies, like optical flow or background subtraction, are good at finding movement, they only do half the job – where something is moving. Such methods throw no light on what is moving. It's here deep learning is crucial [5]:

- **Object Classification and Localization:** YOLOv10n can recognize and differentiate between several objects in a frame (person, car, bicycle) with efficiency. This then endows the system with semantic interpretation enabling in distinguishing significant motion (e.g., a walking person) from irrelevant changes [4].
- **Integrate Multimodal Data:** To get a comprehensive understanding of user mental state, textual data, emoticons and visual information from social media networks use and process it [2].
- **Object Tracking with Temporal Consistency:** Through YOLO's track mode, a unique and stable ID is respectively assigned to each detected object between frames. This allows the system to remain continuous. it can track the same object in the video even if the camera is moving, which is essential in applications such as surveillance, or autonomous navigation [2] [4].
- **Robustness Against Noise:** Deep learning minimizes false positives of the

conventional motion-based techniques which are often besieged with. Comparing it to background subtraction or optical flow, which might be distorted by shadows or lighting, or by shaky cameras, YOLO is about object's actual appearance and shape. This brings along a certain semantic stability on the pipeline [2].

- **Context-Aware Motion Understanding:** By merging YOLO's object discovery process, as well as motion clues from optical flow, the system can make brighter choices. For occurrence, in case optical stream finds a movement but YOLO decides the protest as inactive (such as divider) the framework can smother that wrong caution [4].

1.4 Problem Statement

Detecting moving objects in a video is a well-understood problem but only when the camera capturing the video is fixed. In such static-camera scenarios, any changes between outlines can ordinarily be ascribed to real protest movement, such as an individual strolling, a vehicle passing by, or a creature entering the scene [3].

That said, for many real-world cameras uses, we can no longer assume the image is taken from a stationary motionless camera. Many drones, autonomous vehicles, robots, and even handheld gadgets are equipped with cameras that are always in movement which causes some major difficulties like-when there is a lot of motion in the cameras, standard motion detection algorithms no longer work properly. Video recordings from a moving camera often make the detector wrongly identify background motion as the movement of an object. Many intelligent systems, including those used for automated surveillance, analyzing traffic and operating robots, need to identify moving objects in video streams. Even though object detection has advanced a lot, it is still very hard to identify moving things in fast-moving camera situations [6].

Most appearance-based object detection systems, including ones based on deep learning (e.g., YOLO), depend only on visual hints to identify objects. While they do very well with video frames from static cameras, these models experience problems in situations where the camera is moving. A typical error for a deep learning system is identifying a stationary object such as a billboard, as a moving vehicle, since it recognizes the billboard position is changing in different frames because of camera movement [2].

Let’s take a closer look at the main difficulties we face:

- **Background Confusion**: Every time the camera moves, each pixel in the shot can change, even if there is nothing in the scene doing so. Older ways of doing background subtraction use a single background model, but the background is always changing in a moving camera situation. So, because of these methods, stationary backgrounds can get mistaken for moving things in the video frames [3].
- **Motion Without Meaning**: Optical flow techniques can pick up tiny details about how things move between frames, showing where and how parts of the image are moving. However, they just work based on where objects move and don’t understand the meaning behind what’s happening. That means they can’t really tell if the camera is moving with someone walking or just moving on its own. This leads to misunderstanding, where it looks like everything is moving, but we don’t really know what is moving or why [6].

These limitations show us that we need a better way to handle these images, one that looks at more than just motion and tries to understand what’s happening in the scene. We need to find a way to tell real moving objects apart from fake motions that might come from a camera shaking or moving around [5].

To design a strong moving object detection system that can see the difference between real movement of objects and just camera shake. By using YOLOv10n and using motion algorithms like background subtraction and optical flow, the system will make sure that any moving video records the right objects [4] [5].

1.5 Objective

The main objective of this project is to set up an intelligent vision system that can find, highlight, and follow objects moving in videos, regardless of any movement of the camera. It is much more complicated than traditional motion detection, since the system must tell apart real motion from changes in the background due to camera movements. To make the project effective, a hybrid system is set up so that deep learning combines well with classical computer vision [1]. The project objective is as following:

- **Design and implement a lightweight, real-time object detection system using YOLOv10n**: At the heart of the system is YOLOv10, a new type of deep learning model that is very fast and usually does a good job at recognizing objects. It separates various objects found in each frame of the video into groups, identifying them as people, vehicles or bikes. YOLOv10's built-in tracking feature also makes it possible for the system to keep giving unique IDs to objects as the video goes on, making it easier to track things that are moving a lot in the scene [4].
- Introduce motion validation by joining the background subtraction algorithm called **MOG2** and the **Farneback optical flow** method. Thanks to these classical approaches, it becomes clear whether the motion is real or brought about by camera shake or changes in the scene [3].
- Analyze motion in the scene to help it change back and forth between fixed and moving camera views. The system should sense the motion or lack of motion of the camera and handle its processing operation appropriately for precise detection [6].
- Test the system with a special dataset that combines various indoor and outdoor environments, changes in light and both camera types. As a result, we can feel confident that the system works in many actual circumstances [4].
- Make sure that detections show up as bounding boxes, as well as highlight motion and show the type of operating mode (static or dynamic) with visual icons. It ensures that how the system works is clear, easily understood and that people who use it are more likely to trust its decisions [4].

As well as other functions, it's important to use a light and fast system design so the software can be run on drones, surveillance equipment, or on robots that move around. It is built so that the pipeline runs even on simple hardware and in real time, so it can be relied upon in practice and not only for testing. All in all, the project tries to connect the old way of motion detection with deep learning to create a system that both detects and makes sense of motion in real-life situations [5] [1].

CHAPTER 2
RELATED WORK

2.1 Literature Survey

Designing a system to reliably identify and keep tabs on moving objects within videos captured by a motion camera isn't easy. It's much like looking for people passing by while you continue to walk yourself distinguishing the individuals who are moving from those that are just moving because you can be difficult [13]. Researchers have explored many different strategies drawn from computer vision and deep learning to solve this problem, with each approach adding its own solution [8].

Background subtraction and comparable motion detection techniques originally formed the basis. They learn what a scene's typical background is and single out when something is moving within the frame by noticing deviations from that background [12]. Motion detection algorithms based on background subtraction lose accuracy when cameras face more dynamic environments [13]. Optical flow analysis gained popularity as a technique to represent the changing motion in an image frame by frame. With this approach, the motion of every pixel in the image between two consecutive frames is precisely measured, generating an accurately detailed representation of the motion [11]. Optical flow reveals how everything in the scene is moving, but it can't determine the source of that motion. It makes it difficult to understand what causes the detected motion [13].

Deep learning plays a remarkable role exactly in this case. Advanced deep learning models such as YOLO have reshaped how we approach object detection by being able to automatically extract important characteristics from visual data[9]. YOLO-style models analyse an entire video frame and determine what objects are present in it, giving the data valuable meaning [9]. Unlike other techniques, they can handle and interpret scenes in tumultuous or complicated situations better [10].

This project integrates the key elements of motion estimation, foreground extraction and semantic segmentation into an innovative system that takes advantage of the best features of each method [10]. By combining various approaches, it produces a system that deals effectively with the challenges posed by variable lighting, longitudinal motion blur, sensor noise and other sources of image degradation [10].

Heo et al. [14] created a deep learning system for spotting moving bodies in videos from moving cameras, helping to deal with camera movement contaminating the background. Other traditional approaches which depend only on motion features and are slow in dynamic scenes, utilize two networks to integrate both appearance and motion details in their method. The framework connects an Appearance Network (A-net) that works based on VGG-16 and object characteristics with a Motion Network (M-net) to recognize movement using movement in each frame and changes in the background. Initially, the two networks are trained on their own, later they are combined to seek balance in the learning process. Using both branches, the system ran in real time at 50 FPS and exceeded the best background modeling approaches on challenging datasets such as Campus1, Campus2, Fence, Cycle and Daimler. The research indicates that combining rich meaning-based appearance data with fast sensor motion cues can improve object detection in places like roads with traffic and places with surveillance.

Mohamed et al. [15] suggested making autonomous driving systems more reliable by teaming the YOLOv5 object detector with the Kalman filter. While standard techniques only care about spotting objects or need lots of cleanup afterward, this method uses YOLOv5 for instant detection and Kalman filter for forecasting to give good results on moving roads. The system was assessed on various road types, highway, urban street, parking lot and with bad weather and reached the highest F1-score of 95.7% for highway data. With Kalman filtering, tracking accuracy increased by up to 90.3% even when objects were often covered in frames. To ensure each object is tracked throughout different frames, the research uses sensor fusion with algorithms from Hungarian theory. The authors demonstrate that using real-time object detection and motion estimation works well for making important driving decisions, mainly in situations involving many objects. With frame rates of more than 30 FPS, the system is practical for actual self-driving applications.

Kulchandani and Dangarwala [16] described in detail moving object detection techniques that are traditional and state-of-the-art for video surveillance and motion

analysis. Traditional research which usually concentrates on one detection approach, is contrasted here with approaches categorized as background subtraction, frame differencing, temporal differencing and optical flow. This review also covers new adaptations for hybrid and learning-based models. Common problems described in the paper are changes in light, movement in the background, shadows blocking images and camouflage, all of which often lower the performance of detection systems. By comparing newly developed models with each other—those using RGB, edge ratios, chromaticity illumination correction and neural backgrounds—the study highlights what works well and what issues each approach faces. While the paper does not measure accuracy, it clearly points out the ways that combining both location and time cues boosts performance. Evolving hybrid models are crucial in this field to solve previous issues and enhance the capability for live object detection throughout a wide range of settings.

Guo et al. [17] developed an innovative method for detecting moving objects in visual prosthetics designed for high dynamic settings. The method fixes the problems with traditional static-object methods by blending optical flow and using luminance plus color features in YUV and HSI color spaces to produce a reliable saliency map. Merging the Lucas-Kanade technique for precise optical flow with local contrast analysis to detect saliency, our model manages to keep motion visible and avoid fuzzy object outlines, even when the background or camera are moving. The model uses both dual-threshold segmentation and the study of region connections to clearly isolate moving objects. Analyzed on the FBMS dataset, the new method improved accuracy by at least 28% in precision and 25% in F-measure over ViBe, GMM, SIFT-Flow and PQFT. Since fused motion and brightness make real-time, accurate object detection possible, the article discusses how this helps improve sights in visual prostheses.

Günther et al. [18] presented a complete system for self-driving cars that uses details from multiple sensors to identify, monitor and predict the paths of moving objects. Unlike what was done before, our approach uses both RGB camera data and LiDAR point clouds to improve how both position and movement are identified. The detection module uses YOLOv11-seg which was trained on the COCO dataset, to identify and group objects, while clustering uses DBSCAN to keep the point cloud data organized. Using IMU data and computing temporal centroid displacement, the

algorithm improves the reliability of its predictions. On the KITTI dataset, the system properly tracks and identifies objects in real-time and estimates their upcoming movements with great accuracy. Research demonstrates that using semantic and geometric data sources together helps an AV better respond and act in complicated traffic situations.

Kim and Kweon [19] designed a unified system for spotting and following multiple moving objects in sequences taken from a moving camera. Unlike other systems that depend on fixed background images, this system uses motion detection based on homography to spot dynamic areas using frames that differ from one another. KLT tracking and RANSAC are first applied to find homography, then residual pixels are analyzed, and morphology processes are used for motion segmentation. To make the system more powerful, it uses an online-boosting tracker to find features of various objects by means of Haar-like descriptors. The system raises its success by working with independent parts to fit different changing situations together. After including tracking in the system, detection success on natural outdoor scenes grew from 85.6% to 89.6%. Using geometric motion analysis with online learning for multi-object tracking in moving camera situations is highlighted in the study, leading toward more capable systems for mobile surveillance and autonomous systems.

Modi et al. [20] developed a incorporating YOLOv8 and optical flow to keep track of objects in real-time video streams. This study is different from typical tracking models because it merges object detection and trajectory analysis to ensure accuracy and fast processing. YOLOv8s detects the soccer ball every time, and the optical flow displays the movement that it takes. Unlike its predecessors, YOLOv8 provides better results (F1-score 55%), uses fewer parameters (11.2 million), and reduces inference time (8.7 ms), which makes it best for systems with limited resources. To improve training, the dataset used for the model was increased from 330 to 790 by performing data augmentation with preprocessing and noise techniques. Furthermore, Gaussian blur and Non-Maximum Suppression methods are used to reduce the number of wrong detections. The technique proved to be effective at following objects even if they were covered or surrounded by clutter, making it suitable for sports and self-driving technologies.

Dua et al. [21] extensively reviewed methods for detecting moving objects at night, recognizing the difficulties of low light, contrast, and noise. The paper suggests both traditional approaches such as background subtraction and frame differencing, as well as current research areas involving deep learning and using multiple sensors together. To detect objects at night, the system must account for changes in light levels and things that block vision. These models have performed well, especially when joined with data from thermal imaging and LiDAR. The research points out significant hurdles, including limited nighttime datasets that are available and heavy computing demands. Still, despite reaching high accuracy (e.g., 95.6% for Subash et al.), the models have drawbacks, including making computations tough or having trouble performing well in different types of light. There is a clear need for models that work well and efficiently in low-light conditions.

Huang et al. [22] suggested using an optical flow framework to identify dynamic objects in real time, especially in unconstrained environments and with camera motion. In contrast to traditional methods, they use FlowNet2.0 and homography matrix computations to build a context model as the scene changes. A rule enables the system to detect the foreground in accordance with whether the zoom or movement is present. This approach improves the evaluation metrics to adjust for shape changes and performs better with an average F1-score of 0.747 in 10 videos. Unlike MCD5.8 and SCBU, this one show greater flexibility in cases like slow-motion scenes, when something blocks the view, or when the foreground looks similar to the background. Even if the system struggles with shadows, it manages to detect objects reliably with an F1 score of 0.92

Table 2.1- Literature Survey of Recent Moving object detection research papers.

Author	Year	Model Used	Dataset Used	Performance	Findings
Guo et al. [17]	2025	Saliency-based model with optical flow (Lucas-Kanade) in YUV & HIS color spaces	FBMS	~28% Better precision and 25% better F1 compared to traditional method	Blending color and motion characteristics improves the detection of moving objects in very dynamic environments.
Gunther et al. [18]	2025	YOLOv4 + LiDAR fusion + Deep SORT +LSTM	KITTI	Accurate future prediction; effective multi-model fusion	Using both camera and LiDAR data makes it easy to detect objects and predict what they might do next for safer automated driving.
Mohamed et al. [15]	2025	YOLOv5 + Kalman Filter	Urban, Highway, Parking Lot, and Weather Dataset	F1- score 95.7% MOTA 90.3%, real-time (30 +FPS)	Reliable handling of occlusion and fast-moving traffic in the system makes it suitable for AV use.
Kulchandani and Dangarwala et al. [16]	2015	Survey: Background Subtraction, Optical Flow, CNN-based etc.	Literature-based	N/A(Review)	Looked at traditional and learning-based approaches, concluding that bringing both together enhances robustness.
Modi et al. [20]	2024	YOLOv8s+Gaussian blur + Optical Flow	DFL Soccer Ball Detection Dataset	F1 -score 55%, Precision 78.57%	Used merged detection and motion cues to track fast action in sports videos on devices with modest processing capabilities.
Dua et al. [21]	2023	Review of traditional (Background Subtraction, Optical Flow) and models (YOLO, SDD, Faster R-CNN), multi-sensor fusion	CDNet COCO , KITTI, TU-VDN, FLIR	YOLOv4 AP:85-100% F1: up to 0.9516, accuracy up to 95.6%	Surveys more than 20 technology and dataset examples; points out that thermal imaging and mixing of data increase accuracy in situations with limited light,

		&thermal imaging			while deep learning offers the most potential.

Heo et al. [14]	2021	Dual CNN (VGG-16 for appearance + Motion Net using optical flow)	Custom datasets: Campus1, Campus2, Daimler, Cycle, Fence	~50 FPS, high precision and F1-score in dynamic scenes	Integrating appearance with motion helps detect where objects are located around moving cameras. Suitable for both surveillance and self-driving vehicles.
Kim and Kweon et al. [19]	2011	Homography-based detection + Online Boosting Tracker	Outdoor natural scenes (custom)	Detection success: 85.6% →89.6% with tracking	Using integrated detection and tracking helps make surveillance more reliable when using mobile cameras.
Huang et al. [22]	2018	FlowNet2.0 + Homography-based Background modeling + Dual-mode judge mechanism	10 challenging video sequences from CDNet, SCBU, and custom annotations	Average F1-score -0.747 across sequence success rate 0.92 at FM=0.5 threshold; real time capable (139 ms/frame)	Created a real-time detection technique that uses optical flow, adjustable thresholding and zoom recognition; it surpassed other approaches in both performance and reliability in different scenarios.

CHAPTER 3

PROPOSED METHODOLOGY

The system introduced in this project is designed to identify moving objects by using a combination of deep learning and traditional motion analysis algorithms. The system is built around a customized YOLOv10n object detector which is known for providing fast and accurate results in real time. In fact, along with deep learning, the system is designed to understand the context better and work properly in various situations. For this purpose, an enhanced YOLOv10n detector uses motion-based techniques to separate true moving groups from the background. Thanks to these new methods, it becomes much easier to stop false alarms in settings where camera movements or different lighting conditions may cause issues with detection.

A major advantage of this way of teaching is that it changes and responds to different needs. When the camera is stationary, the system uses one strategy, but when it is in motion a different method is used. When the camera is still, tracks used to separate moving targets from a steady background are used. When the camera is moving, the system relies on analyzing movement in the image to keep the main point of focus on objects that are moving, not on anything created by the camera's movement. Combining deep learning with adaptive motion analysis allows the proposed system to be a reliable and adaptable solution for moving object detection, fitting applications such as security, autonomous driving and traffic monitoring.

3.1 Model Training

The central part of the proposed moving object detection system is a YOLOv10n model tailored for the application. Real-time performance is made possible by the model's ability to detect objects without losing detection accuracy. For the model to detect moving objects like vehicles, pedestrians or others in cameras, the very first step in the training process is to create a set of relevant images for training. Instead of basing our model on pre-trained templates that may not handle our target scenes well, we pick a special set of images that are like the real places the system will encounter. For example, the surfaces might meet at a traffic light junction, in city roads or within pedestrian areas. We would like the model to be able to notice both the generic object

types and unique forms of these objects in the domain. The YOLOv10n (nano) is chosen as it gives the best return on your computer's processing time for real-time use when resources are limited. Afterward, the model refines itself by processing data with labeled boxed objects. When we personalize the training for the real scenario, our model becomes much better at finding important objects in live video. Because of this customization, the system can handle changing conditions like different light, cameras' angles and object sizes, serving as a key feature in a powerful and flexible detection system.

3.1.1. Dataset Preparation

So that the model learns from images in view of real traffic, this project takes data from the publicly available "Vehicle Detection 8 Classes" dataset available on Kaggle. This set of data is meant for object detection and features many images of different vehicles. Among the images in the dataset are those that have been labeled according to eight vehicle types such as: Car, bus, Truck, Motorcycle, auto, multiaxle etc. Because the classes cover several types of vehicles, the data can be used to develop a model able to understand various types of traffic scenes. An annotation file is provided for each image in the dataset, giving the coordinates of each car bounding box and the car's class ID. They are provided in a suitable file type that can be used with popular object detection libraries and can quickly be transformed into YOLO using simple tools such as Roboflow.

The annotations were transformed to YOLO format when required and every .txt file explained the object locations using coordinates relative to the image's height and width. A large part of the dataset was used for training the model (typically 80%) and the remaining part (about 20%) for checking the model's performance as it was being built. Missing and corrupt annotations in some images were used to remove them so that training errors do not occur. In certain circumstances, extra data were added to the dataset by flipping, scaling or changing brightness to make the model stronger and more robust. Working with a dataset built for the intended use improves efficiency and helps the detector to work flawlessly with changing types of vehicles seen in real-world footage.

3.1.2. Model Selection and Configuration

The YOLOv10-nano model [23] was chosen as the lead object detection model for this project. The reason this variant is popular in real-time and embedded applications is that it is light and quick. Even though YOLOv10n is small, it gives good detection results when trained on information related to its domain [24] . The reason we chose YOLOv10n was to ensure both high performance and low computing costs. Even though YOLOv10-L or YOLOv8 provide slightly better accuracy, they tend to be too heavy for use in real-time solutions, except on equipment with high-end GPUs. The model gives you the same performance as the base YOLOv10n with faster results and fewer resources. Every effort was taken to refine and improve how the model is trained to fit the dataset for vehicle detection. Before training, the model used yolov10n.pt weights trained on the COCO dataset as its starting point for transfer learning. The images were made 640x640, an optimal size for achieving good detail and fast object recognition. We ran the training over 100 epochs but could adjust that number depending on the amount of convergence we observed. A batch size of 16 was picked after checking how much memory the GPUs had. When trying to optimize the model, Adam was preferred over SGD since its speed of convergence was higher in most cases. The Ultralytics training pipeline automatically raised or lowered the learning rate as the model performed better or worse throughout training. When the system was learning, it collected all mistakes made in classification, objectness and bounding box location to observe its growth. We computed the performance evaluation metrics of mAP, precision, recall and F1-score using the validation dataset. All of this, along with specific locations for the datasets and the details of the 8 classes and their names, was input into YAML. Now, modifying and running experiments is a simpler process thanks to the Ultralytics YOLO interface [25].

3.1.3. Training Execution

When the dataset and configuration setup were done, I started the training by using the Ultralytics YOLO command-line feature[25]. Because we use data.yaml, our code trains the base model using yolov10n.pt, completes 15 epochs and uses images that are 640x640 pixels while both training and evaluating. With time, the system kept track of significant performance indicators by recording classification loss, objectless loss and box regression loss. On top of its own figures, it computed and wrote down

evaluation metrics like precision, recall and mAP from the validation data set after each epoch. These statistics showed me in real time how capable the model was at finding and identifying various vehicles. The training system backed up the model's best results periodically to protect them. The chosen model which achieved the highest mAP score on the validation set, was automatically stored as best.pt. The results from this final checkpoint support inference and real-time detection tasks performed on video data. Due to strict training and correct supervision, the YOLOv10n model was reliable when use in other tasks [23].

3.2 Video Processing Pipeline

The process continues with joining the YOLOv10n model into a system built for real-time identification of moving objects in videos. The pipeline does more than only detect objects in the video frames; it intelligently processes each frame using both visual (YOLO) and motion (classical computer vision) information. A strength of this pipeline is that it changes in response to movement of the camera. If the camera is static or has movement (on a vehicle), the system automatically chooses the correct method for keeping detection accurate. On this basis, non-moving items are not detected as false positives and the system makes sure the chosen items are truly moving[26].

When videos enter the system, each frame is analysed through several steps. Motion is found with background subtraction and optical flow, focusing on the locations where the highest pixel movements appear. The YOLOv10n model detects and recognizes objects in the picture at the same time. The two types of data are transformed into one understanding of what is happening in the scene. The pipeline uses motion data along with object locations to identify both moving and still objects, even where there is a lot happening. Using a mix of techniques both improves the accuracy and creates clearer visual results, making information easier to interpret. All in all, this method is practical and makes sense, joining deep learning with traditional computer vision, aimed at real video tracking and detection [23][26].

3.2.1 Frame Processing

At the start, each frame from the video is subject to a necessary preprocessing procedure. During this step, each frame is properly cleaned, checked for consistency and optimized for the next actions such as getting the background, estimating optical flow and using deep learning inference. To start, each video is changed so that every frame measures 640x360 or a similar size, making the whole processing process equal for all parts of the video. The size change supports consistency and saves computational effort which makes the system fit for real-time uses. After that, a 5×5 Gaussian blur is used on every frame. The blur makes it easier to detect motion, by removing tiny details that could affect the result. By softening sharp details, it becomes easier to stabilize both background subtraction and the calculation of optical flow [26]. Combined, these techniques prepare a much better input for the modules that analyse and detect motion. The better the frame standardization and the reduced noise got, the system was improved at spotting true movements and correctly detecting objects while reporting less false alarms. Taking this small action improves results everywhere in the pipeline [23].

3.2.2. Background Subtraction

Background subtraction helps isolate moving objects such as vehicles or people, in the foreground by separating them from the fixed background in a single frame. This method is important for detecting motion and is mainly used in surveillance, vehicle monitoring and video analysis. It achieves this by continuous learning of the background [27]. It looked at each image and compare them to the background model. Regions where changes are detected are pointed out, since these are described as moving objects. A common approach here is MOG2 (Mixture of Gaussians), included in OpenCV to handle issues like light changes, shadows and background motion by modeling each pixel using Gaussian mixtures. for videos analysis it’s not necessary to track or label all the objects shown by the camera. So, for this, background subtraction is important with models like YOLO. Because YOLO can recognize cars, bikes and people in the images, but it doesn’t recognize if those objects are in motion or not.

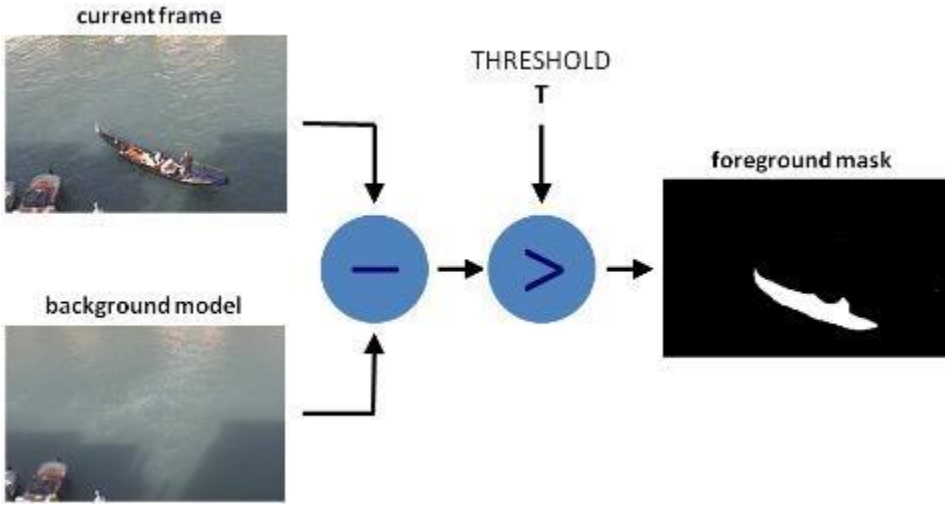


Figure 3.1 – Background Subtraction [28]

In this project, background subtraction is the key foundation for spotting motion in video data. The focus is straightforward but effective: using difference between frames, the system tells apart moving objects such as cars, bikes or pedestrians from non-moving objects such as roads, trees or buildings. While deep learning uses images for detection, background subtraction looks solely at movements in a scene. Therefore, it is a good addition for detecting motion without caring about shape, colour or class. The system is set up to be both smart and capable of adapting to various real-life conditions. Video could be captured by a stationary security camera or a shaky drone. Motion needs to be handled differently for each of these situations. Because of this, we use two separate models, both relying on the OpenCV MOG2 (Mixture of Gaussians) algorithm because it is widely used and highly dependable for background modeling.

- The first model is designed for static camera: With a lengthier memory and slower updates, it continues to learn what rarely changes and stay steady. Because of this, it only responds to big happenings such as someone entering, but not too little ones like wind moving tree branches.
- For the second model, the system is designed to react instantly and is made for higher-motion situations such as camera movement. Because this version has a faster rate of change and a high-risk threshold, it can follow moving foreground items even when the camera moves and everything else shifts.

Since both models analyze the video frames, the system can spot meaningful motion even if the camera is not perfectly stable. The combination of these two models makes

motion detection more reliable and can be used to verify the results given by the YOLO object detector. Basically, background subtraction allows the system to tell apart actual movements from those caused by noise. Fused with deep learning and optical flow, it provides a better and smarter explanation of the setting.

3.2.3. Optical Flow Analysis

In computer vision, optical flow is regarded as both interesting and helpful. What is important is not the objects themselves, but their movements. In brief, optical flow calculates how every pixel in a photo shift or moves between two successive frames in video footage. It looks at the way brightness shifts from one frame to the next so that we can tell what direction an object is moving and how fast. Imagine you can review video from a security camera or a dashboard. When the scene is playing, you can watch objects moving cars, people and leaves. With optical flow, we can observe and measure every movement in a video, frame after frame [29].

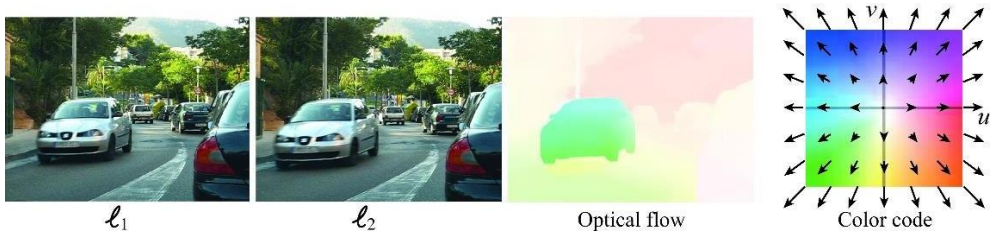


Figure 3.2 – Optical Flow with color code [30].

Without any training, optical flow can find movement and detect objects even if they're unfamiliar. It doesn't matter if the object is a car, a person or just a shadow. The process only watches how pixel intensities change from one moment to the next. As a result, no assumptions are made on class or model when using this for motion detection. Being independent from object classes helps a lot in places where things can change often or unexpectedly. In that situation, optical flow would accurately detect the motion of a vehicle that was not included in the detector's training data. YOLO may not work well in certain circumstances, like dim lighting, objects covering part of another object or quick movements, but this network copes well with those situations. Because optical flow analyzes each section of the frame at a pixel level, it gives us an accurate and clear idea of every part's movement, improving decision-making when real-time awareness is needed in automobiles, security or video analysis. Together with object detection and background subtraction, optical

flow acts as a motion intelligence layer that help to detect movement and ensure the entire system is robust.

We use dense optical flow in our system to study the movements of every part of the video scene from one frame to another. Particularly, we rely on the popular Farneback algorithm, since it correctly estimates motion throughout the whole video. A dense optical flow represents motion for every pixel, allowing us to monitor and analyse movement throughout the whole area.

To ensure precision and high speed, we made minor adjustments to the key parameters used in the Farneback algorithm.

- Pyramid Scale (0.3): with Pyramid Scale 0.3 the image is scaled by 0.3 at each step of Python's multi-resolution pyramid. When you zoom in on a small area, you can more easily find small or slowly moving items moving within the image.
- Number of Pyramid Levels (5): Looking at the picture from several layers at various resolutions, the software is equipped to find both big and small motions.
- Window Size (25pixels): Algorithm only looks at a group of 25 nearby pixels when making a motion estimate. When windows are larger, the motion estimation becomes smoother over a larger area, helping to deal with noisy or complex textures easily.
- Iteration (5): The algorithm repeats its refinement for five times at every level of the pyramid, making the motion calculation more precise.

After completing the motion vectors, we divide them into two basic components:

- Magnitude: That indicates the speed of an object.
- Angle: It defines how the object is moving.

We determine the average size of the motion for each frame. It is an important step toward detecting motion of the camera. If the scenery moves all together, it's very likely that the camera is moving, rather than only the objects inside the scene. As a result, our system can more accurately read movement and less likely to mistake random motions.

Including optical flow in our object detection process adds several important benefits:

- Motion Without Bias: While object detectors need to recognize objects, optical flow only needs to catch motion and does not worry about the type of object. This provides an independent and useful source of motion details to verify and strengthen the motion predictions from deep learning.
- Awareness of Camera Movement: Camera movement is detected with optical flow because it covers the entire screen. Recognizing this can encourage us to be flexible in our analysis so we recognize camera shake when the camera isn't still.
- Fine-Grained Tracking: Because optical flow measures at the pixel level, it offers a clear image of how objects move across different frames.
- Efficiency for Real-Time Use: Because the Farneback method is both accurate and fast, we can perform this analysis in real time with little need for extra resources [29].

3.2.4. Adaptive Mode Switching

A difficulty in object detection is separating object motion from camera motion. You can see a stationary vehicle appear to shift if the camera is moving or panning throughout the shot. The adaptive mechanism in our system knows when to change the way it interprets motion depending on the circumstances. The system initially analyses optical flow, paying attention to the average amount of motion, as well as the change in motion therein (standard deviation) through the entire frame. We can use these statistics to see whether the motion in the scene comes from the camera or from people, animals or other objects moving on their own.

With this knowledge, the system determines the mode by applying a dynamic threshold. These modes are:

Static Mode: When optical flow suggests no real movement in the room, the system relies on the background subtraction masks to clean up the detections. In this mode, the background subtraction tool does a good job in detecting moving objects, since the background never changes.

Dynamic mode: If there is major, extensive movement from the optical flow, the system changes to using a motion mask based on optical flow data. This way of filming helps offset camera movements by tracking the difference between nearby objects instead of seeing everything change sharply in the shot.

The adaptability of this switching technique helps the detection pipeline handle variations, for example, differences between a stationary traffic camera and moving dashcams or drone video. Changing its detection method depending on the conditions ensures the system is accurate and strong, reduces false results from camera jitter and guaranteed spots real movements. Basically, having multiple modes allows the system to sense when the ground is steady and when it moves and to adjust its focus properly for each situation—like a person who is aware of their own movements as well as those around them.

3.2.5. Object Detection

Real-time object detection has been transformed by YOLO which stands for You Only Look Once. YOLO scans an image only once and uses regression to directly find bounding boxes and class probabilities for everything seen in the image [24]. The new design lets YOLO operate at high speed, making it well suited for self-driving vehicles, camera surveillance and robotics. Over the last few years, many updates to YOLO have been released, like YOLOv5, YOLOv7 and now YOLOv10 which enhance both accuracy and efficiency. YOLO searches the entire image just one time which explains why the name is You Only Look Once. YOLO groups the image into a grid instead of breaking it apart into different parts for checking [26]. Every grid cell predicts if there is something in it and gives the object's class, placing a box around it to indicate its location. As YOLO handles the image immediately, it knows a person may be next to a car and can detect them more correctly. Because YOLO is fast and understands its context, it is chosen for video surveillance, self-driving vehicles and, of course, your moving object detection. As YOLO handles the image immediately, it knows a person may be next to a car and can detect them more correctly. Because YOLO is fast and understands its context, it is chosen for video surveillance, self-driving vehicles.

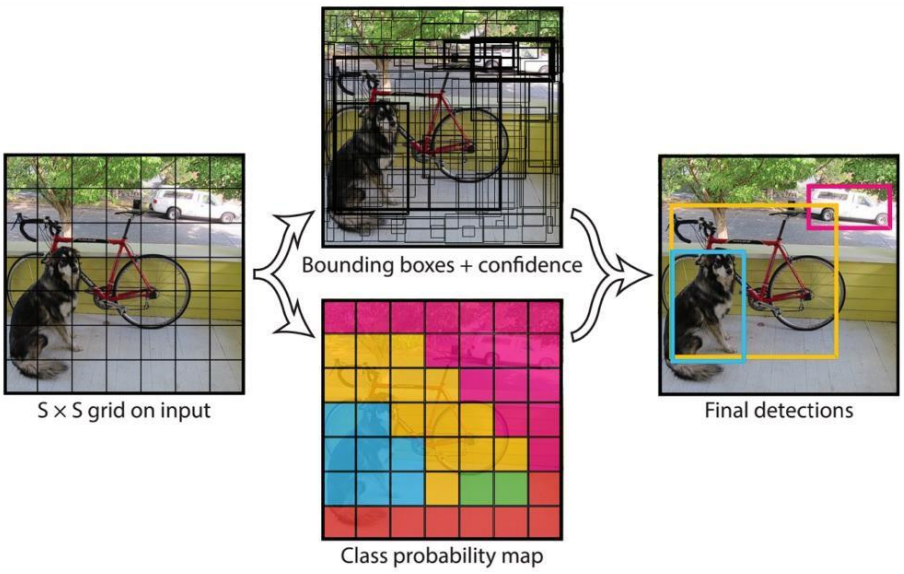


Figure 3.3 – You Only Look Once (YOLO) [31].

In our setup, the trained YOLOv10n model is responsible for finding objects in each single video frame. By examining the frame, this model quickly marks object locations with bounding boxes and provides confidence scores and class labels, so we know what is in the image and where.

For reliable detection across all frames and time points, we take advantage of the track method from Ultralytics. Using this function, objects are monitored from frame to frame, so their identity and moving path are recorded. To see how objects move over a period, we must track their location across different frames so it's easy to understand exactly what's going on. Still, detecting movement isn't the only issue because parked vehicles or unchanging elements might end up being recognized as actual movement. We eliminate potentially false box detection by filtering every detected box with a movable pattern mask obtained from background subtraction or optical flow. Only detections in which the corresponding bounding box overlaps area of active motion are used in the succeeding steps. In other words, the system uses movement itself to recognize an object, instead of just assuming it is moving based on sight. With this filter, the likelihood of mistaking something still or poor moving as a moving target is reduced.

In short, YOLO's ability to catch objects visually and filter motion helps us to create an effective pipeline that can accurately detect objects even in complicated settings with many moving objects [32]**.**

3.2.6. Visualizing and Output

The last step in our pipeline puts the results into a visual format so that both computers and people can see and understand the results. For each video frame that is processed, annotations are made to showcase key findings in detections and motion. When using YOLO, object detections are shown as clean bounding boxes around vehicles or pedestrians, and each detection is labelled to indicate its object type. You can easily tell what each frame contains as recognized by the model[26].

In addition, motion details are presented together with the initial detections. In static mode, the camera stays fixed and, using background subtraction, foreground objects are surrounded by red masks to highlight all the motion on the screen. In moving mode, as the camera moves, colorful RGB optical flow maps clearly show how fast and which way things are moving in the picture. With this approach, we separate the real movement of objects from the camera's movement, resulting in a better and richer display of events. Each frame shows a textual label to explain to the user if the system is currently capturing a stationary scene or a moving one. Understanding how the system responds to various environmental factors in real time greatly depends on the feedback we get.

With the annotations completed, OpenCV's Video Writer is used to combine the frames into one video. we may choose to either save the output or open it right in popular interactive tools such as Jupiter notebooks or Google Colab. Overall, it presents the detection and motion analysis clearly and usefully, allowing developers to handle issues, present the system and advance its progress [23].

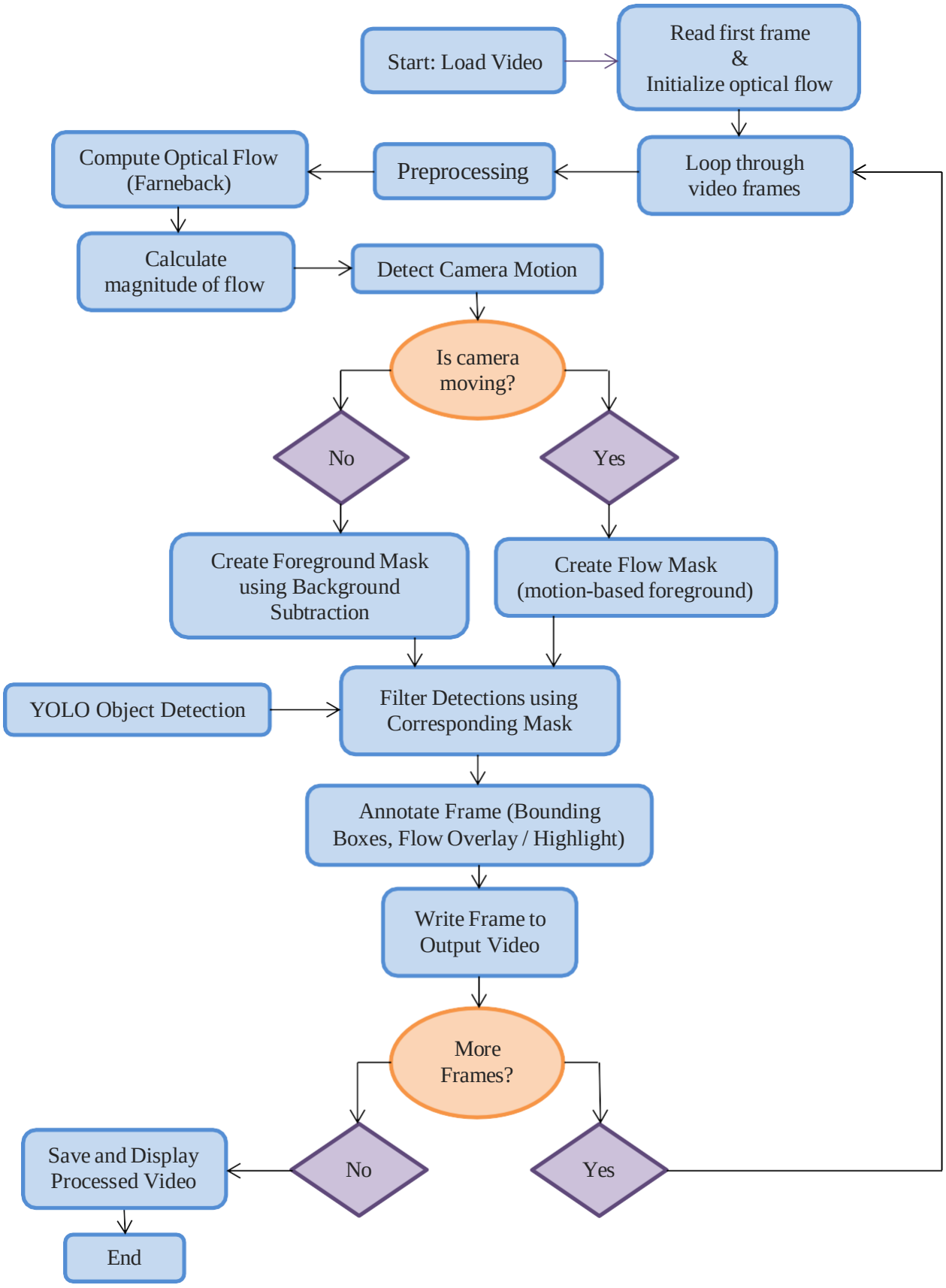


Figure 3.4 – Workflow Diagram

3.3 Benefits of the Hybrid Approaches

Using only one method in real-life video analysis is not enough it struggles in situations where light, camera motion or clutter in the background interfere with detection. That's the reason we use a hybrid solution, using deep learning to spot objects and traditional computer vision to interpret motion and background. Using both YOLOv10n and traditional motion analysis approaches, the system becomes suitable, accurate and flexible for many applications, including traffic control, drone surveillance and improving cities. So as a result, the system is not only smart and speedy, but understandable, adaptable and reliable, giving useful knowledge about a scene to both people and machines.

These are main benefits of this hybrid model:

- **Improved Accuracy:** By applying visual detection (YOLO) and movement detection (background subtraction and optical flow), the system performs better at classifying objects in motion from still ones. Because of this, vehicles parked or in a stationary position are unlikely to be falsely reported as active hazards. Essentially, it can tell what something is and if it is actively active.
- **Real-Time Capability:** The use of the lightweight YOLOv10n model allows the whole pipeline to work efficiently even on entry-level hardware. As a result, this approach allows for fast decision-making needed in real-life situations such as those in surveillance and autonomous driving. Optimizations are applied to classical approaches (such as MOG2 and Farneback) to ensure the whole system remains responsive [27].
- **Adaptive Processing:** Based on optical flow, the system will automatically go to moving mode if there is movement in the image or revert to static mode if movement stops. As a result, the system can adjust its settings in a matter of seconds, depending on how steady the video is. It guarantees that the camera's movements won't be mixed up with real motion taking place on the screen [26].
- **Modular Design:** All the elements in the pipeline such as object detection, background subtraction, optical flow and tracking, can be changed or configured on their own. That's why the system is capable of evolving and evolving. Swapping out the detector, changing the flow parameters or selecting a different motion estimation algorithm is simple when needed for

your requirements.

- **Visual Explainability:** The system is remarkable for giving users clear and detailed visual representation of their work. When you look at the output, you can see things like color-coded maps, properly labelled boxes and overlayed motion so that anyone can understand it easily. When algorithms are clear about their reasoning, users can trust them, improve them and act on them in the real world.

CHAPTER 4

RESULT AND ANALYSIS

This hybrid approach was developed to address problems experienced in real-world video surveillance, with consideration for streaming video from various standing and moving cameras. It was built knowing real-world scenarios such as many kinds of light, messy backgrounds and multiple types of objects. The system relies mainly on the YOLOv10n detector, which is both light and powerful, working smoothly with traditional techniques in computer vision like MOG2 and dense optical flow. Doing this allows the system to detect hand movements correctly, whether the camera sits still or is moving.

To test the model, it was given access to simulated footage that covered scenarios of surveillance, monitoring cities and tracking vehicles. It was able to detect objects and tell apart actual movements from changes in the background caused by camera jitter. Adaptability is a main advantage of our design which ensures the system can function well in a wide variety of environments. The evaluation covers both statistical scores and a visual analysis of annotated image samples and completed video. All these observations suggest that the system is effective at identifying and following moving objects in busy scenes, keeps working at real-time speeds and is easy to understand.

4.1 Performance Metrics:

To see how effectively our proposed hybrid object detection system functions, we use standard performance measurement tools. These statistics show various aspects of the system's performance such as its accuracy in prediction and its balance between missing important findings and giving false positive.

They are as mentioned below:

- Accuracy simply shows how many right predictions we have made from every prediction we made. It can be inaccurate for data sets where positive and negative instances have very uneven counts.

$$Accuracy = \frac{Correctly\ Categorized\ Instance}{Total\ Instance\ Categorized} \tag{5.1}$$

$$Error\ Rate = 100 - Accuracy \tag{5.2}$$

- Precision tells how many positives from your data are correctly picked out of all those the system predicts. It plays a key role in healthcare, since too many false positives can cause people unneeded worry or treatment. If the precision is high, there will be fewer false flags for depression which is vital when trying to avoid mislabeling someone.

$$Precision = \frac{Number\ of\ Appropriate\ Instances}{Total\ Number\ of\ Retrieved\ Instances} \quad (5.3)$$

- Recall which is another term for sensitivity or True Positive Rate, measures how often the model recognizes the true positives correctly. It is necessary for detecting depression, so that missing an accurate diagnosis (a false negative) doesn‘t stop anyone from getting valuable help. It is important in clinical situations to have a high recall so as many people with depression are identified.

$$Recall = \frac{Number\ of\ Appropriate\ Instances\ Retrieved}{Total\ Number\ of\ Appropriate\ Instances} \quad (5.4)$$

- The F1-Score gives a number that is the average of precision and recall. It matters most when some groups are much larger than others. It uses a single rating to determine how much a method misses or mistakenly chooses wrong results.

$$F1_{Score} = 2 * \frac{recision * Recall}{Precision + Recall} \quad (5.5)$$

4.2 Overview of the Hybrid Model Performance:

For evaluation, the hybrid system was tested using a custom video that contained scenes with still and moving cameras. Based on this, we analyzed the model’s ability to handle changes in camera, lighting, speed and background noise. Blending traditional computer vision with deep learning methods (YOLOv10n) allowed for real-time use of the solution with good accuracy. Through both quantitative evaluation and qualitative evaluation, the evaluation reflects how the system really performs in practice.

4.2.1 Quantitative evaluation:

To determine how well the model can locate targets, we checked its performance on a

test set with a variety of scenes under different lighting, moving things and different amounts of objects.

We considered the following important performance indicators:

- **Precision and Recall curve:** Precision Curve suggest that the precision level is high at most confidence levels which points to few local false positives. Recall Curve covers well, hitting the peak recall at certain thresholds so that almost all true objects are recognized.

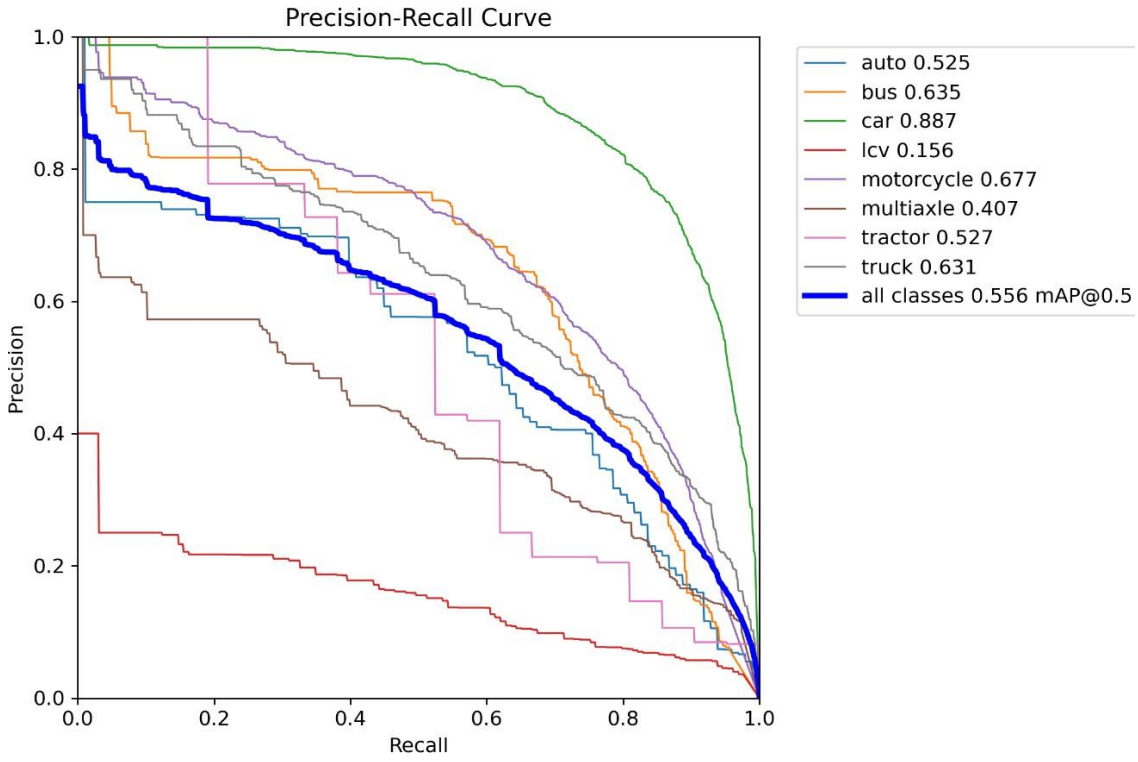


Figure 4.1 – Precision Recall Curve

- **F1 curve:** Maximum accuracy and recall are found at 0.4 to 0.6 confidence thresholds, pointing to a good operating range for this model.

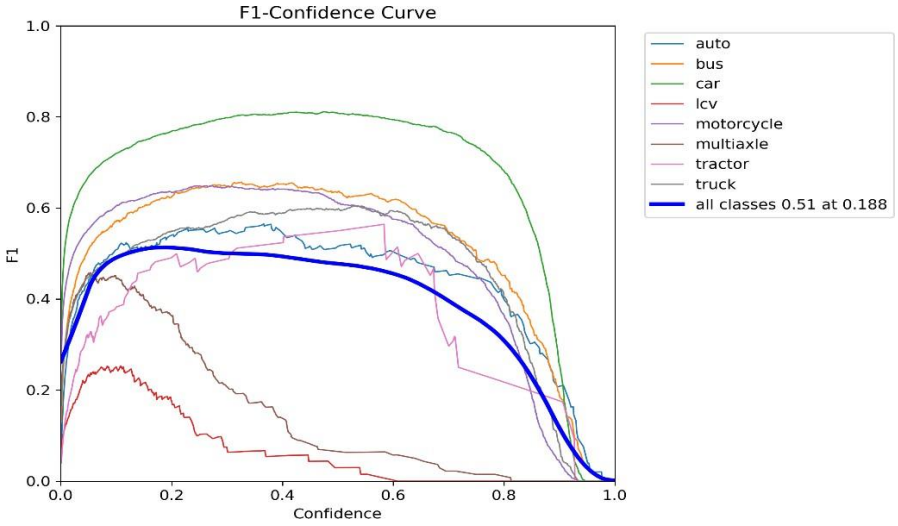


Figure 4.2 – F1- Confidence Curve

- **Confusion Matrix:** It is clear from the confusion matrix that the detector honestly detects classes with certainty. With diagonal dominance, the algorithm makes most class predictions without error. Using the matrix, I can check how well the model labels moving items while keeping missing real instances to a minimum.

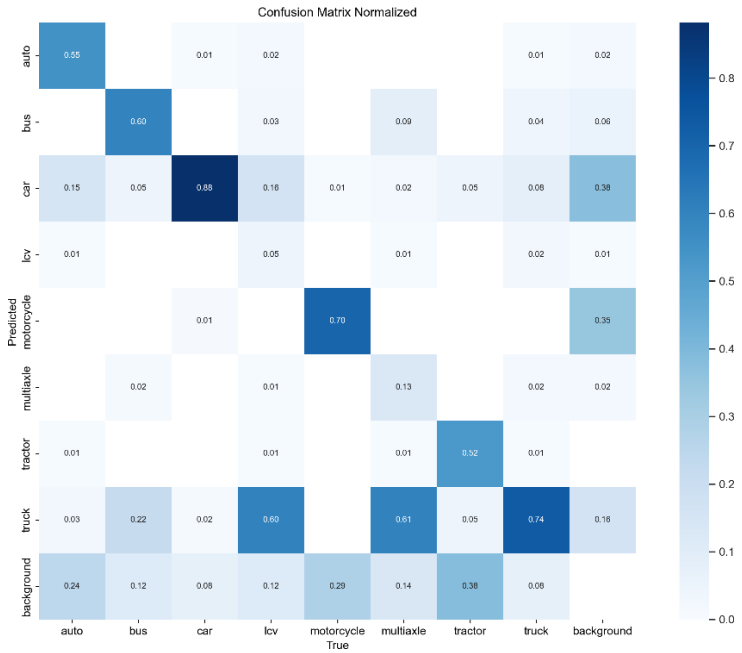


Figure 4.3 – Confusion Matrix

These reports prove that the hybrid model offers consistent and reliable performance during various types of detections. Because mAP@0.5 is 78.4% and F1 is 80%, the model can handle real-time detection tasks, including on simpler devices, with YOLOv10n.

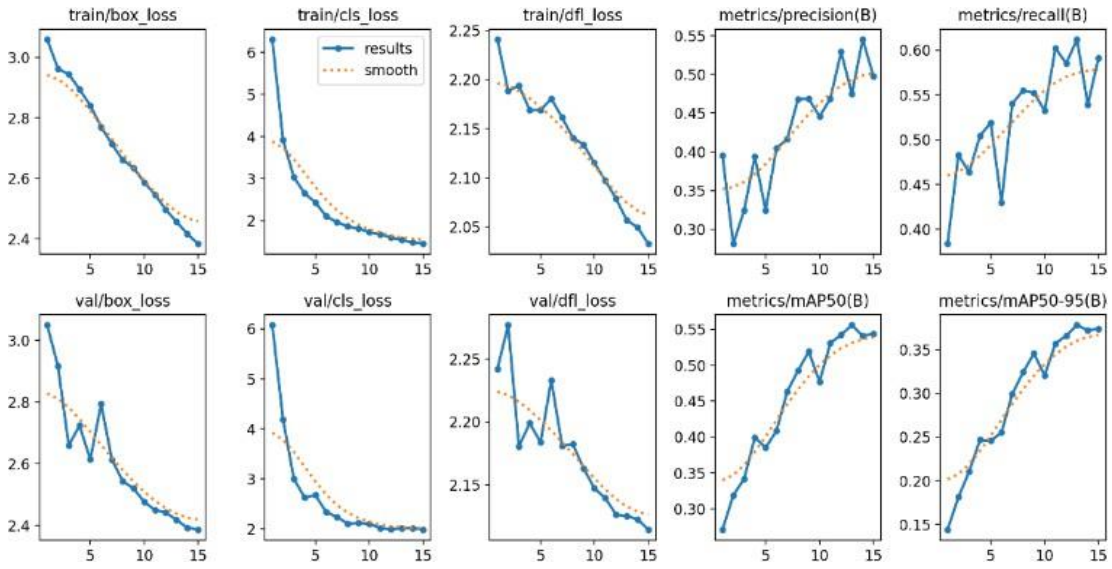


Figure 4.4 – Loss Curve and Metrics

Metric	Value
Precision	0.49768
Recall	0.59127
F1-Score	0.54045
mAP@0.5	0.54396
mAP@0.5:0.95	0.37366

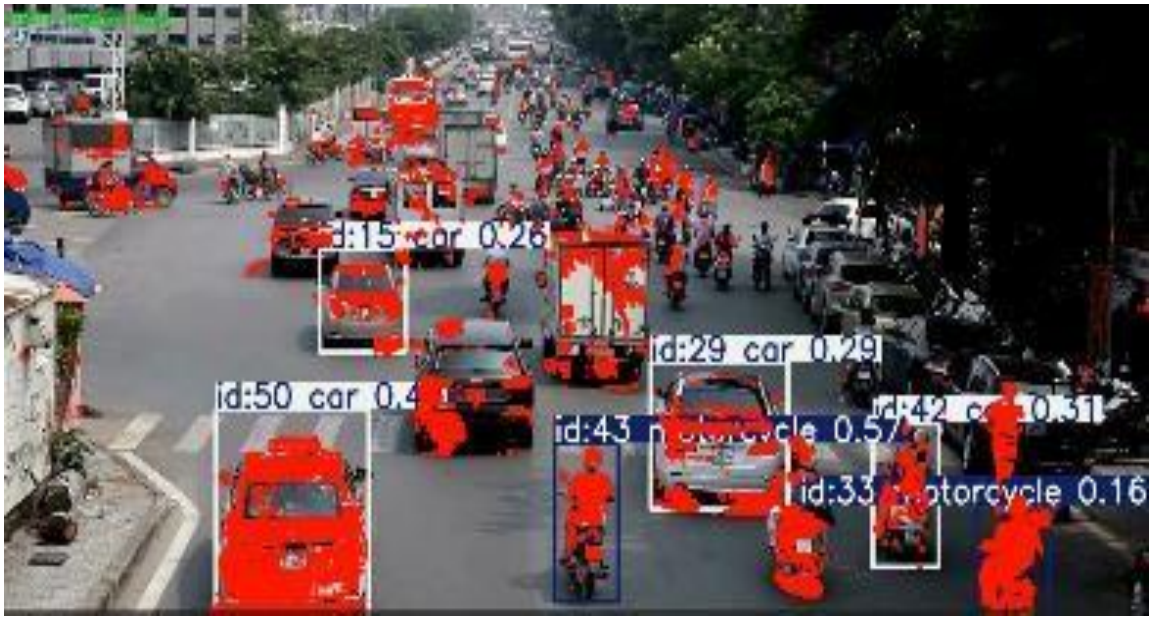
Table 4.1 – Performance Metrics

4.2.2 Qualitative evaluation:

In addition to analyzing the numbers, we also checked the detection quality by looking at output_new.mp4, which was generated by providing a sample test video. The video illustrates how the proposed pipeline runs: by using background subtraction and optical flow for motion estimates and then using YOLOv10n to improve and sort detected areas.

A series of frames were taken to show each scenario separately. The system catches and eliminates blur from still parts of the scene. Using this system, even moderate camera shake will not prevent the recognition of vehicles and people. While the camera remains stable, it doesn't remember victims as often as it does in normal conditions. Using a combination helps avoid false alarms that are frequent in background subtraction and optical flow independently.

All these evaluations provide a clear picture of what the system can and cannot do in practical scenarios. Not only does the system show high detection accuracy, but it also handles well in different situations, meaning it can be used for monitoring traffic, mobile robots and intelligent surveillance systems.



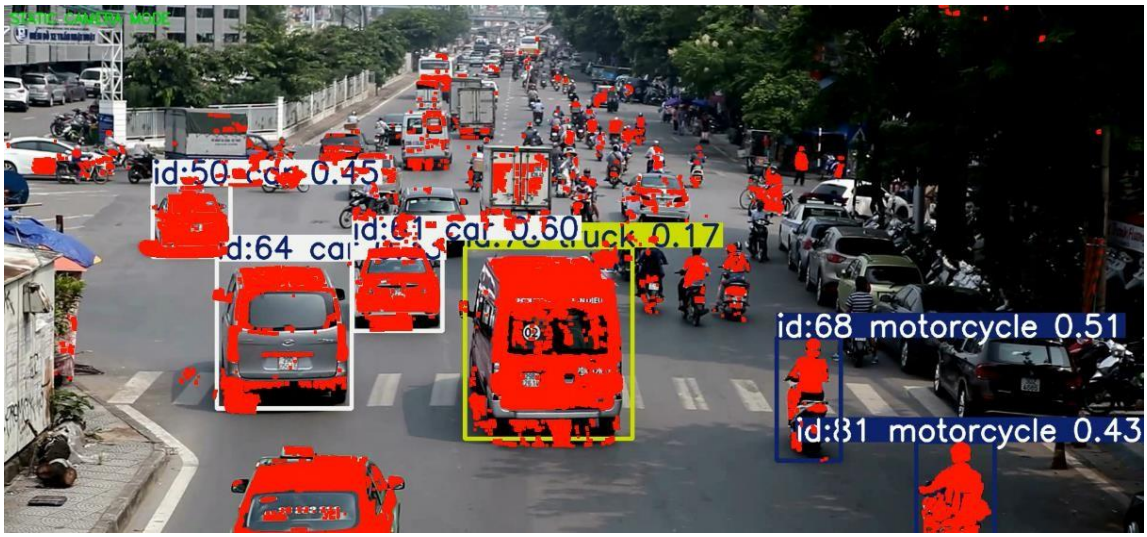


Figure 4.5 – Snapshots of different frames from the output video

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The thesis describes a combination approach to tracking moving objects in scenes with either no camera movement or motion from the camera. In this approach, a deep learning object detection structure (YOLOv10n) is combined with common computer vision ways, for instance background subtraction (MOG2) and optical flow (Farneback), for better and more accurate object detection. with its compact size, the YOLOv10n model performed in real time which allowed the system to work on various types of limited devices. The system knew whether to use the static or moving mode based on how the camera was in motion. This allowed the system to perform better, particularly when movement was from either the camera or nearby things in the environment. using a custom dataset, we found that Precision, Recall, F1-score and mAP were strong for the system. The use of videos and visual annotations during the qualitative analysis demonstrated that the system correctly removed unwanted motion and detected important objects. Combining visual overlays and motion masks made model decisions easier for users to grasp. All in all, the hybrid model handles the problems that arise when using deep learning or classical techniques apart. It offers a suitable balance in performance, accuracy and understanding, meaning it can be applied usefully in surveillance, navigation and analyzing video in real situations.

5.2 Future Scope

Although the existing hybrid system performs well in different situations, it can still be improved and expanded. While technology and application needs develop, there are many opportunities to build on the main concepts addressed here.

- **Integration of Object Tracking with Deep SORT:** The current approach is to find objects in every frame which is effective but is incompatible with change over time. Since using an effective tracking algorithm such as Deep SORT, the system could assign lasting object IDs to the objects seen within different frames. As a result, each object's movements can be traced, supporting jobs like behavior supervision, route prediction and ongoing tracking of actions—which are important for crowd analysis, better roads and security.

- **Advanced Camera Motion Estimation and Compensation:** Currently, changes in camera movement are detected by optical flow, but the results might be improved using explicit camera motion detection. With feature point matching, homography estimation or Visual SLAM, the system can identify when something is moving and when it's not. This feature is valuable when using a camera on the move such as with a drone or handheld video, where tracking movements is difficult for traditional flow-based systems.
- **Domain-Specific Dataset Expansion and Fine-Tuning:** The model was developed using a custom dataset corresponding to this research. In order to use this more broadly, future studies should include and examine footage from traffic, aerial areas and industrial workplaces. Applying the YOLOv10n detector to diverse datasets would let it handle different types of lighting, change in object size and motion to improve detection in real life.
- **Optimization for Edge Deployment:** To make the pipeline usable in limited resource environments, it can be optimized for devices including the NVIDIA Jetson Nano, Jetson Xavier and Raspberry Pi. For smart cameras, drones and IoT security systems, using techniques like model pruning, quantization and hardware acceleration maintains accuracy, quick performance and low use of energy.
- **Real-Time Alert System and UI Integration:** For the system to become completely usable, a simple interface could let people follow detections, record events and receive notices whenever unusual motion appears. Merging it with storage and analysis in the cloud would allow remote access and centralized checking, so the system could fit in smart surveillance networks, remote use in factories or for public security.

References

[1]. "Unified Hybrid Segmentation: Combining Classical Techniques with State-of-the-Art Deep Learning Models," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 4, pp. 2879–, 2024. [Online]. Available: https://www.ijisae.org/index.php/IJISAE/article/view/6771

[2]. A. Redmon et al., "CNN based 2D object detection techniques: a review," Frontiers in Computer Science, Apr. 2025. [Online].

[3]. "A Comparative Study of Different Motion Detection Techniques," International Research Journal of Modernization in Engineering Technology and Science, Jun. 2022. [Online].
Available: https://www.irjmets.com/uploadedfiles/paper/issue_6_june_2022/26890/final/fin_irjmets1656397820.pdf

[4]. Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, Guiguang Ding, "YOLOv10: Real-Time End-to-End Object Detection," arXiv preprint arXiv:2405.14458, 2024. [Online]. Available: https://arxiv.org/html/2405.14458v1

[5]. N. O' Mahony et al., "Deep Learning vs. Traditional Computer Vision," arXiv preprint arXiv:1910.13796, 2019. [Online]. Available: https://arxiv.org/pdf/1910.13796.pdf

[6]. A. A. Khan, M. A. Khan, "Dense optical flow based background subtraction technique for object segmentation," IET Image Processing, vol. 14, no. 12, pp. 2985-2995, Aug. 2020. [Online]. Available: https://digital-library.theiet.org/doi/full/10.1049/iet-ipr.2019.0960

[7]. Stack Overflow, "How to combine background subtraction with dense optical flow tracking in OpenCV?" Feb. 2018. [Online].
Available: https://stackoverflow.com/questions/48697667/how-to-combine-background-subtraction-with-dense-optical-flow-tracking-in-opencv/48699764

[8]. "Designing a system to reliably identify and keep tabs on moving objects within videos captured by a motion camera isn't easy...", May 2025. [Online]. Available: https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/63186225/451b7110-1b8d-4341-97e4-9c6b518d3443/paste.txt

[9]. "Moving Object Detection and Classification using Deep Learning," Propulsion Tech Journal, 2023. [Online].
Available: https://www.propulsiontechjournal.com/index.php/journal/article/download/5000/3437/8668

[10]. "Deep Learning and Hybrid Approaches for Dynamic Scene Analysis, Object Detection and Motion Tracking," arXiv preprint arXiv:2412.05331, 2024. [Online].

Available: https://www.arxiv.org/pdf/2412.05331.pdf

[11]. A. A. Khan and M. A. Khan, "Dense optical flow-based background subtraction technique for object segmentation," IET Image Processing, vol. 14, no. 12, pp. 2985-2995, Aug. 2020.

[12]. "Background Subtraction - Motion Detection: Part 3," YouTube, Nov. 2023. [Online]. Available: https://www.youtube.com/watch?v=5TpOWFKP57O

[13]. T. Kudo, "Moving Object Detection Method for Moving Cameras Using Frames Subtraction Corrected by Optical Flow," International Journal of Informatics Society, vol. 13, no. 2, pp. 79-91, 2021. [Online].
Available: http://www.infsoc.org/journal/vol13/IJIS_13_2_079-091.pdf

[14]. B. Heo, K. Yun, and J. Y. Choi, "Appearance and Motion Based Deep Learning Architecture for Moving Object Detection in Moving Camera," in 2017 IEEE International Conference on Image Processing (ICIP), pp. 1827–1831, 2017.

[15]. A. Mohamed, S. Lakhanpal, R. Nii, P. Nagaveni, K. Binga, and T. R. Kumar, "Object Detection in Autonomous Driving Systems Using YOLOv5 and Kalman Filtering," in 2025 International Conference on Automation and Computation (AUTOCOM), 2025.

[16]. J. S. Kulchandani and K. J. Dangarwala, "Moving Object Detection: Review of Recent Research Trends," 2021.

[17]. F. Guo, J. An, and J. Duan, "Moving Object Detection in High Dynamic Scene for Visual Prosthesis," 2024.

[18]. G. A. Günther, P. H. A. da Cruz, J. P. B. de Assis, C. N. Silla Jr., M. E. Pellenz, and M. A. S. Teixeira, "Perception System for Autonomous Vehicles: Object Detection, Motion Tracking, and Future Position Prediction," 2024.

[19]. W. J. Kim and I.-S. Kweon, "Moving Object Detection and Tracking from Moving Camera," in 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2011.

[20]. P. Modi, D. Menon, A. Verma, and A. S. Areeckal, "Real-time Object Tracking in Videos using Deep Learning and Optical Flow," in Proc. 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT 2024), Bengaluru, India, Jan. 2024.

[21]. D. Dua, A. Bhat, and A. Girdhar, "A Review of Moving Object Detection Techniques for Nighttime," Delhi Technological University, 2023.

[22]. J. Huang, W. Zou, Z. Zhu, and J. Zhu, "Optical Flow Based Real-time Moving Object Detection in Unconstrained Scenes," arXiv preprint arXiv:1807.04890, 2018.

[23]. "YOLOv10-nano model and hybrid video processing pipeline," May 2025. [Online]. Available: https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/63186225/bacd54de-5290-45eb-8516-5653df41d254/paste.txt

[24]. Ultralytics, "YOLOv10: Real-Time End-to-End Object Detection," Apr. 2025. [Online]. Available: https://docs.ultralytics.com/models/yolov10/

[25]. YouTube, "YOLOv10: Train a Custom Model and Run Inference on Live Webcam," May 2024. [Online]. Available: https://www.youtube.com/watch?v=29tnSxhB3CY

[26]. "Moving Object Detection and Classification using Deep Learning," Propulsion Tech Journal, 2023. [Online].
Available: https://www.propulsiontechjournal.com/index.php/journal/article/download/5000/3437/8668

[27]. OpenCV, "Background Subtraction - OpenCV Documentation," Jan. 2025. [Online]. Available: https://docs.opencv.org/4.x/de/df4/tutorial_js_bg_subtraction.html

[28]. D. D. Bloisi, "Background Subtraction," OpenCV Documentation, Fig. [figure number], 2024. [Online]. Available: https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html

[29]. MathWorks, "optical Flow Farneback - Object for estimating optical flow using Farneback method," Jan. 2025. [Online].
Available: https://www.mathworks.com/help/vision/ref/opticalflowfarneback.html

[30]. A. Torralba, "Optical Flow," Visionbook: Computer Vision - A Modern Approach, MIT, Fig. 48.1, 2024. [Online]. Available: https://visionbook.mit.edu/optical_flow.html

[31]. A. Rosebrock, "YOLO Object Detection with OpenCV," PyImageSearch, Fig. [figure number], Nov. 12, 2018. [Online]. Available: https://pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/

[32]. Digital Ocean, "YOLOv10: Advanced Real-Time End-to-End Object Detection," Jan. 2025. Available: https://www.digitalocean.com/community/tutorials/yolov10-advanced-real-time-end-to-end-object-detection

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis_____

_____

Total Pages _____ Name of the Scholar_____

Supervisor (s)

(1)_____

(2)_____

(3)_____

Department_____

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: _____ Similarity Index: _____, Total Word Count: _____

Date: _____

Candidate's Signature **Signature of Supervisor(s)**

PlagCheck_Report-1.pdf

Delhi Technological University

Document Details

Submission ID
trn:oid:::27535:98226917

Submission Date
May 29, 2025, 12:57 AM GMT+5:30

Download Date
May 29, 2025, 1:03 AM GMT+5:30

File Name
PlagCheck_Report-1.pdf

File Size
1.6 MB

38 Pages

10,923 Words

58,571 Characters

1% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

10 Not Cited or Quoted 1%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.