**Mini Project Report on**



**Object Detection in Video Surveillance System**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name: Dikshant Singh Chib University Roll No.-2021193**

***Under the Mentorship of***

**Dr. Pawan Kumar Mishra**

**Associate Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Object Detection in Video Surveillance System”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Pawan Kumar Mishra , Associate Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name: Dikshant Singh Chib University Roll no.:2021193

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-3** |
| Chapter 2 | Literature Survey | **4-6** |
| Chapter 3 | Methodology | **7-8** |
| Chapter 4 | Result and Discussion | **9-11** |
| Chapter 5 | Conclusion and Future Work | **12-13** |
|  | References | **14** |

**Chapter 1**

**Introduction**

For keeping an eye on and safeguarding public areas, roads, and private properties, video surveillance is essential. In this context, the work focuses on the domain of vehicle detection in security cameras. Real-time vehicle detection and monitoring is extremely important for traffic control, law enforcement, and security applications. The goal of this project is to create a reliable and accurate system for detecting vehicles in video feeds by utilizing deep learning and sophisticated computer vision techniques.

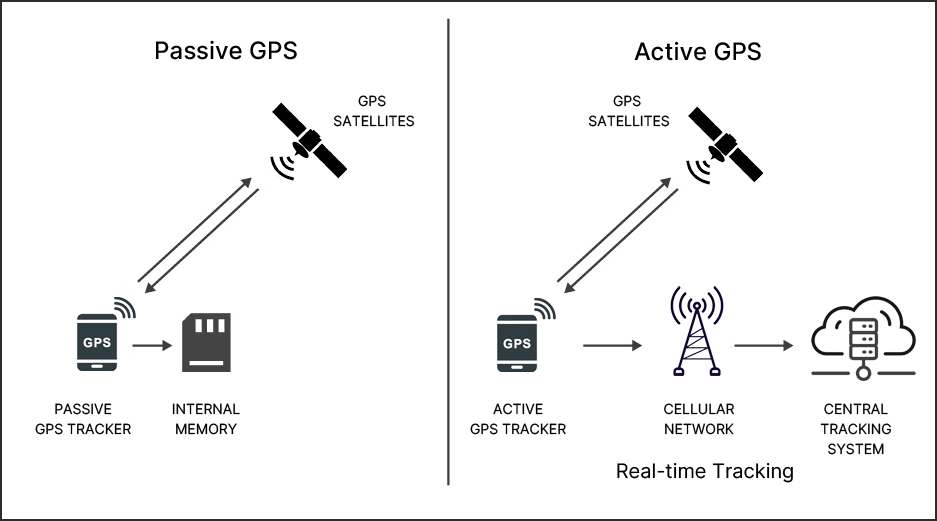
The effectiveness of vehicle detection in video surveillance systems is currently hampered by several issues. Difficulties include changing lighting, occlusions, and the requirement for real-time processing make it difficult to identify vehicles accurately and consistently. False positives can also result in inefficiencies in traffic monitoring and jeopardize the surveillance system's overall dependability.

* 1. **Introduction**

Foreground tracking and detection, several proposals have recently been made and have been effectively proven. These algorithms must, however, overcome challenges such abrupt changes and target drift that arise throughout the tracking process. Getting the most accurate and efficient estimation of object motion is the primary difficulty faced by motion tracking algorithms. Any surveillance application, including traffic control, medical imaging, video analysis, video communication, and military service, needs to have moving object detection. Video frames typically include both foreground and background information. The foreground information in a frame is made up of the feature points that are in the region of interest, while the background information is made up of the remaining feature points.

Two main components of a video surveillance system are typically motion detection and motion estimation. Since object detection is directly impacted by background information, it is the first and most important step. In video surveillance applications, the video data must be compressed as soon as possible because there is a lot of redundant and unnecessary information in the video across time and location. Reducing the video's temporal and spatial repetitions will help with compression. In the past, video data was compressed using either frame reduction or frame skipping, resulting in a slight drop in video quality. Modern video coding standards include motion compensation and 2D orthogonal transformations to eliminate temporal and spatial redundancy. Because of its maximum energy compaction, the 2D discrete cosine transform is employed for video compression in the suggested method. The two main components of a video surveillance system are motion detection and motion estimation. While motion estimation computes motion vectors to predict the positions of moving objects, motion detection identifies the moving item by extracting changes in object boundaries. By determining the displacement of coordinates of the best match in a reference frame for the block in a current frame, the optimal motion vector is investigated.

Innovation has transformed communication and collaboration amongst people. Fundamentally, vehicle tracking frameworks allow for the following of cars as their name suggests. They use a variety of innovations to monitor a vehicle's condition over time or to create a background based on the locations of vehicles. These frameworks are used by a variety of businesses and are also an essential component of most stolen car recovery techniques. A major advancement in both conventional driver safety and the safety of the fleet of vehicles is vehicle tracking. As the innovation becomes more widely available and affordable, it will become increasingly significant for traffic safety. Vehicle monitoring comes in two flavors, both of which are useful under certain conditions.

**Passive vehicle tracking:** these devices typically use a GPS device to log a vehicle's location over time. The data can be transferred to a PC and destroyed at the moment the tracker is removed. While they have various uses, the following frameworks are beneficial for fleet management.

**Active vehicle tracking:** tracking systems get more complex as they broadcast a vehicle's location over time. This data is typically observed from a focal point for fleet management and dispatch purposes. This type of framework can also be applied.

**Figure 1.1** Active & Passive GPS tracking system

to recover stolen vehicles.

The suggested system continuously monitors traffic, particularly cars on the road or highway, using CCTV cameras positioned at a specific height (on the bridge). Every frame of the real-time CCTV video stream is subjected to the object detection algorithm. The object detection system begins tracking each identified vehicle on each frame if it finds a vehicle (car). The suggested system can also be used to escort a specific car in the autonomous vehicle sector. Autonomous vehicles are able to make choices based on information from tracked vehicles.

Tracking is one of the most important areas in computer vision. Estimating an object's position over continuous image sequences is known as tracking.

Additionally, this is divided into two subcategories: tracking of a single object and tracking of multiple objects. They both require slightly different approaches.

A road with cars on it

Description automatically generated

**Figure 1.2** Vehicle Detection And counting Number of vehicles.

**Chapter 2**

**Literature Survey**

**2.1 Literature Review**

One of the most important tasks in computer vision is object detection, which is recognizing and categorizing objects in pictures or videos. Although they were fundamental, traditional methods like Haar cascades and HOG with SVM had trouble handling complex situations. Object detection has been transformed by the introduction of deep learning, a branch of machine learning that uses multi-layered neural networks. Among the most important are Convolutional Neural Networks (CNNs), which enable the automated extraction of hierarchical features from images. Prominent models like SSD, Faster R-CNN, and YOLO demonstrate the power of deep learning by allowing end-to-end learning of complex representations. Using pre-trained models in transfer learning has become commonplace to improve performance, particularly when labeled datasets are scarce. Deep learning is driving advances in computer vision, finding applications in image classification, natural language processing, and other fields. However, there are still challenges, such as the need for large amounts of labeled data and computational resources. These advances will shape the future of these fields.

**2.1.1 Traditional Approaches to Object Detection:**

Traditionally, object detection has been mostly accomplished by standard computer vision algorithms, such as Haar cascades and Histogram of Oriented Gradients (HOG). These methods comprised creating features by hand and classifying data using algorithms like Support Vector Machines (SVM). Although they worked well in some cases, they had trouble with scale changes and occlusions, which limited their use in other circumstances.

For example, HOG used image intensity gradient analysis to capture the appearance and form of local objects. In contrast, Haar cascades employed a series of basic classifiers in a cascaded fashion to recognize objects according to attributes such as edges and textures. Nevertheless, the requirement for characteristics created by humans presented challenges, particularly when dealing with intricate and diverse situations.

**2.1.2 Rise of Deep Learning in Object Detection:**

Convolutional neural networks (CNNs) represented a paradigm shift towards deep learning that revolutionized object detection. Deep learning models have demonstrated better performance than conventional techniques by automatically extracting hierarchical features from data. The limitations of manually created features were overcome by CNNs, which excelled in handling a variety of complex scenarios due to their ability to recognize intricate patterns and representations.

**2.1.3 Pioneering Models in Deep Learning:**

Modern deep learning models designed for object detection have been made possible by recent developments. A unified approach to real-time detection was provided by innovative models such as YOLO (You Only Look Once), which processed entire images in a single forward pass. Additional architectures that increased accuracy and efficiency were Faster R-CNN and Single Shot Multibox Detector (SSD). These models fixed the flaws in previous approaches and established new standards for object detection performance by optimizing the trade-off between speed and precision.

**2.1.4 Transfer Learning and Pre-trained Models:**

The extensive use of transfer learning in object detection is one noteworthy development in recent research. Leveraging pre-trained models on massive datasets such as ImageNet is known as transfer learning. This method makes it easier to transfer learned features from one domain to another, which is especially useful for tasks that require handling sparsely labeled data. Transfer learning improves the accuracy, robustness, and generalization of object detection systems by leveraging the knowledge gathered from large datasets during the training phase.

**2.1.5 Real-Time Object Detection:**

The need for real-time applications was also highlighted by the development of object detection. The limited processing speed of traditional methods made them less useful in dynamic environments. For example, YOLO brought about a paradigm change by making it possible to detect objects in real time by processing images in a single pass. Applications in surveillance, video analysis, and other fields where real-time processing is essential have been made possible by this breakthrough.

**2.2 Challenges in tracking systems**

Object tracking is a fundamental task in computer vision, crucial for applications such as surveillance, autonomous vehicles, and augmented reality. However, several challenges impede the seamless tracking of objects in dynamic environments.

Understanding the difficulties, we must overcome while tracking is vital. Several prevalent and significant obstacles are listed below.

**2.2.1 Object Occlusion:**

When one object in a series of photos obscures another, it's known as object occlusion, and it breaks the tracking continuity. Occlusion is a regular occurrence in real-world circumstances such as surveillance or crowded environments. This creates two challenges: the requirement to reidentify the object upon its reappearance, and the loss of visual information during occlusion. Robust tracking performance is ensured even in the event of occlusion thanks to advanced tracking algorithms that address this through multi-object tracking and object re-identification.

**2.2.2 Change of Shape:**

It becomes more difficult to track non-rigid objects since they can flex or change shape. When the tracked item experiences considerable form changes, like in the case of flexible elements like human beings or moving fabrics, traditional algorithms find it difficult to adjust. Deformable part models and shape-context-based tracking are used as solutions to precisely capture the non-rigid nature of the item and maintain tracking continuity in spite of shape fluctuations.

**2.2.3 False Positives:**

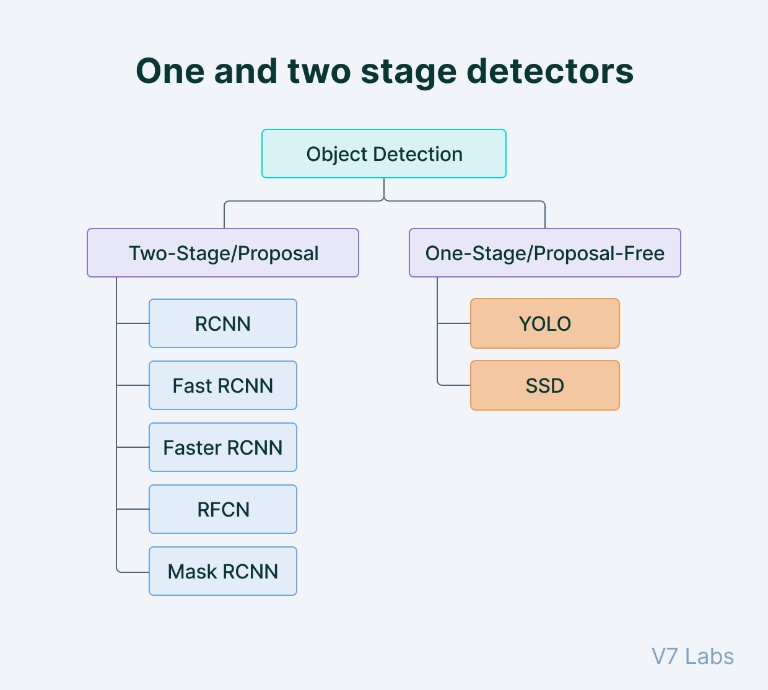
In settings with visually identical objects, false positives—where the tracker mistakenly identifies a non-target object—are frequent. The algorithm may begin tracking the incorrect entity because of this confusion, which might cause tracking problems. Tracking algorithms use machine learning for object categorization, contextual data, and tracking-by-detection techniques to mitigate false positives. By improving the capacity to discern between comparable items, these methods guarantee precise tracking and guard against the tracker being duped by incorrect identifications.

**Chapter 3**

**Methodology**

**3.1 Model and Setup**

**3.1.2 YOLOv8n Model**

The model used in this implementation is YOLOv8n, a variant of the YOLO (You Only Look Once) object detection model. The weights for the YOLOv8n model are loaded from the file "yolov8n.pt".

**3.2.2 Dependencies**

The implementation utilizes the following libraries:

1. OpenCV (cv2) for video processing.
2. NumPy for numerical operations.
3. Ultralytics for YOLO model integration.
4. matplotlib. pyplot for plotting graphs

**Figure 3.1** object detection architectures

**3.2 Data Preprocessing**

**3.2.1 Class List**

The names of the object classes that the YOLOv8n model can identify are contained in the class list, which is loaded from the file "coco.txt." The Microsoft Common Objects in Context (COCO) dataset, which is frequently used for object detection and segmentation tasks, is referred to as "coco" in several contexts.

**3.2.2 Color Generation**

Bounding box colors are produced at random for every class, resulting in a visually distinct set of hues. The output frames' display of the items that have been detected depends on this stage.

An RGB color is produced at random for every class in the class list. This is done in order to give each class's bounding boxes a distinct and identifiable hue. The detection colors list then contains the created colors.

**3.3 Video Processing**

**3.3.1 Frame Resizing**

The video frames are resized to a width of 640 pixels and a height of 480 pixels for optimized processing.

**3.3.2 Object Detection**

The YOLOv8n model is used to predict objects in each frame with a confidence threshold of 0.45. Detected objects are then drawn on the frame with bounding boxes, class labels, and confidence scores.

**3.3.3 Line Crossing Detection**

A horizontal line is drawn on the frame at position 550 pixels. The system counts vehicles that cross this line based on the center of their bounding boxes.

**Chapter 4**

**Result and Discussion**

**4.1 Result**

The YOLOv8n deep learning model is used by the implemented Python script to recognize vehicles in real-time within a video stream. The Ultralytics library is used to integrate the YOLOv8n model, which demonstrates effective object recognition skills and provides bounding box visualizations, class labels, and confidence scores.

**4.1.1 Key Features**

**Vehicle Counting:**

* A vehicle counting system built into the script counts how many cars cross a given line in the video.
* Each frame shows the counts in real time, and the cumulative counts are charted over time.

**Visualizations:**

* Around identified vehicles, bounding boxes are depicted together with confidence scores and class names.
* Marking the centers of bounding boxes adds more spatial information.

**4.1.2 Statistical Analysis**

**Confidence Score Distribution:**

* To help evaluate the forecast reliability of the model, a histogram shows the distribution of confidence ratings for items that have been discovered.

**Vehicle Count Over Time:**

* Throughout the video, a line plot that shows the temporal evolution of vehicle counts provides insights into traffic patterns.

**Bounding Box Size Distribution:**

* The bounding box size distribution is shown in a histogram, which offers details on the size variations of objects that have been detected.

**4.2 Graphs**

A graph with green lines

Description automatically generated

**Figure 4.1** Graph between Number of vehicles and frame Number

**A blue graph with white text

Description automatically generated**

**Figure 4.2** Graph between Frequency and Confidence Scores

**A graph of a distribution of a number of boxes

Description automatically generated**

**Figure 4.3** Graph between Frequency and Bounding Box Area

**Chapter 5**

**Conclusion and Future Work**

**5.1 Conclusion**

Based on the YOLOv8n model, the object detection system implemented in the provided code effectively identifies and tracks objects in video streams. Important features for surveillance applications are added by integrating line-crossing detection, random color coding for class distinction, and real-time processing.

The literature review emphasized the development of object detection and the role deep learning models, such as YOLO, play in producing precise and effective outcomes. A thorough grasp of the system's architecture is aided by the investigation of transfer learning, model integration with OpenCV, and real-world application consideration.

There are multiple avenues that show promise for future research. Improvements in multi-object tracking, real-time performance, and model optimization may raise the system's capabilities even higher. Investigating adaptive learning, privacy-preserving methods, and anomaly detection would address new issues in surveillance applications. Contextual data, behavior analysis, and enhanced object recognition under difficult circumstances could all be integrated to improve the system's resilience and dependability.

These future work areas offer a roadmap for improving object detection capabilities, guaranteeing adaptability to various scenarios, and addressing new needs in the field as video surveillance systems continue to develop. Substantial algorithms, model optimization, and user-focused features working together will be essential in forming the next wave of intelligent video surveillance systems.

The object detection system that has been put into place is an example of how technology can enhance surveillance capabilities and promote safer environments and more intelligent decision-making. The convergence of deep learning, computer vision, and practical applications is expected to produce increasingly more advanced and morally sound surveillance systems as this field's research and development continue. The process of developing intelligent video surveillance is dynamic, and this work is a significant advancement in that process.

**5.2 Future Work**

1. Model Optimization: Investigate optimization strategies to improve the YOLOv8n model's effectiveness. For deployment on edge devices, this can entail quantization, model pruning, or format conversion to a lighter version of the model.
2. Hyperparameter Tuning: To possibly increase the model's precision and recall, perform a thorough hyperparameter search to fine-tune parameters like confidence thresholds, anchor box sizes, and other model-specific parameters.
3. Data Augmentation: To make the model more resilient to changes in lighting, scale, and orientation, incorporate extra data augmentation techniques during training. Improved generalization on a variety of datasets may result from this.
4. Transfer Learning Variants: Try out various transfer learning techniques, like applying more recent state-of-the-art object detection models or pre-trained models from other domains.
5. Real-Time Stream Processing: Optimize code efficiency, investigate parallel processing methods, or make use of specialized hardware such as GPUs or TPUs to improve the system's ability to process video streams in real-time.
6. Tracking Objects: To track objects between frames, use object tracking algorithms. This can lead to more coherent object trajectories and enhance the system's ability to handle occlusions.
7. Analysis Based on Class: Analyze detection performance for individual classes in greater detail. Determine which classes have poorer detection accuracy and investigate ways to make those classes more detectable.
8. Comprehending the Scene: Expand the system's capabilities to include object detection and scene context understanding. For a more thorough understanding of the environment, this might entail integrating with other AI systems or implementing semantic segmentation.
9. Deployment on Edge Devices: Make the model as efficient as possible for deployment on edge servers or Internet of Things devices. This entails compressing the model, quantizing it, and modifying the system to operate effectively with limited resources.
10. User Interface and Interaction: Create an intuitive user interface that allows users to examine statistics, analyze and interact with the objects that have been detected, and visually represent the results of object detection.

**References**

[1] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[2] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., & Reed, S. (2016). SSD: Single Shot Multibox Detector. European Conference on Computer Vision (ECCV).

[3] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision.

[4] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition

[5] Girshick, R. (2015). Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[8] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[9] Hofmann, M., & Tiefenbacher, P. (2018). Improving YOLO. arXiv preprint arXiv:1804.06816.

[10] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence.