

Deep CNN Framework for Object Detection and Classification System from Real Time Videos

Dr Subba Reddy Borra¹, B Gayatri², B Rekha², B Akshitha²

¹Professor, ²UG Students, Department of Information Technology

^{1,2}Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India.

To Cite this Article

Dr Subba Reddy Borra, B Gayatri, B Rekha, B Akshitha, "Deep CNN Framework for Object Detection and Classification System from Real Time Videos" *Journal of Science and Technology*, Vol. 08, Issue 12 - Dec 2023, pp78 - 93

Article Info

Received: 22-11-2023

Revised: 02-12-2023

Accepted: 12-12-2023

Published: 22-12-2023

Abstract

In today's world, accurately counting and classifying vehicles in real-time has become a critical task for effective traffic management, surveillance, and transportation systems. It plays a crucial role in optimizing road infrastructure, enhancing safety measures, and making informed decisions for traffic planning. With the ever-increasing traffic congestion and road safety concerns, the demand for a robust and automated vehicle counting and classification system has grown significantly. Traditionally, vehicle counting, and classification involved manual deployment of sensors or fixed cameras at specific locations. However, these methods had limitations in handling complex traffic scenarios, especially in real-time, and were less efficient in dealing with varying environmental conditions, occlusions, and different vehicle types. Fortunately, recent advancements in deep learning models have revolutionized object detection, making real-time vehicle counting and classification achievable. One such model is the YOLO (You Only Look Once) algorithm based on the Darknet framework. Leveraging the power of this model, a real-time vehicle counting, and classification system has been developed, utilizing the OpenCV library. The system employs a pretrained YOLO model to detect the number of vehicles present in a given video and classifies the type of each vehicle. By doing so, it eliminates the need for extensive human intervention and ensures automated and accurate counting of vehicles in real-time. Moreover, this system excels in handling varying traffic conditions and different vehicle types, which enhances its accuracy and reliability. The benefits of this proposed system are numerous. It provides valuable data for traffic analysis, enabling better traffic management strategies and improved infrastructure planning. With this system in place, authorities can efficiently address traffic congestion, implement targeted safety measures, and optimize traffic flow. Further, the integration of the YOLO algorithm within the Darknet framework in the proposed system has opened new possibilities for real-time traffic management. By leveraging deep learning, this system offers a reliable and efficient solution to the challenges posed by modern traffic scenarios, helping to create safer and more organized road networks for everyone.

Keywords: Object Detection, CNN framework, YOLO, Darknet framework.

1. INTRODUCTION

In the realm of computer vision, the development of a robust system for object detection and classification in real-time videos is a significant endeavor. This involves the utilization of a sophisticated Deep Convolutional Neural Network (CNN) framework, a type of artificial intelligence architecture specifically designed to excel at visual tasks. The primary objective of this system is twofold: firstly, to accurately detect objects within a given video stream, and secondly, to classify these detected objects into predefined categories. The real-time nature of the videos adds an extra layer of complexity, requiring the system to process and analyze frames swiftly and efficiently.

In simpler terms, imagine a smart system that can automatically identify and categorize various objects appearing in videos as they unfold in real-time. This could find applications in diverse fields such as surveillance, autonomous vehicles, or even in enhancing user experiences in entertainment and gaming. The cornerstone of this technology lies in the deep learning capabilities of the CNN framework. Deep learning enables the system to learn intricate patterns and features from vast amounts of data, allowing it to recognize and differentiate objects with a high degree of accuracy. By combining the power of deep learning with the real-time processing of videos, this framework promises to open new avenues for applications where swift and precise object detection and classification are crucial. The ensuing sections will delve into the technical aspects, methodologies, and potential applications of this advanced system, shedding light on its significance in the evolving landscape of computer vision and artificial intelligence. The impetus for undertaking research in the development of a Deep Convolutional Neural Network (CNN) framework for real-time object detection and classification in videos stems from the pressing need for advanced and efficient computer vision systems. In contemporary scenarios, there is an escalating demand for intelligent systems capable of swiftly analyzing and comprehending dynamic visual information, especially in domains like surveillance, autonomous systems, and interactive media. Traditional computer vision methods have exhibited limitations in handling the complexity of real-time video streams with a multitude of objects. The motivation arises from the recognition that a more sophisticated approach, rooted in deep learning, is required to surmount these challenges effectively. Real-time video analysis demands not only accurate detection of objects within frames but also rapid classification, a task that necessitates a high level of computational efficiency. The proposed CNN framework aims to address these requisites by leveraging the hierarchical and discriminative features learned through deep neural networks. The growing ubiquity of video data in various applications intensifies the need for systems that can discern and interpret visual content with precision. Applications such as video surveillance require swift and reliable object detection to ensure timely responses to potential threats. Similarly, in autonomous vehicles, the ability to rapidly identify and classify objects in the vehicle's vicinity is imperative for safe navigation. This research is motivated by the conviction that a well-designed CNN framework, adept at handling the intricacies of real-time video analysis, can significantly enhance the capabilities of computer vision systems. The overarching goal is to contribute to the development of intelligent systems that can operate seamlessly in dynamic environments, catering to the increasing demands for accuracy and speed in object detection and classification within real-time video streams.

2. LITERATURE SURVEY

Image classification, as a classical research topic in recent years, is one of the core issues of computer vision and the basis of various fields of visual recognition. The improvement of classification network

performance tends to significantly improve its application level, for example to object-detection, segmentation, human pose estimation, video classification, object tracking, and super-resolution technology. Improving image classification technology is an important part of promoting the development of computer vision. Its main process includes image data preprocessing, feature extraction and representation, and classifier design.

The focus of image classification research has always been image feature extraction, which is the basis of image classification. Traditional image feature extraction algorithms focus more on manually setting specific image features

In the last decade, modern video surveillance systems have attracted increasing interest with several studies focusing on automated video surveillance systems, which involve a network of surveillance cameras with sensors that can monitor human and nonhuman objects in a specific environment. Pattern recognition can be used to find specific arrangements of features or data, which usually yield details regarding a presented system or data set. In a technical context, a pattern can involve repeating sequences of data with time, and patterns can be utilized to predict trends and specific featural configurations in images to recognize objects. Many recognition approaches involving the use of the support vector machine (SVM) [5], artificial neural network (ANN) [6], deep learning [7], and other rule-based classification systems have been developed. Performing classification using an ANN is a supervised practical strategy that has achieved satisfactory results in many classification tasks. The SVM requires fewer computational requirements than the ANN; however, the SVM provides lower recognition accuracy than the ANN. In recent years, networks have played a significant role in a wide range of applications, and they have been applied to surveillance systems. In recent years, as the amounts of unstructured and structured data have increased to big data levels, researchers have developed deep learning systems that are basically neural networks with several layers. Deep learning allows one to capture and mine larger amounts of data, including unstructured data. This approach can be used to model complicated relationships between inputs and outputs or to find patterns. However, the associated accuracy and classification efficiency are generally low [8]. Many strategies have been developed to increase the recognition accuracy. In this work, we discuss the accuracy gains from adopting certain saliency methods to improve the recognition and detection of an object and isolate it from a scene.

The performance efficiency of existing surveillance systems is highly dependent on the activity of human operators who are responsible for monitoring the camera footage. In general, most medium and large surveillance systems involve numerous screens (approximately 50 or more) that display the streams captured by numerous cameras. As the number of simultaneous video streams to be viewed increases, the work of surveillance operators becomes considerably challenging and fatiguing. Practically, after twenty minutes of continuous work, the attention of the operators is expected to degrade considerably. In general, the operators check for the absence or presence of objects (i.e., people and vehicles) in surveillance areas and ensure that the maximum capacity of a place remains intact, such as by ensuring that no unauthorized people are present in restricted areas and no objects are present in unexpected places. The failures of such systems in alarming authorities can be attributed to the limitations of manual processing. Generally, most traditional methods used to obtain evidence depend heavily on the records of the security camera systems in or near accident sites. Practically, when an incident occurs in a vast space or considerable time has elapsed since its occurrence, it is difficult to find any valuable evidence pertaining to the perpetrators from the large number of surveillance videos, which hinders the resolution of the cases. Thus, to minimize the

mental burden of the operators and enhance their attention spans, it is desirable that an automated system that can reliably alert an operator of the presence of target objects (e.g., a human) or the occurrence of an anomalous event be developed.

Pattern recognition, which is widely used in many recognition applications, can be performed to find arrangements of features or data, and this technique can be applied in the surveillance domain. Several recognition approaches involving the support vector machine, artificial neural networks, decision trees, and other rule-based classification systems have been proposed. Machine learning typically uses two types of approaches, namely, supervised and unsupervised learning. Using these approaches, especially supervised learning, we can train a model with known input and output data to ensure that it can estimate any future output. Moreover, in some existing systems, an artificial immune system (AIS)-inspired framework, where the AIS is a computational paradigm that is a part of the computational intelligence family and is inspired by the biological immune system that can reliably identify unknown patterns within sequences of input images, has been utilized to achieve real-time vision analysis designed for surveillance applications.

The field of video surveillance is very wide. Active research is ongoing in subjects, such as automatic thread detection and alarms, large-scale video surveillance systems, face recognition and license plate recognition systems, and human behavior analysis. Intelligent video surveillance is of significant interest in industry applications because of the increased requirement to decrease the time it takes to analyze large-scale video data. Relating to the terminology, Elliott [9] recently described an intelligent video system (termed IVS) as “any kind of video surveillance method that makes use of technology to automatically manipulate process and/or achieved actions, detection, alarming and stored video images without human intervention.” Academic and industry studies are focused on developing key technologies for designing powerful intelligent surveillance systems along with low-cost computing hardware; and the applications include object tracking [10], pedestrian detection, gait analysis, vehicle recognition, privacy protection, face and iris recognition, video summarization, and crowd counting. Nguyen [11] described the implementation and design of an intelligent low-cost monitoring system using a Raspberry Pi and a motion detection algorithm programmed in Python as a traditional programming environment. Additionally, the system utilizes the motion detection algorithm to considerably reduce storage usage and save expense costs. The motion detection algorithm is executed on a Raspberry Pi that enables live streaming cameras together with motion detection. The real-time video camera can be viewed from almost any web browser, even by mobile devices. Sabri et al. [12] present a real-time intruder monitoring system based on a Raspberry Pi to deploy a surveillance system that is effective in remote and scattered places, such as universities. The system hardware consists of a Raspberry Pi, long-distance sensors, cameras, a wireless module and alert circuitry; and the detection algorithm is designed in Python. This system is a novel cost-effective solution with good flexibility and improvement needed for monitoring pervasive remote locations. The results show that the system has high reliability for smooth working while using web applications; in addition, it is cost-effective. Therefore, it can be integrated as several units to catch and concisely monitor remote and scattered areas. Their system can also be controlled by a remote user geographically or sparsely far from any networked workstation. The recognition results prove that the system efficiently recognized intruders and provided alerts when detecting intruders at distances between one to three meters from the system camera. The recognition accuracy is between 83% and 95% and the reliable warning alert is in the range of 86–97%. Turchini et al. [13] proposes an object tracking system that was merged with their lately developed abnormality detection system to provide protection and intelligence for critical regions.

In recent years, many studies have focused on using artificial intelligence for intelligence surveillance systems. These techniques involve different approaches, such as the SVM, the ANN, and the latest developed types based on deep learning techniques. However, deep neural networks are computationally challenging and memory hungry; therefore, it is difficult to run these models in low computational systems, such as single board computers [14]. Several approaches have been utilized to address this problem. Many approaches have reduced the size of neural networks and maintained the accuracy, such as MobileNet, while other approaches minimize the number of parameters or the size [15].

3. PROPOSED SYSTEM

The research work begins with the acquisition of real-time video streams as the primary input source, typically sourced from surveillance cameras or traffic monitoring systems. The first module, Background Subtraction and ROI, plays a crucial role in isolating moving objects, i.e., vehicles, from the stationary background. This is achieved using advanced algorithms to create a Region of Interest (ROI) that narrows down the area for subsequent analysis, reducing computational load and minimizing false positives. The heart of the system lies in the Vehicle Detection and Tracking module, which employs the YOLO (You Only Look Once) deep learning model. YOLO excels at real-time object detection and tracking, enabling the system to identify vehicles within the defined ROI and track their movements across frames, allowing for continuous monitoring. The next step involves the Vehicle Classification module, powered by the Darknet framework, which classifies the detected vehicles into specific categories, such as cars, trucks, or motorcycles. Following this, the Counting and Analytics module quantifies vehicle movements, including counting, speed measurement, and other relevant analytics, providing valuable data for traffic management and research purposes. Finally, the system generates a video output that overlays processed information onto the original video feed, offering a user-friendly visual representation of the vehicle counting and classification results, making it a versatile tool for various applications in traffic monitoring, urban planning, and transportation research. Figure 4.1 shows the proposed system model. The detailed operation illustrated as follows:

Step 1: This is the starting point of system. Acquire real-time video streams as input data for the system. These video streams could come from surveillance cameras, traffic cameras, or any source capturing vehicle movements.

Step 2: Background subtraction is a crucial step for isolating moving objects (vehicles) from the stationary background. In this module, use algorithms and techniques to detect the background and create a Region of Interest (ROI) where vehicle detection will occur. This step helps reduce noise and focus on the relevant area.

Step 3: Use YOLO (You Only Look Once), a deep learning-based object detection model, for vehicle detection. YOLO can efficiently detect and locate objects in real-time video frames. It identifies the vehicles within the defined ROI and can track their movements across frames, allowing us to follow vehicles as they move through the video.

Step 4: After detecting and tracking vehicles, further analyze and classify them using the Darknet framework. Darknet is a neural network framework well-suited for classification tasks. It can classify vehicles into different categories such as cars, trucks, motorcycles, or any other relevant classes.

Step 5: In this step, count and analyze the detected and classified vehicles. We can track the number of vehicles passing through specific points or regions of interest, calculate vehicle speed, and gather other relevant analytics data. This information can be useful for traffic management, surveillance, or research purposes.

Step 6: Finally, the system provides video output with the processed information overlaid on the original video feed. This output can include counted vehicles, their classifications, and any other relevant data. It allows users to visualize and interpret the results of the vehicle counting and classification system.

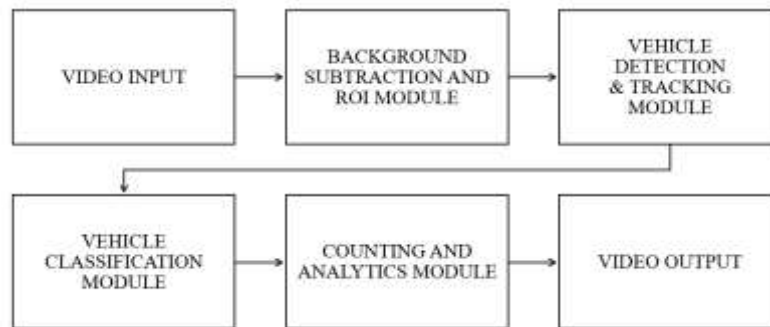


Figure 1: Block diagram of proposed system.

3.1 Background Subtraction

This is the first module in the system whose main purpose is to learn about the background in a sense that how it is different from the foreground. Furthermore, as proposed system works on a video feed, this module extracts the frames from it and learns about the background. In a traffic scene captured with a static camera installed on the roadside, the moving objects can be considered as the foreground and static objects as the background. Image processing algorithms are used to learn about the background using the above-mentioned technique.

3.2 Vehicle Detection and Counting

The third and the last module in the proposed system is classification. After applying foreground extraction module, proper contours are acquired, Features of these contours such as centroid. Aspect ratio, area, size and solidity are extracted and are used for the classification of the vehicles. This module consists of three steps, background subtraction, image enhancement and foreground extraction. Background is subtracted so that foreground objects are visible. This is done usually by static pixels of static objects to binary 0. After background subtraction image enhancement techniques such as noise filtering, dilation and erosion are used to get proper contours of the foreground objects. The result obtained from this module is the foreground.

Region of Interest selection: In the very first frame of the video, I define a ROI by drawing a close line on the image. The goal is to recognize that ROI in a later frame, but that ROI is not a salient vehicle. It is just a part of a vehicle, and it can deform, rotate, translate and even not be fully in the frame.

Vehicle Detection: Active strategy to choose a search window for vehicle detection using an image context was proposed GMM framework to capture the vehicle by sequential actions with top-down

attention. It has achieved satisfactory performance on vehicle detection benchmark, by sequentially refining the bounding boxes. Proposed a sequential search strategy to detect visual vehicles in images, where the detection model was trained by proposed a deep RL framework to select a proper action to capture a vehicle in an image.

Vehicle Counting: In this module detected vehicles will be counted and these counted results will be updated frequents based on vehicle detection, results will be printed streaming video using OpenCV.

3.4 YOLO-V3 Model

Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos. Some of the objects detected include people, cars, chairs, stones, buildings, and animals. This phenomenon seeks to answer two basic questions:

1. What is the object? This question seeks to identify the object in a specific image.
2. Where is it? This question seeks to establish the exact location of the object within the image.

Object detection consists of various approaches such as fast R-CNN, Retina-Net, and Single-Shot MultiBox Detector (SSD). Although these approaches have solved the challenges of data limitation and modeling in object detection, they are not able to detect objects in a single algorithm run. YOLO algorithm has gained popularity because of its superior performance over the aforementioned object detection techniques.

YOLO Definition: YOLO is an abbreviation for the term ‘You Only Look Once’. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run. CNN is used to predict various class probabilities and bounding boxes simultaneously. The YOLO algorithm consists of various variants. Some of the common ones include tiny YOLO and YOLOv3.

Importance of YOLO: YOLO algorithm is important because of the following reasons:

- Speed: This algorithm improves the speed of detection because it can predict objects in real-time.
- High accuracy: YOLO is a predictive technique that provides accurate results with minimal background errors.
- Learning capabilities: The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

YOLO algorithm working: YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection Over Union (IOU)

Residual blocks: First, the image is divided into various grids. Each grid has a dimension of $S \times S$. The following Figure 2 shows how an input image is divided into grids. In the Figure 2, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.



Figure: 2. Example of residual blocks.

Bounding box regression: A bounding box is an outline that highlights an object in an image. Every bounding box in the image consists of the following attributes:

- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
- Bounding box center (bx,by)

The following Figure 3 shows an example of a bounding box. The bounding box has been represented by a yellow outline. YOLO uses a single bounding box regression to predict the height, width, center, and class of objects. In the image above, represents the probability of an object appearing in the bounding box.

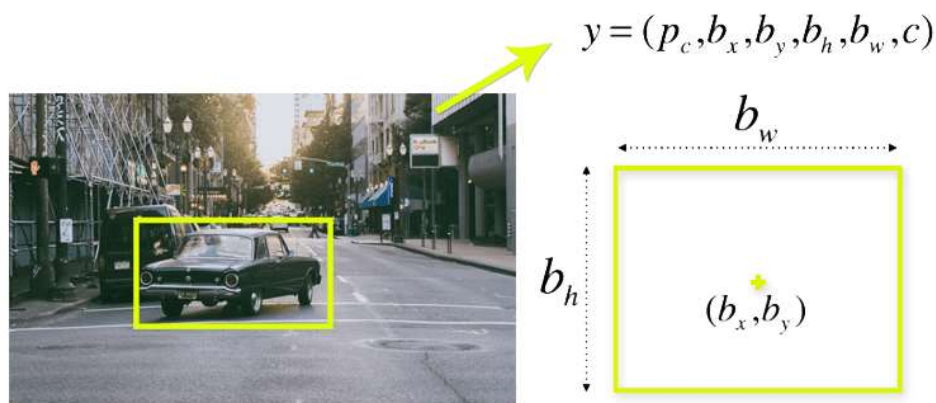


Figure 3. Bounding box regression

Intersection over union (IOU): Intersection over union (IOU) is a phenomenon in object detection that describes how boxes overlap. YOLO uses IOU to provide an output box that surrounds the objects perfectly. Each grid cell is responsible for predicting the bounding boxes and their confidence scores. The IOU is equal to 1 if the predicted bounding box is the same as the real box. This mechanism eliminates bounding boxes that are not equal to the real box.

Combination of the three techniques: The following image shows how the three techniques are applied to produce the final detection results.

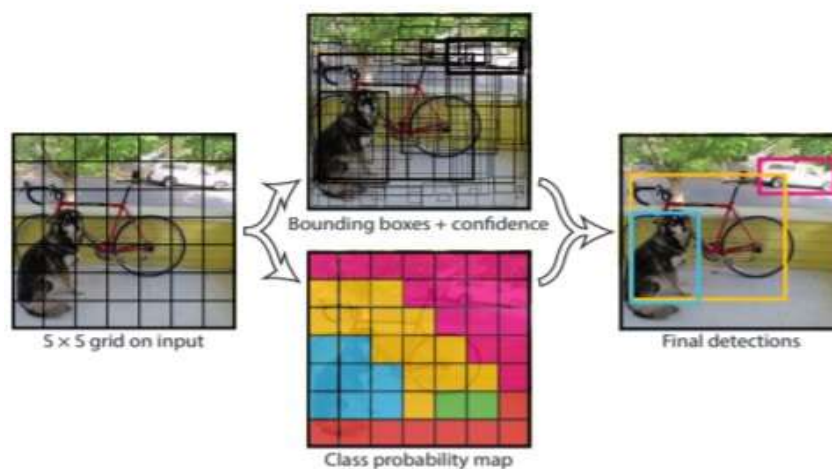


Figure 4. Combination of three modules.

First, the image is divided into grid cells. Each grid cell forecasts B bounding boxes and provides their confidence scores. The cells predict the class probabilities to establish the class of each object. For example, we can notice at least three classes of objects: a car, a dog, and a bicycle. All the predictions are made simultaneously using a single convolutional neural network. Intersection over union ensures that the predicted bounding boxes are equal to the real boxes of the objects. This phenomenon eliminates unnecessary bounding boxes that do not meet the characteristics of the objects (like height and width). The final detection will consist of unique bounding boxes that fit the objects perfectly. For example, the car is surrounded by the pink bounding box while the bicycle is surrounded by the yellow bounding box. The dog has been highlighted using the blue bounding box.

Importing Libraries:

- `numpy`: For numerical operations.
- `imutils`: A library for basic image processing tasks.
- `time`: For time-related operations.
- `scipy.spatial`: For spatial distance calculations.
- `cv2`: OpenCV library for computer vision tasks.
- `input_retrieval`: A module that seems to handle command-line argument parsing, but it's not provided here.
- List of Vehicles:
`list_of_vehicles`: A list of strings representing different types of vehicles.
- Constants:
`FRAMES_BEFORE_CURRENT`: An integer constant set to 10.
- Parsing Command Line Arguments:
It is to retrieve various parameters like labels, file paths, confidence thresholds, and GPU usage flag from the command line using a function called `parseCommandLineArguments()`. However, the details of this function are not provided.
- Setting up Random Colors:
Generates random RGB colors for different classes.
- Displaying Vehicle Count:
`displayVehicleCount()`: A function to display the vehicle count on the frame.
- Box and Line Overlap Check:
`boxAndLineOverlap()`: A function to check if a bounding box overlaps with a given line.
- Displaying FPS:
`displayFPS()`: A function to display frames per second.
- Drawing Detection Boxes:
`drawDetectionBoxes()`: Draws bounding boxes, labels, and green dots at the center of the detected objects.
- Initializing Video Writer:
`initializeVideoWriter()`: Initializes a video writer for saving the processed frames.
- Box in Previous Frames Check:
`boxInPreviousFrames()`: Checks if a box from the current frame has appeared in previous frames.
- Counting Vehicles:
`count_vehicles()`: Counts vehicles and updates their IDs.
- Loading YOLO Model:
Reads the YOLO model from the provided configuration and weights files.
- Using GPU (if specified).
- Getting Layer Names:
`In`: Gets the layer names from the YOLO network.
- Initializing Video Stream:
Captures video from the provided input file.
- Setting Up Line Coordinates:
`x1_line`, `y1_line`, `x2_line`, `y2_line`: Define coordinates for a default line.
- Initialization:
Initializes various variables and data structures.

- Main Loop:
Iterates over frames from the video stream.
- Performs object detection, vehicle counting, and classification.
Draws boxes, updates vehicle count, and displays frames.
- GUI using Tkinter:
Creates a GUI window for uploading videos.
Includes a button to trigger real-time object detection and counting.
- Button Actions:
upload_video(): Opens a file dialog to select a video file.
classify(file_path): Calls a function to initiate object detection on the uploaded video.
- Running the GUI:
Starts the Tkinter main loop to run the GUI.

4.2 Results and Description

Figure 6 represents a single frame from a video. In this frame, there are two vehicles detected: a bus and a car. The object detection model has identified a car in the frame, and it has assigned a high confidence score to this classification, indicating a 97% accuracy in identifying it as a car. Similarly, the model has also identified a bus in the frame, but with a slightly lower confidence score, indicating a 61.74% accuracy in identifying it as a bus.

Figure 2 represents another frame from the same video. In this frame, there are two objects detected: a person and a motorbike. The object detection model has identified a person in the frame, and it has assigned a high confidence score to this classification, indicating a 94% accuracy in identifying it as a person. Similarly, the model has also identified a motorbike in the frame, but with a somewhat lower confidence score, indicating a 64% accuracy in identifying it as a motorbike.



Figure 6: Frame with Bus and car vehicle classification.

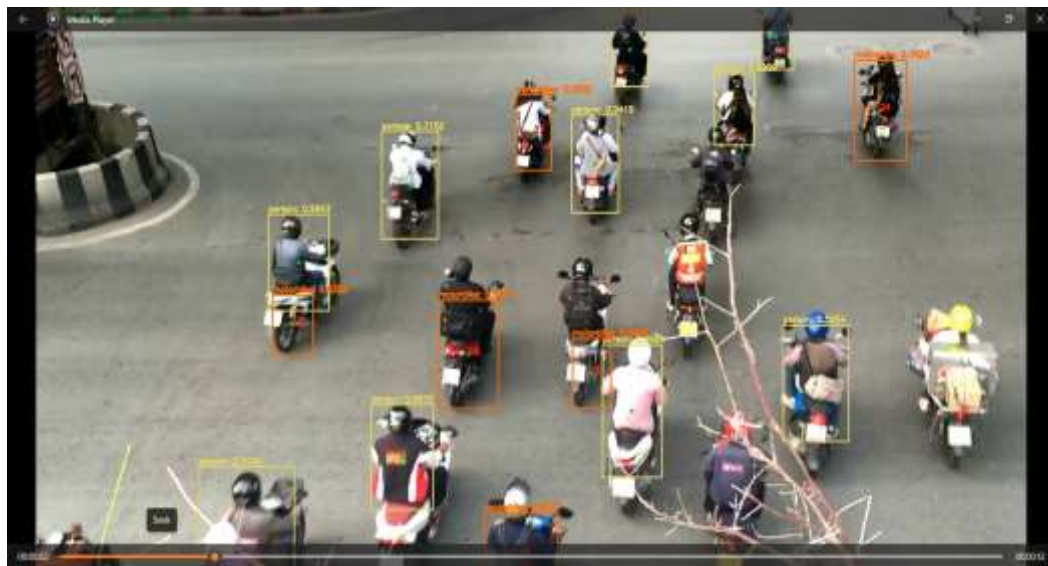


Figure 7: video frame with classification of both person and motor bike.



Figure 8: video frame with classification of persons, cars & buses with an accuracy of 94% ,96% & 99%

Figure 8 represents yet another frame from the same video. In this frame, there are multiple objects detected, including persons, cars, and buses. The object detection model has identified persons in the frame with a high confidence score, indicating a 94% accuracy in identifying them as persons. The model has also detected cars in the frame, and it is highly confident in this classification, indicating a 96% accuracy in identifying them as cars. Furthermore, the model has detected buses in the frame with an extremely high confidence score, indicating a 99% accuracy in identifying them as buses.

5. CONCLUSION

The proposed solution is implemented on python, using the OpenCV bindings. The traffic camera footages from variety of sources are in implementation. A simple interface is developed for the user to select the region of interest to be analyzed and then image processing techniques are applied to

calculate vehicle count and classified the vehicles using machine learning algorithms. From experiments it is apparent that CC method outperforms than BoF and SVM method in all results and gives more close classification results to the ground truth values.

REFERENCES

- [1] Alpatov, Boris & Babayan, Pavel & Ershov, Maksim. (2018). Vehicle detection and counting system for real-time traffic surveillance. 1-4. 10.1109/MECO.2018.8406017.
- [2] Song, H., Liang, H., Li, H. et al. Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Transp. Res. Rev.* 11, 51 (2019). <https://doi.org/10.1186/s12544-019-0390-4>.
- [3] Neupane, Bipul et al. "Real-Time Vehicle Classification and Tracking Using a Transfer Learning-Improved Deep Learning Network." *Sensors (Basel, Switzerland)* vol. 22,10 3813. 18 May. 2022, doi:10.3390/s22103813.
- [4] C. J Lin, Shiou-Yun Jeng, Hong-Wei Lioa, "A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO", *Mathematical Problems in Engineering*, vol. 2021, Article ID 1577614, 10 pages, 2021. <https://doi.org/10.1155/2021/1577614>.
- [5] M. S. Chauhan, A. Singh, M. Khemka, A. Prateek, and Rijurekha Sen. 2019. Embedded CNN based vehicle classification and counting in non-laned road traffic. In *Proceedings of the Tenth International Conference on Information and Communication Technologies and Development (ICTD '19)*. Association for Computing Machinery, New York, NY, USA, Article 5, 1–11. <https://doi.org/10.1145/3287098.3287118>.
- [6] A. Arinaldi, J. A. Pradana, A. A. Gurusanga, "Detection and classification of vehicles for traffic video analytics", *Procedia Computer Science*, Volume 144, 2018, Pages 259-268, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2018.10.527>.
- [7] Gomaa, A., Minematsu, T., Abdelwahab, M.M. et al. Faster CNN-based vehicle detection and counting strategy for fixed camera scenes. *Multimed Tools Appl* 81, 25443–25471 (2022). <https://doi.org/10.1007/s11042-022-12370-9>.
- [8] G. Oltean, C. Florea, R. Orghidan and V. Oltean, "Towards Real Time Vehicle Counting using YOLO-Tiny and Fast Motion Estimation," 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), 2019, pp. 240-243, doi: 10.1109/SIITME47687.2019.8990708.
- [9] L. C. Pico and D. S. Benítez, "A Low-Cost Real-Time Embedded Vehicle Counting and Classification System for Traffic Management Applications," 2018 IEEE Colombian Conference on Communications and Computing (COLCOM), 2018, pp. 1-6, doi: 10.1109/ColComCon.2018.8466734.
- [10] D. E. V. Tituana, S. G. Yoo and R. O. Andrade, "Vehicle Counting using Computer Vision: A Survey," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), 2022, pp. 1-8, doi: 10.1109/I2CT54291.2022.9824432.
- [11] A. Khan, A., Sabeenian, R.S., Janani, A.S., Akash, P. (2022). Vehicle Classification and Counting from Surveillance Camera Using Computer Vision. In: Suma, V., Baig, Z., K. Shanmugam, S., Lorenz, P. (eds) *Inventive Systems and Control. Lecture Notes in Networks and Systems*, vol 436. Springer, Singapore. https://doi.org/10.1007/978-981-19-1012-8_31.
- [12] W. Balid, H. Tafish and H. H. Refai, "Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance," in *IEEE Transactions on Intelligent*

Transportation Systems, vol. 19, no. 6, pp. 1784-1794, June 2018, doi: 10.1109/TITS.2017.2741507.

- [13] N. Jahan, S. Islam and M. F. A. Foysal, "Real-Time Vehicle Classification Using CNN," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225623.
- [14] M. A. Butt, A. M. Khattak, S. Shafique, B. Hayat, S. Abid, Ki-Il Kim, M. W. Ayub, A. Sajid, A. Adnan, "Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems", Complexity, vol. 2021, Article ID 6644861, 11 pages, 2021. <https://doi.org/10.1155/2021/6644861>.
- [15] P. Gonzalez, Raul & Nuño-Maganda, Marco Aurelio. (2014). Computer vision based real-time vehicle tracking and classification system. Midwest Symposium on Circuits and Systems. 679-682. 10.1109/MWSCAS.2014.6908506.