iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.42088

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Object Detection using YOLO

Durriya Bandukwala[1], Muskan Momin[3], Akmal Khan[2], Aasim Khan[5], Dr. Lutful Islam[4]
[1, 2, 3, 4, 5]*Dept. of Computer Engineering, M.H. Saboo Siddik College of Engineering,* Mumbai, India

Abstract: A technical assessment is always conducted prior to the installation of traffic signs at crosswalks, safety lanes, or even expanding areas.
In these cases, using cameras to gather local photos and then analysing them is a valuable method. This paper describes a technique for detecting and tracking four types of vehicles: automobiles, buses, trucks, and motorcyclists, by analysing video footage of road crossings. The first findings of vehicle route accounting reveal that the technique is highly promising, with good outcomes in a number of scenarios, but much more study is needed to make these systems resistant against occlusions and other unforeseen events.
Keywords: Deep Learning, Tracking, Multiple Objects Tracking, CNNs, YOLO, DeepSORT.

I. INTRODUCTION

The management of public policy in metropolitan areas relies heavily on vehicle traffic monitoring. Vehicle flow data assists administrators in determining where resources should be allocated for the construction of road signs, traffic signals, road expansion, and roadside spaces, among other things. These data may also be used to improve traffic light synchronisation and develop methods to reduce urban traffic congestion.

To recognise, categorise, and track cars over several frames, such applications require the use of image processing algorithms. Classic image processing techniques such as particle filters [1], morphological operations [2], and optical flow [3] have already been used to accomplish these goals.

However, significant progress was only made following the widespread adoption of algorithms like Deep Neural Networks (DNN), which demonstrated the capacity to extract complicated and abstract properties from data and express them as a series of quantitative outputs.

Convolutional neural networks (CNNs) are now the state-of-the-art in spatial pattern extraction, and are used in tasks such as picture categorization [4]–[6]. CNN architectures [7] have also been employed as backbones for object detection algorithms [8], [9], and they continue to evolve at a rapid pace, improving detection accuracy and processing times [10]–[12].

Object recognition on video should not only provide an accurate characterization of the target (typically identified by bounding boxes), but it should also do it quickly.

When staring at dynamic items entering and departing the line of sight, the difficulty of comprehending a scene increases dramatically, even for the human brain. In these circumstances, we must categorise, memorise, and monitor moving objects depending on the required output.

Multiple Object Tracking (MOT) is a classification of algorithms used in a computer vision task to evaluate films in order to detect and track objects belonging to several classes [13]. They have applications in everything from video surveillance to self-driving vehicles, recognition to crowd behaviour monitoring, and so on.

This paper suggests employing a flexible computer vision system for recognising and tracking objects to automate a difficult activity that is done in a variety of contexts while using minimum resources.In the instance of the prototype exhibited, the only resources available were a single camera and a single picture storage system. The suggested technology, once approved, might become an IoT node, allowing it to be integrated into smart cities. We offer a method for counting, categorising, and tracking vehicle paths in road crossings in particular.

An administrative agent requests that the photographs be captured by a camera that is installed for a few days in a specified location. Traffic information such as the kind, number, and directions of vehicles flowing, entering, and exiting road junctions are among the expected outcomes.

The YOLOv4 framework is used to create the suggested object detection system, which is suited to four types of vehicles: cars, trucks, buses, and motorcyclists. The DeepSORT (DS) MOT was used to implement the tracking of these items. Finally, an interface was created to define the entry and exit points as well as to check for discrepancies in the findings.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

II. LITERATURE SURVEY

Traditional object detection algorithms have three phases in their pipeline: informative region selection, feature extraction, and classification.

A. Informative region selection.

Because distinct items may show in different parts of the image and have varying aspect ratios or sizes, scanning the entire image with a multi-scale sliding window is a reasonable choice. Despite the fact that this thorough technique can locate all feasible placements for the items, it has apparent flaws. It is computationally intensive and creates an excessive number of redundant windows due to the enormous number of candidate windows. If just a limited number of sliding window templates are used, however, undesirable areas may result.

B. Feature extraction

We need to extract visual characteristics that can offer a meaningful and robust representation in order to distinguish diverse things. The representative characteristics include SIFT, HOG, and Haar-like. This is because these characteristics can result in representations linked with complicated cells in the human brain. However, because of the wide range of looks, lighting situations, and backdrops, manually designing a comprehensive feature descriptor to accurately characterise all types of objects is challenging.

C. Classification

A classifier is also required to separate a target item from all other categories and to make visual recognition representations more hierarchical, semantic, and informative. Supported Vector Machine (SVM), AdaBoost, and Deformable Part-based Model (DPM) are also popular options. Among these classifiers, the DPM is a versatile model that handles severe deformations by mixing object components with deformation cost. Carefully defined low-level characteristics and kinematically inspired component decompositions are merged in DPM using a graphical model. Furthermore, discriminative learning of graphical models enables the creation of high-precision part-based models for a wide range of object classes.

In the PASCAL VOC object identification competition, state-of-the-art results were attained using these discriminant local feature descriptors and shallow learnable architectures, and real-time embedded systems with low hardware load were produced. During the year 2010-2012, however, small increases were achieved by just developing ensemble systems and using minor variations of effective strategies. This is because of the following factors: 1) Using a sliding window technique to generate candidate bounding boxes is redundant, wasteful, and incorrect. 2) Combining manually created low-level descriptors with discriminatively trained shallow models will not close the semantic gap.

III. GENERIC OBJECT DETECTION

Generic object detection helps to locate and categorise existent items in any image, and to label each using rectangular shape boxes to show the degree of certainty that they exist. The frameworks of generic object identification algorithms can be divided into two categories.

One can use the typical object detection pipeline, which involves first producing region proposals and then dividing each proposal into multiple item categories. The other approaches object detection as a regression or classification problem, employing a unified framework to obtain direct results (categories and locations). R-CNN, SPP-net, Fast R-CNN, Faster R-CNN, R-FCN, FPN, and Mask R-CNN are some of the region proposal based approaches, some of which have been associated with one another (e.g. SPP-net modifies R-CNN with a SPP layer). The anchors proposed in Faster R- CNN help to bridge the gap between these two pipelines.

IV. PROPOSED SYSTEM

A. YOLO v4

YOLO: Redmon et al. [17] introduced a unique framework called YOLO, which uses the entire topmost feature map to forecast both confidences and bounding boxes for multiple categories. The state-of-the-art object detection technique is You Only Look Once (YOLO). Unlike standard object detection algorithms, YOLO just looks at a picture once to see whether it contains any objects. YOLOv4 is the most recent and sophisticated iteration of YOLO, out of all the previous versions (Bochkovskiy et al. 2020). It has the quickest operating speed, making it ideal for usage in production systems and parallel computing optimization. [15] Weighted-Residual-Connections[16] Cross-Stage-Partial-Connections, [17] Cross iteration batch normalisation, [18] Self-adversarial-training, [19]

Mish-activation, and so on are some of the new techniques used in YOLO. YOLOv4 employs a Dense Block, a deeper and more complicated network, to get greater accuracy results. Similarly, the feature extractor's backbone is CSPDarknet-53, which leverages CSP connections in addition to Darkenet-53 from the previous YOLOv3. YOLOv4's design includes SPP extra module, PANet path-aggregation neck, and YOLOv3 anchor-based head, in addition to CSPDarknet-53. The SPP block is piled on top of CSPDarknet53 to extend the receptive field that can discretize the most notable context elements while also ensuring that the network's throughput does not suffer.Similarly, PANet is utilised instead of the Feature Pyramid Network (FPN) in YOLOv3 for parameter aggregation from many tiers of backbone. The training time for YOLOv4 models was about 24 hours, and they shared the same hardware resources as CenterNet. The optimizer weight decay is 0.0005 and the batch size is 64. Throughout the training phase, the initial learning rate of 0.01 is used, and the momentum is set at 0.9. CenterNet took around 340 milliseconds per picture to infer.

B. DeepSort

By integrating appearance information with its tracking components, the Simple Online and Realtime Tracking with a Deep Association metric (Deep SORT) allows for multiple object tracking. Tracking is accomplished using a Kalman Filter and a Hungarian algorithm. In this case, Kalman filtering is done in picture space, and the Hungarian technique uses an association metric to generate bounding box overlap to simplify frame-by-frame data association. A trained convolutional neural network (CNN) is used to extract motion and appearance information. The tracker improves better robustness against object misses and occlusions by including CNN, while maintaining its flexibility to swiftly apply to online and realtime settings. In Table 1, the system's CNN architecture is depicted. With two convolutional layers, a large residual network is created.

A broad residual network is used, which consists of two convolutional layers followed by six residual blocks. A global feature map with dimensions 128 is calculated in dense layer 10. Finally, the unit hypersphere's batch and l2 normalisation characteristics enable compatibility with the cosine arrival metric. As evidenced by the MOT16 benchmark for Multi-object Tracking, Deep SORT is a very adaptable tracker that can match performance capabilities with other state-of-the-art tracking algorithms.

Name	Patch size/stride	Output size
Conv1	3×3/1	32×128×64
Conv2	3×3/1	32×128×64
Max Pool 3	3×3/2	32×64×32
Residual 4	3×3/1	32×64×32
Residual 5	3×3/1	32×64×32
Residual 6	3×3/2	64×32×16
Residual 7	3×3/1	64×32×16
Residual 8	3×3/2	128×16×8
Residual 9	3×3/1	128×16×8
Dense 10		128
Batch and ℓ2 normalisation		128

Table 1: Overview of Deep SORT's CNN Architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

V. METHODOLOGY

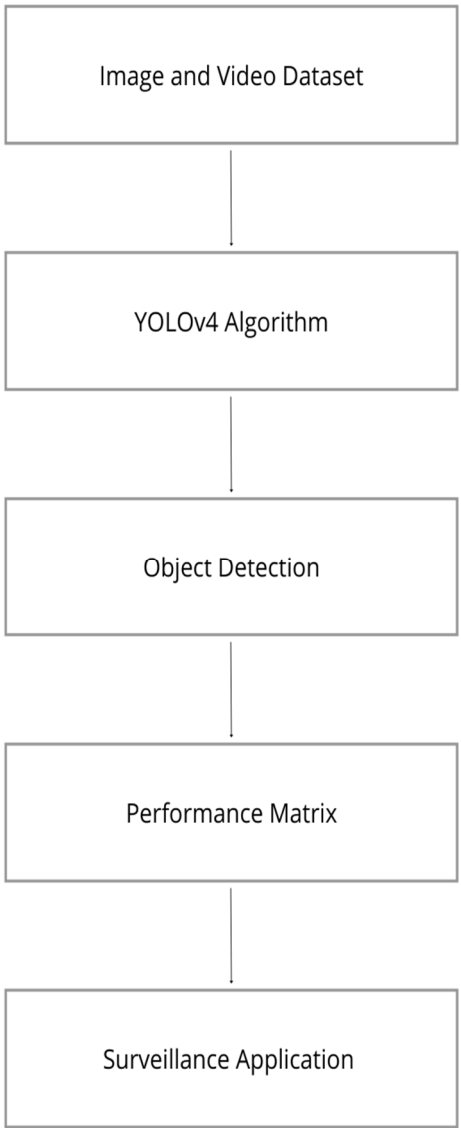


Fig 1: Methodology of implementation

Methodology of implementation:

1) *Data Collection:* Vehicle detection dataset for urban roads was used as the data source. It's crucial since the quality and quantity of data will have a direct impact on how well the trained model performs under different environmental conditions.
2) *Data Preparation:* Data is prepared so that it can be used in a deep learning model.
3) *Network Architectures:* Selecting a suitable model for training from a large number of models with varying properties.
4) *Network Parameters:* Weights and bias of deep learning models are randomly initialised to avoid symmetry, which could adversely impair the training process.
5) *Hyper Parameters:* Before training, hyper parameters such as learning rate, regularisation parameter, and dropout probability are all initialised.
6) *Training Deep Learning Model:* A GPU is used to train a deep learning model on an image and video dataset.
7) *Analysis and Evaluation:* Based on the acquired results, the network model is evaluated. To optimise the overall results, more hyper settings are tweaked. Darknet-53 is used to extract features, and the YOLOv4 algorithm is used to detect vehicles.

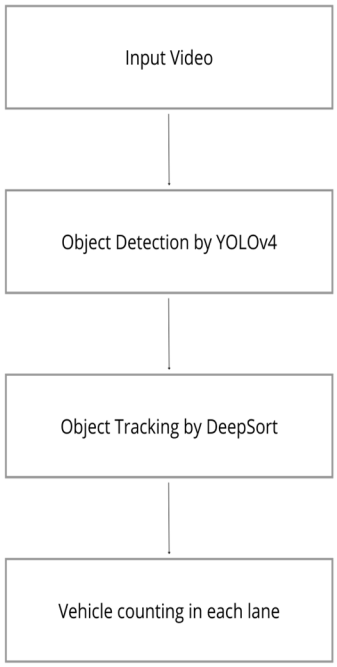


Fig 2: Flowchart of Vehicle Tracking

VI. RESULT

This section compares the performance of various object detector and tracker combinations. The primary purpose of this research is to find the optimal item detector-tracker combo. The models are evaluated on a total of 546 video clips with a length of 1 minute each, totaling almost 9 hours of video time. Figure 1 depicts all camera views with manually created green and blue polygons that count the number of cars going through in both north and southbound directions.

The vehicle counts are categorised into four categories: [15] overall vehicle counts, [16] total automobile counts, [17]total truck counts, and [18] overall vehicle counts during different periods of the day (i.e., daylight, nighttime, rain). All of the cars are manually tallied using the existing 9-hour video test data to confirm ground truth. The accuracy of the system is measured by comparing the automated counts derived from various model combinations to the ground truth value given in per hundredths or percentages.

Heat maps depicting False Negatives (FN), False Positives (FP), and True Positives (TP) for all of the object detectors utilised in the study are presented in Fig. 3 to analyse their performance. The models were put to the test on six distinct camera perspectives at various times of the day. The heat maps for CenterNet, YOLOv4, and Efficient Set are shown in the left, centre, and right columns, respectively. The first column signifies FN, the second column denotes FP, and the third column denotes TP for each of these object detectors. If the detector fails to identify the vehicle despite it being present at that location, the detection is labelled as False Negative (FN).As a result, the column depicting FN should not have increased colour intensity along such areas of the road. Except for CenterNet's 5th camera view, where it makes heat maps in its south boundaries as well, almost all object detectors have worked well at detecting FN in most camera views. This is primarily due to the fact that certain camera viewpoints had an insufficient amount of traffic photographs for training, and such sites may have had significant congestion. Heat maps closer to the camera in night views, for example, are usually formed when big gross vehicles such as buses and trucks stay jammed for an extended period of time. Similarly, if the detector identifies a vehicle when there are none present, the detection is labelled as False Positive (FP). The FP columns for object detectors are typically clean, as seen in the heat maps, with a few camera views in Efficient Set causing inaccurate classifications. The algorithm misclassified several of the detections due to the camera view with flyovers or overpass roads. Misclassifications can also be caused by camera motions and situations such as rain stuck on the lens or complete darkness. We don't want to observe severe heat maps for both false negatives and false positives in the ideal world.If, on the other hand, we have more false positives but fewer false negatives, the model may have been overconfident, which isn't desirable. Finally, True Positive (TP) identifies automobiles accurately when there are real vehicles on the road. Except for a few camera views where the cars were either too far away or encountered lowlight or nighttime situations, when only the vehicles' headlights were visible, most object detection models produced correct true positives.

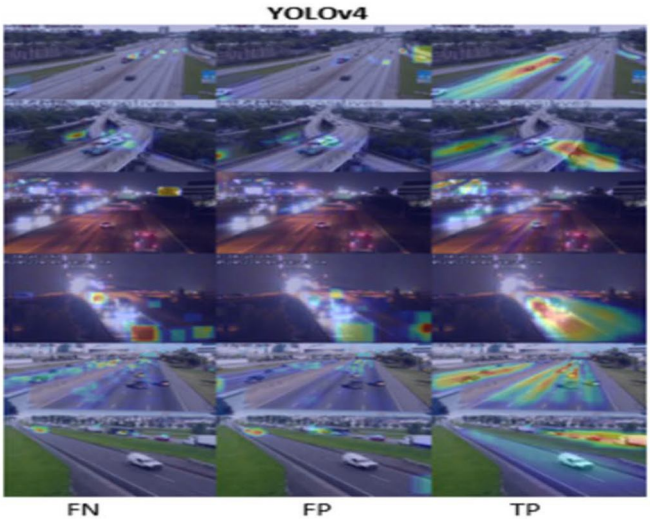


Fig 3: Heat maps generated for YOLOv4 object detector

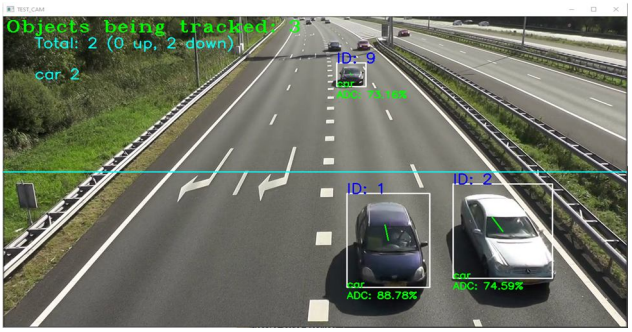


Fig 4. Result of the Experiment

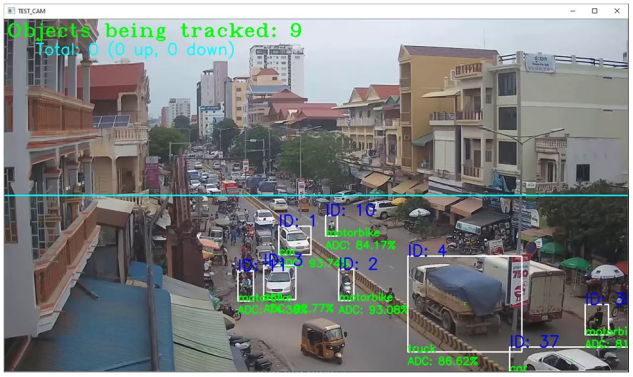


Fig 5. Result of the Experiment

VII. FUTURE SCOPE

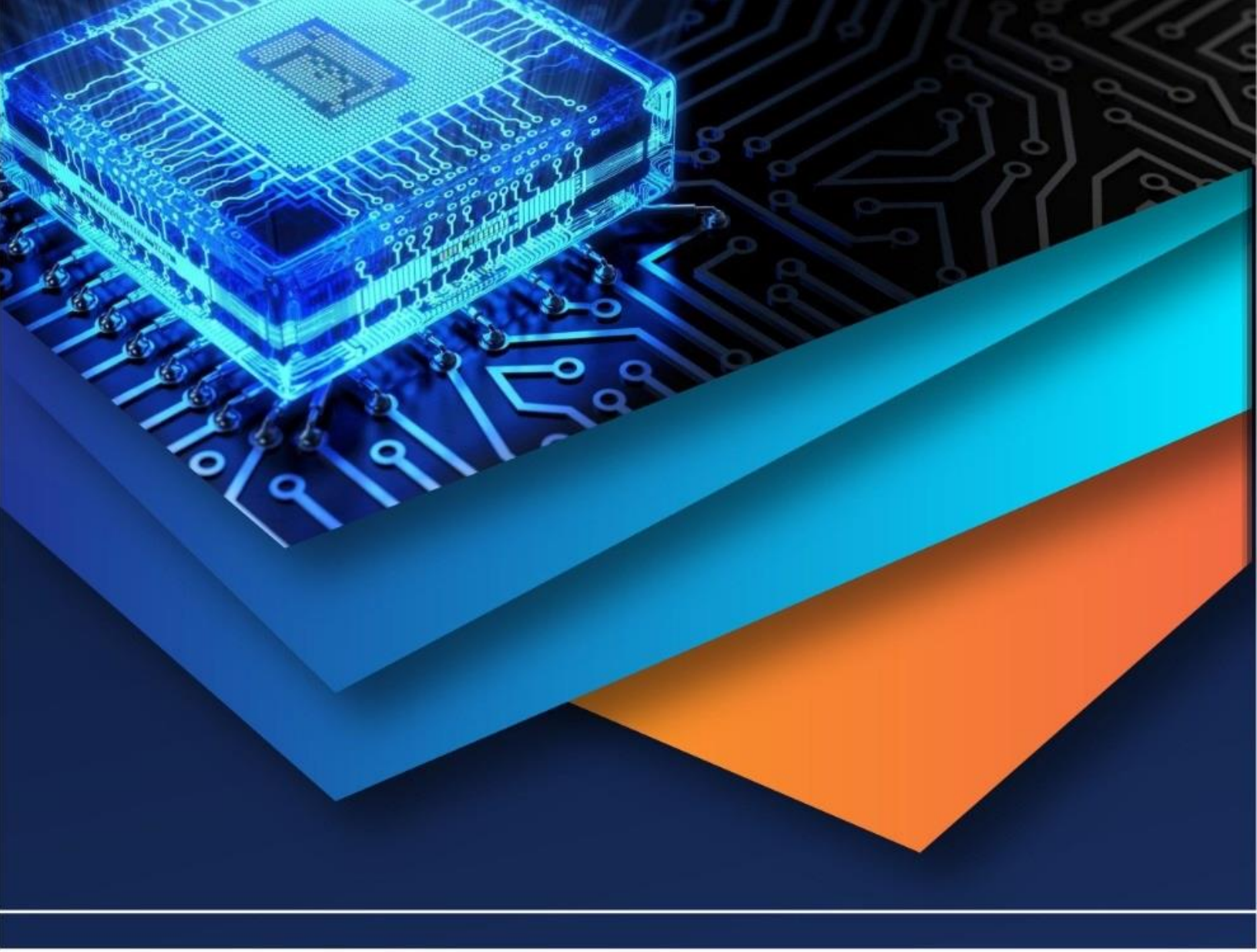
Multiple object detection in video surveillance is a difficult task that is influenced by the density of items in the monitoring area or on the road, as well as timings. The multiple object detection technique implemented in this work is useful for traffic and various surveillance applications. For most computer and AI (Artificial Intelligence) vision systems, multiple object detection is a critical capability. The obtained data is analysed and tabulated. The dataset is made up of (images and videos) with different levels of light. The system efficiently recognises several objects with an accuracy of 98 percent for image datasets and 99 percent for video datasets (roughly average of all frames), according to the obtained findings. Furthermore, the suggested YOLO model versions may be implemented on DSP and FPGA platforms. [15] [16]

VIII. CONCLUSION

A detection-tracking system is used in this study to automatically count the number of cars on the road. The authors have updated the state-of-the-art detector-tracker model combinations to obtain considerable improvements in vehicle counting findings, while there are still several flaws that they want to solve in a future research. Occlusion and reduced visibility caused identity switches, and the same vehicles were detected multiple times. While camera quality, occlusion, and low light conditions made it difficult to accurately detect different classes of vehicles, certain detector-tracker framework combinations worked well in challenging conditions. Real-time object detections might be combined with monitoring vehicle movement trajectories using deep learning-based object identification models combined with both online and offline multi-object tracking systems. This plan was approved, allowing for more precise car counts. Furthermore, we tested the detector-ability tracker's to correctly detect different classes of vehicles, estimate vehicle speed, direction, and trajectory information, and identified some of the best-performing models that could be fine-tuned to remain robust at counting vehicles in various directions and environmental conditions. The figures and tables presented a systematic depiction of which model combinations work effectively when it comes to acquiring vehicle counts in various scenarios. Overall, the experimental findings show that YOLOv4 and Deep SORT, as well as CenterNet and Deep SORT, are the best combinations for counting all cars on the road.

REFERENCES

[1] Bouvi´e, J. Scharcanski, P. Barcellos, and F. L. Escouto, “Tracking and counting vehicles in traffic video sequences using particle filtering.” IEEE, 2013, pp. 812–815.
[2] H. T. P. Ranga, M. R. kiran, S. R. shekar, and S. K. N. kumar, “Vehicle detection and classification based on morphological technique,” in International Conference on Signal and Image Processing, 2010.
[3] A. Abdagic, O. Tanovic, A. Aksamovic, and S. Huseinbegovic, “Counting traffic using optical flow algorithm on video footage of a complex crossroad,” in Proceedings ELMAR-2010, Sep. 2010, pp. 41–45.
[4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, vol. abs/1409.1556, 2015.
[5] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.
[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[7] Machine Learning: Methods and Applications to Brain Disorders. ACADEMIC PR INC, 2019. [Online]. Avail- able: https://www.ebook.de/de/product/36978686/machine learning methods and applications to brain disorders.html
[8] J. Redmon and A. Farhadi, “Yolov3: An incremental improve- ment,” 2018.
[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks.”
[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,”IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.
[11] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525.
[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.
[13] G.Ciaparrone,F.LuqueS´anchez,S.Tabik,L.Troiano,R.Tagli- aferri, and F. Herrera, “Deep learning in video multi-object tracking: A survey,” Neurocomputing, vol. 381, pp. 61–88, 2020.
[14] real-time object detection with region proposal networks,”IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.
[15] Alp Güler R, Neverova N, Kokkinos I (2018) Densepose: Dense human pose estimation in the wild. In: Proceedings of the IEEE Confer- ence on Computer Vision and Pattern Recognition, pp 7297–7306
[16] Arteta C, Lempitsky V, Zisserman A (2016) Counting in the wild. In: European conference on computer vision, Springer, pp 483–498 Asha C, Narasimha Than A (2018) Vehicle counting for traffic manage-
[17] Ment system using YOLO and correlation filter. In: 2018 IEEE International Conference on Electronics, Computing and Com- munication Technologies (CONNECT), IEEE, pp 1–6
[18] Awang S, Azmi NMAN (2018) Vehicle counting system based on vehi- cle type classification using deep learning method. In: IT Conver- gence and Security 2017. Springer, pp 52–59
[19] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B (2016) Simple online and realtime tracking. In: 2016 IEEE International



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY