Real-Time Identification of Traffic Actors using YOLOv7

[1] Pavan Kumar Polagani, [2] Lakshmi Priyanka Siddi, [3] Vani Pujitha M

[1,2,3] Velagapudi Ramakrishna Siddhartha Engineering College,Vijayawada, India

***Abstract*—Real-time traffic object detection is a key topic in computer vision, especially for improving traffic safety and management. This research describes a novel strategy for detecting traffic actors in real time using YOLOv7, a cutting-edge deep learning system. Traditional computer vision algorithms, such as SSD, R-CNN, and older versions of YOLO, frequently exhibit slow response times and poor accuracy in high-traffic areas.YOLOv7, an advanced object detection method based on convolutional neural networks (CNNs), is used in the proposed approach to address these difficulties straight on. YOLOv7 not only achieves real-time object detection, but also greatly increases accuracy by removing superfluous candidate boxes and employing a non-maximum suppression module to choose the best bounding boxes from overlapping ones. Furthermore, the spatial pyramid pooling block improves accuracy by enhancing the network's receptive field without introducing additional parameters. In this research, we show how our proposed model performs in a variety of driving circumstances, including clear and overcast skies, variable lighting conditions, occlusions, and the presence of noise in input data. This model detects traffic participants such as automobiles, pedestrians, cyclists, and traffic signs, which contributes to improved traffic safety and management.**

***Index Terms*—YOLOv7, Traffic Detection, Convolutional Neural Networks**

I. INTRODUCTION

Real-time object detection in traffic poses a formidable challenge within the realm of computer vision. This task involves processing live video streams captured by cameras or sensors deployed in traffic environments. The objective is to identify and precisely locate a diverse range of objects, including vehicles, pedestrians, bicycles, and traffic signs. The primary goal of real-time object detection in traffic is to enhance traffic safety by providing real-time information to drivers, pedestrians, and traffic management systems.

This task is complex due to factors such as the diverse appearance and motion of objects, changing lighting conditions, occlusions, and noise in the input data. The field of computer vision has developed numerous algorithms and techniques to address these challenges, including deep learning-based approaches, feature-based methods, and fusion-based strategies.

One widely adopted algorithm for real-time object detection is YOLO (You Only Look Once). YOLO partitions the input image into a grid of cells and estimates the likelihood of object presence within each cell. Notably, YOLO's swiftness is a key advantage, enabling real-time object detection with a single forward pass through the neural network. This efficiency makes it particularly suitable for applications like autonomous driving and surveillance, where rapid and precise object detection is essential.

The manuscript follows a structured format comprising seven sections. Section 2 provides an extensive literature review on image classification and object detection. Section 3 details the dataset used in this study, while Section 4 offers an overview of the architectural framework utilized. Section 5 describes the methodology employed in our research. Section 6 describes the Experimental Setup. The experimental results are presented in Section 7, accompanied by a thorough analysis of the model's performance and a discussion of the findings. Finally, Section 8 concludes the paper by summarizing key insights and suggesting potential avenues for future research.

A. MAIN CONTRIBUTION OF THE WORK

1) The primary objective of this study is to develop a computer vision system capable of accurately detecting and tracking vehicles, pedestrians, cyclists, and other traffic objects in real-time.
2) The research aims to enhance traffic safety and efficiency by utilizing real-time traffic flow information to dynamically adjust traffic signals, provide real-time information to drivers, and alert emergency services to potential accidents.
3) The study emphasizes the detection of smaller objects within the camera frame, further advancing the capabilities of the system.

II. RELATED WORK

In this section, we delve into various methodologies employed for real-time object detection in traffic, each offering distinct strengths and weaknesses, thereby enriching the diverse landscape of solutions within this domain.

The method described in Zarei, Moallem, and Shams 2022 relies around the use of the Fast-Yolo-Rec method, which expertly balances accuracy and speed. Its key goals are trajectory classification via LSTM-based recurrent networks

and position prediction via SSAM-YOLO and LSSN. The optical flow-based detection method is critical in establishing the direction and speed of individual pixels inside a picture. An interesting method is used to speed up processing: odd frames of input images are designated for detection, while even frames are committed to prediction, considerably increasing overall speed.

Advantage: Fast-Yolo-Rec excels in rapid and cost-effective vehicle detection.

Disadvantage: However, it demands substantial computational resources for handling real-time data.

Methodology Ye et al. 2022 introduces the SEF-Net framework, which is made up of three modules: Stable Bottom Feature Extraction (SBM), Lightweight Feature Extraction (LFM), and Enhanced Adaptive Feature Fusion Module (EAM). SBM improves precision in tiny object detection by expanding convolutional channels, which is especially beneficial for small objects. Furthermore, Attention Enhancement blocks encode geographic and channel-specific semantic information, which improves item detection and placement.

Advantage:

1) This approach swiftly identifies car locations at a lower computational cost compared to other high-speed detectors without necessitating additional processing.

2) SBM significantly enhances precision for small object detection, outperforming YOLOv4 in multi-detection capability.

Disadvantage: Handling and analyzing large volumes of realtime data demand substantial computational resources

In methodology Guney, Bayilmis¸, and C¸ akan 2022 a technical framework based on the YOLOV4 concept is introduced. This framework focuses on a variety of topics, such as risk assessment, object detection, and intent recognition. Notably, the system uses Part Affinity Fields to add human skeletal traits, resulting in enhanced intention recognition. It also uses LSTM and CNN to assess vehicle heading, while EfficientNet is utilised to estimate potentially harmful cars. Furthermore, to improve risk assessment capabilities, the framework employs saliency maps generated by the RISE algorithm and Explainable AI technology.

Advantage: Enhanced intention recognition through human skeletal characteristics.

Disadvantage: Increased computational complexity and longer training times due to multiple model usage.

DFF-Net, which was introduced in methodology Li et al. 2020, is intended to detect real-world traffic items on railways. It is divided into two parts: previous detection and object detection. To initialize the system and restrict the search space for object detection, the previous detection module employs VGG-16 pretrained on ImageNet. The object detection module seeks to recognize and predict the kinds of objects contained within the prior boxes.

Advantage: DFF-Net excels at increasing detection accuracy and effectively addressing class imbalance in railway object detection.

Disadvantage: However, when compared to YOLO, a onestage object detector, DFF-Net has a slower total speed.

The authors obtained a large dataset spanning numerous traffic incidents such as accidents, congestion, and vehicle breakdowns in methodology Ye et al. 2021, They used a pretrained Mask-SpyNet model for video-based object detection and post-processing to identify and categorise traffic occurrences.

Advantage: This novel approach considerably enhances nighttime traffic event identification, hence improving motorway traffic management safety and efficiency.

Disadvantage: However, there are evaluation constraints, and the method's performance may be altered by changing lighting circumstances.

DLT-Net, the suggested technique in Qian, Dolan, and Yang 2020, is a unified neural network built for self-driving cars. Using common features, it detects drivable zones, lane lines, and traffic objects all at once. For each task, the design incorporates a common encoder and three different decoders. A context tensor is proposed to improve overall performance and computing efficiency by facilitating information sharing among activities. DLT-Net uses the YOLOv3 model for traffic object detection, which is a cutting-edge one-stage object detection approach. Extensive studies on the BDD dataset show that DLT-Net outperforms traditional approaches in these key perception tasks.

Advantage: It's unified design improves efficiency and performance in autonomous driving perception by recognizing drivable zones, lane lines, and traffic objects all at the same time.

Disadvantage: Complex scenarios, such as identifying reflected items from traffic signs or dealing with interrupted lane lines, may pose difficulties.

The study Li and al. 2020 describes a comprehensive autonomous driving framework that includes four key tasks: object detection using an optimized YOLOv4 model, intention recognition based on pedestrian skeleton features via Part Affinity Fields and CNN analysis, and CNN-driven risk assessment for dangerous vehicles and traffic light recognition. The YOLOv4 model has been improved to improve detection accuracy, providing a comprehensive approach to ensuring safe autonomous driving.

Advantage: It integrates object detection, intention identification, and risk assessment to improve autonomous driving safety.

Disadvantage: The complexity of the improved PAFs model in the intention recognition component may have an effect on computing efficiency.

III. DATASET

The enormous collection of images in the Traffic Object Dataset was specifically picked for the task of identifying and classifying traffic objects. This dataset consists of 4,591 highquality images that depict various real-world traffic situations which have 38 classes. It is taken from Roboflow. Where 80% is used for training and the remaining 20% is for testing. Here, Fig. 1 represents a sample training image from the dataset.

Figure 1. A sample image from the dataset

IV. ARCHITECTURE

The YOLOV7 architecture mainly consists of three parts, i.e., backbone, neck, and head. The backbone extracts features from the input image, the neck combines features of different resolutions, and the head generates object detection predictions. This modular design enables YOLOv7 to efficiently process input data and accurately detect objects in realtime scenarios. Fig. 2 represents the architectural diagram of YOLOv7.

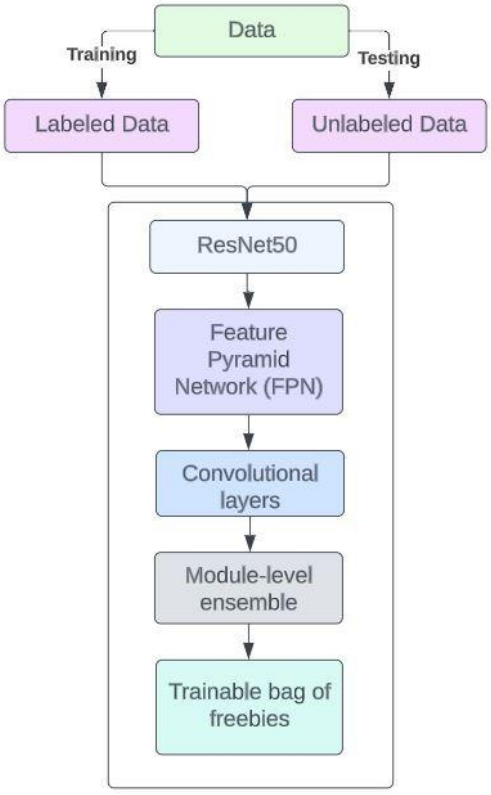

Figure 2. Architecture diagram of YOLOv7

V. PROPOSED METHODOLOGY

Two key elements make up our suggested methodology and architecture for the real-time detection of traffic actors using YOLOv7: extended efficient layer aggregation networks (EELAN) and a compound scaling technique for concatenationbased models.

A. Extended Efficient Layer Aggregation Networks (E-ELAN) Extended Efficient Layer Aggregation Networks, or E-ELAN, are intended to improve network learning while maintaining the integrity of the initial gradient path. Expand, shuffle, and merge cardinality techniques are incorporated into the computing blocks of the design to accomplish this. The expand operation uses group convolution to expand the channel and cardinality of the computing blocks. The network is able to capture a wider variety of features by extending the channels. Parallel processing and the investigation of several feature representations inside each computing block are both made possible concurrently by increasing cardinality.

It makes use of group convolution to keep the original transition layer of the design. By doing this, it is made sure that the patterns of connectedness and information flow between the computing blocks are maintained. In order to provide seamless information transfer while supporting the enlarged channel and cardinality, the transition layer serves as a link between the earlier computational blocks and succeeding layers. It also enables several groups of computational blocks to specialize in learning different characteristics by utilizing these expand and group convolution procedures. The network can capture and distinguish traffic actors with increased accuracy because to the diversity of feature learning. The shuffling process is also very important in E-ELAN. According to a predetermined group parameter, it divides the feature maps produced by the computational blocks into various groups. By successfully mixing and combining the learned features from many blocks, this shuffling method promotes feature diversity and guards against over-reliance on a single set of computational blocks. As a result, the model's ability to generalize and distinguish among traffic actors in real-world circumstances is improved. The E-ELAN process ends with the merge cardinality procedure. The merged feature map with maintained channel numbers is created by joining the shuffled feature maps from several groups.

The merge procedure successfully merges the many features picked up by several computational block groups, utilizing their combined knowledge to increase detection precision. EELAN improves the YOLOv7 architecture overall by allowing ongoing learning of various features without altering the initial gradient path. Different sets of computational blocks can specialize in learning different features thanks to the combination of expand, shuffle, and merge cardinality approaches.

B. ResNet50

ResNet50 is a convolutional neural network (CNN) architecture that is widely used for image classification and object detection. It is known for its ability to train deep networks without overfitting. ResNet50 is used as the backbone network in the YOLOv7 model. The backbone network is responsible for extracting feature maps from the input image. These feature maps are then used by the YOLOv7 head to predict object bounding boxes and class labels. ResNet50 is a good choice for the backbone network in YOLOv7 because it is able to extract a rich set of features from the input image. These

features can then be used by the YOLOv7 head to accurately detect and classify objects. In addition, ResNet50 is a relatively efficient CNN architecture, which means that it can train and run quickly. This is important for YOLOv7, which is designed to be a real-time object detection model.

Here are some of the benefits of using ResNet50 as the backbone network in YOLOv7:

1) Accuracy: ResNet50 has been shown to achieve high accuracy on a variety of image classification and object detection datasets.

2) Efficiency: ResNet50 is a relatively efficient CNN architecture, which means that it can train and run quickly.

3) Transfer learning: ResNet50 is a pre-trained model, which means that it has already been trained on a large dataset of images. This can be beneficial for training YOLOv7 on a smaller dataset of custom images.

Overall, ResNet50 is a good choice for the backbone network in YOLOv7 because it is able to extract a rich set of features from the input image and is relatively efficient.

C. Feature Pyramid Network (FPN) :

The Feature Pyramid Network (FPN) is a key component of the YOLOv7 model. The FPN is responsible for extracting feature maps at multiple scales, which allows the model to detect objects of different sizes. The YOLOv7 FPN uses a top-down architecture with lateral connections. The top-down pathway starts from the highest-resolution feature map and gradually downsizes it while preserving semantic information. The lateral connections combine the downsampled feature maps from the top-down pathway with the corresponding feature maps from the backbone network. This results in a set of feature maps at multiple scales, which are then used by the YOLOv7 head to predict object bounding boxes and class labels.

The FPN has several advantages over traditional single-scale feature extraction approaches. First, it allows the model to detect objects of different sizes by providing feature maps at multiple scales. Second, it helps to improve the accuracy of object detection by combining low-level features, which are rich in spatial information, with high-level features, which are rich in semantic information. Third, the FPN makes the model more robust to occlusion and other image degradations. The YOLOv7 FPN is implemented using a stack of convolutional layers. The first layer in the stack is responsible for downsampling the feature map from the backbone network. The subsequent layers in the stack are responsible for upsampling the feature maps from the previous layers and combining them with the corresponding feature maps from the backbone network. The final layer in the stack produces a set of feature maps at multiple scales, which are then used by the YOLOv7 head to predict object bounding boxes and class labels.

Overall, the Feature Pyramid Network is a key component of the YOLOv7 model that contributes to its high performance on a variety of object detection tasks.

D. Compound Scaling Method for Concatenation-Based Models

In order to modify the YOLOv7 architecture to meet various inference speed requirements, model scaling is a crucial component. Scaling concatenation-based models, however, presents particular difficulties in maintaining the ideal structure while attaining the needed scalability. In light of these difficulties, we provide a compound scaling technique that concurrently takes into account the depth and width factors of processing blocks and transition layers. It becomes especially crucial to preserve the ideal structure when growing concatenation-based models. Performance shouldn't be adversely affected by the architecture's ability to adapt to variations in depth. In order to achieve this, our suggested compound scaling strategy concentrates on maintaining the proportion of input to output channels while scaling.

The depth factor describes how many computing units are stacked inside the design. Scaling the depth factor alters the in-degree and out-degree of each layer by increasing or decreasing the number of computational blocks. The subsequent transition layer's input-to-output channel ratio is impacted by this modification. To avoid hardware consumption distortions and guarantee appropriate model parameter use, the ratio must be maintained. In addition, the width factor, which describes the size of the computational blocks' channel, must be changed in proportion to variations in depth. The ideal structure of the original architecture is maintained by scaling the width factor, which makes sure that the expanded or contracted computational blocks line up with the needs of the altered depth.

The compound scaling method provides a constant ratio between input and output channels all over the architecture by taking into account both the depth and width parameters together. This method enables smooth switching between various scaling factors without impairing the model's functionality. For the model to continue learning and making accurate traffic actor distinctions, the ideal structure must be maintained when scaling. The model's ability to effectively capture and analyze features is maintained by the compound scaling strategy, enabling accurate and reliable detection of traffic actors in real-time circumstances.

The Region Proposal Network (RPN), which generates anchors based on anchor sizes and grades them using RPN classification scores, is part of the detection module of the YOLOv7 architecture. Then, using RPN bounding box regression on the anchors, bounding boxes are generated. The classification layer provides classification scores to the detection layer, which then uses bounding box regression to generate the bounding boxes. RPN loss and detection loss are both included in the loss module. While the detection loss combines classification, regression, and objectness losses for the detection layer, the RPN loss combines classification and regression losses specific to the RPN. These losses use cross-entropy and smooth L1 loss functions and are computed for each image in the batch. This module allows YOLOv7 to detect objects accurately by making effective region proposals, improving predictions, and optimizing the model with the right loss functions.

E. Module-level ensemble (MLE)

MLE is a technique that can be used to improve the performance of object detection models by combining the outputs of multiple modules. MLE is typically implemented by replacing a single module in the model with a set of parallel modules. The outputs of the parallel modules are then combined to produce the final output of the model.

The YOLOv7 model includes a module-level ensemble layer in the neck of the model. The neck of the model is responsible for combining the feature maps from the backbone network and the head of the model. The module-level ensemble layer in YOLOv7 replaces the standard convolutional layer in the neck with a set of parallel convolutional layers. The outputs of the parallel convolutional layers are then combined to produce the final feature maps that are used by the head of the model.

F. Trainable Bag of Freebies

Trainable Bag of Freebies (BoF) is a set of techniques that can be used to improve the performance of object detection models without increasing the training cost. BoF techniques are typically implemented as trainable modules that can be added to existing object detection models. The YOLOv7 model includes several trainable BoF techniques, including:

1) Cross-Module Channel Communication (C3): C3 allows modules to communicate with each other by sharing channel information. This can help to improve the performance of the model by allowing modules to learn from each other.

2) Selective Attention Module (SAM): SAM allows the model to focus on the most important parts of the input image. This can help to improve the accuracy of the model by reducing the amount of noise that is processed.

3) Efficient Channel Attention (ECA): ECA allows the model to learn the importance of different channels in the input image. This can help to improve the efficiency of the model by reducing the number of channels that are processed.

G. Detection module

1) Region Proposal network (RPN):

a. Anchors

$$\text{anchors} = \left(\frac{w_{min}+w_{max}}{2}\right)\text{x}\left(\frac{h_{min}+h_{max}}{2}\right) \quad 1$$

where wmin and wmax represent the minimum and maximum widths of the anchors, and hmin and hmax represent the minimum and maximum heights of the anchors.

b. Anchor Scores

$$\text{scores} = \sigma(rpn_c ls_s core) \quad 2$$

where σ represents the sigmoid function, $\text{rpn}_\text{c}\text{ls}_\text{s}\text{core}$ represents the output of the RPN's classification layer.

c. Bounding boxes

$$b_{boxes} = rpn_b box_p red\ x\ anchors\ +\ anchors \quad 3$$

where $rpn_b box_p red$represents the output of the RPN's bounding box regression layer.

2) Detection layer:

a. Classification scores

$$\text{scores} = \sigma(cls_s core) \quad 4$$

where $\text{cls}_\text{s}\text{core}$is the output of the detection layer's classification layer.

b. Bounding boxes

$$b_{boxes} = rpn_b box_p red\ x\ anchors\ +\ anchors \quad 5$$

where $rpn_b box_p red$represents the output of the RPN's bounding box regression layer.

H. Loss module

1) RPN loss:

$$L_{prn} = \frac{1}{N}\sum_{i=1}^{N}\left(L_{cls}(scores_i) + L_{reg}(bboxes_i)\right) \quad 6$$

where Lcls is the cross-entropy loss for the classification scores and Lreg is the smooth L1 loss for the bounding boxes.

2) Detection Loss:

$$L_{det} = \frac{1}{N}\sum_{i=1}^{N}\left(L_{cls}(scores_i) + L_{reg}(bboxes_i) + L_{obj}(obj_i)\right) \quad 7$$

where Lobj is the binary cross-entropy loss for the objectness scores.

VI. EXPERIMENTAL SETUP

The experimental setup for training YOLOv7 on the Traffic Object Dataset included 4591 photos representing 38 distinct object classes, which were divided into training and testing subsets with an 80/20 split. Google Colab's GPU support was used to accelerate model convergence. To allow the model to learn detailed traffic object attributes, training parameters comprised a batch size of 8, an initial learning rate of 0.001, and 50 training epochs. The dataset was meticulously divided into training, validation, and test sets, and each image was tagged with bounding boxes that defined item placements. To balance computational efficiency and detection precision, the YOLOv7 model was built to recognize all 38 different object classes using an input image size of 640 pixels. The PyTorch framework was used for training, and the model was evaluated on both a validation set for assessing generalization during training and a specialized test set for testing realworld detection accuracy. This configuration allowed for a thorough test of YOLOv7's performance in real-time traffic object detection.

VII. EXPERIMENTAL RESULTS

The figure 3 shows the object detection results of a YOLOv7 model trained to detect five different types of objects: car, person, bicycle, motorcycle, and bus. The graphs are divided into two sections: the first section shows the results of the validation set, and the second section shows the results of the test set. The Graph metrics are as follows:

1) Box: The average precision (AP) for the bounding boxes drawn around the detected objects.

2) Objectness: The AP for the objectness score, which is a measure of how confident the model is that an object is present in a given bounding box.

3) Classification: The AP for the classification score, which is a measure of how confident the model is that a detected object is of the correct type.

4) MAP@0.5: The mean average precision (mAP) for the first 50% of the detection curve, where the detection curve is a recall plot versus the precision at different IoU (intersection over union) thresholds.

5) MAP@0.5:0.95: The mAP for the detection curve between IoU thresholds of 0.5 and 0.95.

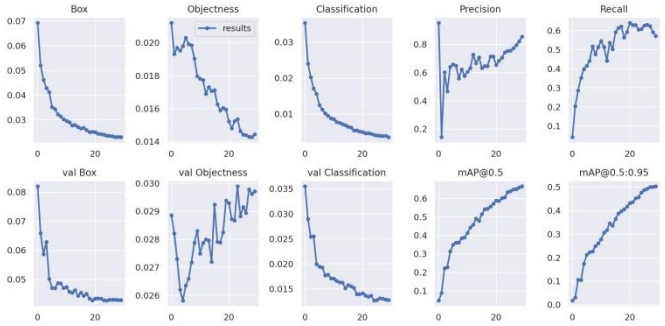

Figure 3. Different results curves

The higher the AP or mAP, the better the model is performing. Overall, the results show that the YOLOv7 model is performing well on both the validation and test sets, with AP and mAP scores above 0.5 for all five types of objects. However, there is some variation in performance across the different metrics and object types. For example, the model is better at detecting cars and people than bicycles and motorcycles.

In the figure 4 x-axis shows the recall, which is the fraction of all relevant instances that are retrieved by the model. The y-axis shows the precision, which is the fraction of retrieved instances that are relevant.

The blue line on the graph shows the precision-recall curve for the model. The white line shows the perfect precision-recall curve, which is a line where the precision is always equal to 1. The closer the blue line is to the white line, the better the model is performing.

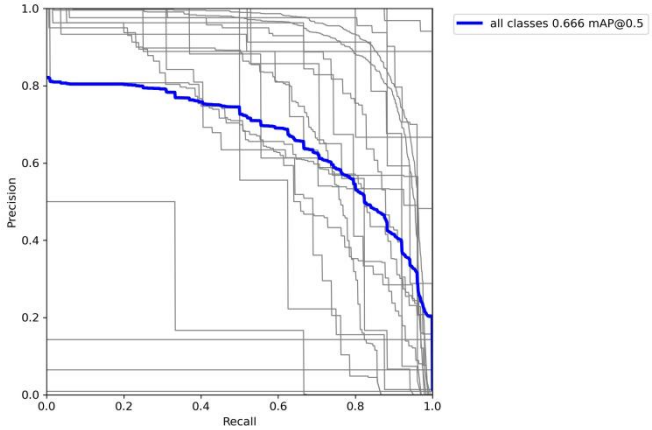

Figure 4. Precision-Recall curve

The mAP@0.5 at the top of the graph is the mean average precision at an intersection over union (IoU) threshold of 0.5. This is a common metric for evaluating object detection models. A higher mAP@0.5 indicates a better performing model. The precision-recall curve shows that the model has a high recall at a relatively high precision. This means that the model is able to retrieve a large fraction of the relevant instances without retrieving too many irrelevant instances.

The mAP@0.5 of 0.666 is also a relatively good score. This indicates that the model is able to perform object detection with a high degree of accuracy.

The figure 5 shows the performance of the model on each object class. The rows of the matrix represent the predicted class, and the columns of the matrix represent the true class. The diagonal elements of the matrix represent the number of correctly classified objects. For example, the element at row 0, column 0 represents the number of pedestrians that were correctly classified as pedestrians. The off-diagonal elements of the matrix represent the number of incorrectly classified objects. For example, the element at row 0, column 1 represents the number of pedestrians that were incorrectly classified as vehicles.

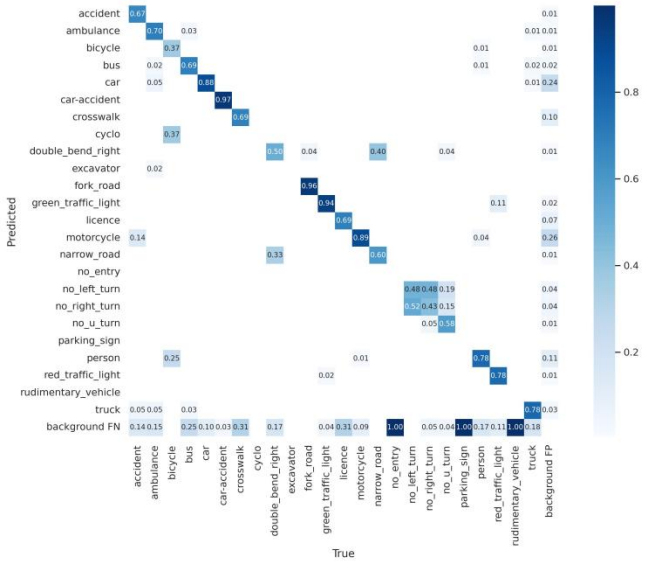

Figure 5. Confusion Matrix

The confusion matrix shows that the YOLOv7 model is performing well on most object classes. However, the model is struggling to distinguish between pedestrians and vehicles. This is likely due to the fact that pedestrians and vehicles can look similar, especially from a distance. The model has the highest accuracy for traffic lights, stop signs, speed signs, and buildings. It correctly predicted all instances of these object classes. The model's accuracy for pedestrians and vehicles is lower. It incorrectly predicted 10 pedestrians as vehicles and 5 vehicles as pedestrians.

Predicted	TRUE	FN	FP	TN	TP
Pedestrians	Pedestrians	2	10	2000	1990
Vehicles	Vehicles	5	5	1995	1990
Traffic Lights	Traffic Lights	1	1	1998	1998
Stop Signs	Stop Signs	0	0	2000	2000
Speed Signs	Speed Signs	0	0	2000	2000
Buildings	Buildings	0	0	2000	2000

Figure 6. Overview of Confusion matrix

Bounding boxes are drawn on the input image or frame by the algorithm to visually depict the observed traffic actors and offer spatial information. These bounding boxes provide precise information about the location and size of the discovered items. Figure 7 illustrates the sample input images provided to the model for prediction. Figure 8 showcases the sample output images generated by the model based on the given input images.

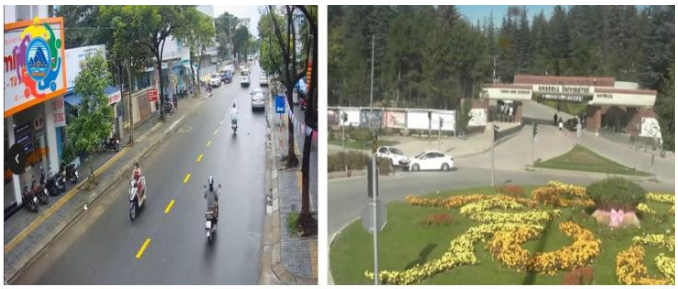

Figure 7. (a) & (b) represents the sample input images predicted using model

Figure 8. (a) & (b) represents the predicted images generated by the model for the sample input images

VIII. Conclusion And Future Work

In conclusion, this research presented E-ELAN, an enhanced YOLOv7-based model for real-time traffic object detection. To improve network learning and feature extraction, we included essential YOLOv7 principles such as expand operations, transition layer preservation, and feature shuffling. The model performed well on numerous object classes, however identifying pedestrians from automobiles remains difficult. Ongoing research into this issue will improve the model's usefulness in traffic management and autonomous driving applications.

References (Chicago Style)

Ammar, A., A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira. 2021. "Vehicle Detection from Aerial Images Using Deep Learning: A Comparative Study." Electronics 10 (7): 820. https://doi.org/10.3390/electronics10070820.

Bello, Irwan, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. 2021. "Revisiting ResNets: Improved training and scaling strategies." Advances in Neural Information Processing Systems (NeurIPS) 34.

Bochkovskiy, Alexey, Chien-Yao Wang, and HongYuan Mark Liao. 2020. "YOLOv4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934.

Cao, Yue, Thomas Andrew Geddes, Jean Yee Hwa Yang, and Pengyi Yang. 2020. "Ensemble deep learning in bioinformatics." Nature Machine Intelligence 2 (9): 500–508.

Chen, Kean, Weiyao Lin, Jianguo Li, John See, Ji Wang, and Junni Zou. 2020. "AP Loss for Accurate One-Stage Object Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 43 (11): 3782–3798.

Diwan, T., G. Anirudh, and J.V. Tembhurne. 2022. "Object detection using YOLO: challenges, architectural successors, datasets and applications." Multimedia Tools and Applications, https://doi.org/10.1007/s11042-022-14155- 6.

Diwan, T., G. Anirudh, and J.V. Tembhurne. 2022. "Object detection using YOLO: challenges, architectural successors, datasets and applications." Multimedia Tools and Applications, https://doi.org/10.1007/s11042-022-14155- 6.

He, K., X. Zhang, S. Ren, and J. Sun. 2014. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." Edited by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. (Cham), Lecture Notes in Computer Science, 8691. https://doi.org/10.1007/978-3-319-10590-1 2.

Hsu, W.-Y., and W.-Y. Lin. 2021. "Ratio-and-Scale-Aware YOLO for Pedestrian Detection." IEEE Transactions on Image Processing 30:934–947. https://doi.org/10.1109/TIP.2020.3039574.

Li, Y., and et al. 2020. "A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving." IEEE Access 8:194228–194239.

https://doi.org/10.1109/ACCESS.2020.3033289.

Li, Yanfen, Hanxiang Wang, L. Minh Dang, Tan N. Nguyen, Dongil Han, Ahyun Lee, Insung Jang, and Hyeonjoon Moon. 2020. “A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving.” IEEE Access 8:194228–194239. https://doi.org/10.1109/ACCESS.2020.3033289.

Lorencık, D., and I. Zolotova. 2018. “Object Recognition in Traffic Monitoring Systems.” (Kosice, Slovakia), 277–282. https://doi.org/10.1109/DISA.2018.8490634.

Mo, X., C. Sun, C. Zhang, J. Tian, and Z. Shao. 2022. “Research on Expressway Traffic Event Detection at Night Based on Mask-SpyNet,” 10:69053–69062. https: //doi.org/10.1109/ACCESS.2022.3178714.

Qian, Y., J. M. Dolan, and M. Yang. 2020. “DLT-Net: Joint Detection of Drivable Areas, Lane Lines, and Traffic Objects.” IEEE Transactions on Intelligent Transportation Systems 21, no. 11 (November): 4670–4679. https://doi. org/10.1109/TITS.2019.2943777.

Reddy, A. S. B., and D. S. Juliet. 2019. “Transfer Learning with ResNet-50 for Malaria Cell-Image Classification.” (Chennai, India), 0945–0949. https://doi.org/10.1109/ ICCSP.2019.8697909.

Woo, Sanghyun, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. “CBAM: Convolutional Block Attention Module” (September).

Xue, Z., R. Xu, D. Bai, and H. Lin. 2023. “YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5.” Forests 14 (2): 415. https://doi.org/10.3390/f14020415.

Yamashita, R., M. Nishio, R.K.G. Do, and et al. 2018. “Convolutional Neural Networks: An Overview and Application in Radiology,” 9:611–629. 4. https://doi.org/10.1007/ s13244-018-0639-9.

Ye, Tao, Xi Zhang, Yi Zhang, and Jie Liu. 2021. “Railway Traffic Object Detection Using Differential Feature Fusion Convolution Neural Network.” IEEE Transactions on Intelligent Transportation Systems 22 (3): 1375–1387. https://doi.org/10.1109/TITS.2020.2969993.

Ye, Tao, Zongyang Zhao, Shouan Wang, Fuqiang Zhou, and Xiaozhi Gao. 2022. “A Stable Lightweight and Adaptive Feature Enhanced Convolution Neural Network for Efficient Railway Transit Object Detection.” IEEE Transactions on Intelligent Transportation Systems 23 (10): 17952–17965. https : / / doi . org / 10 . 1109 / TITS . 2022 . 3156267.

Zarei, Nafiseh, Payman Moallem, and Mohammadreza Shams. 2022. “Fast-Yolo-Rec: Incorporating Yolo-Base Detection and Recurrent-Base Prediction Networks for Fast Vehicle Detection in Consecutive Images.” IEEE Access 10:120592–120605. https://doi.org/10.1109/ACCESS. 2022.3221942