FYNet: A Novel Architecture for Real-time Vehicle Attributes Detection and Tracking on a Multi Lane Highway

Seyyed Aliakbar Hosseini [1], Hossein Khosravi [2*]

1- Ph.D. Student of Electronics, Faculty of Electrical Engineering, Shahrood University of Technology, Daneshgah Blvd., Shahrood, Iran. P.O. Box: 3619995161.
Email: sa.hosseini@shahroodut.ac.ir, Phone: +989132654483
2- Associate Professor of Electronics, Faculty of Electrical Engineering, Shahrood University of Technology, Daneshgah Blvd., Shahrood, Iran. P.O. Box: 3619995161.
Email: hosseinkhosravi@shahroodut.ac.ir (*Corresponding Author) Phone: +989361392929

Abstract

Real-time vehicle detection and tracking in intelligent transportation systems face challenges due to lighting variations, occlusions, high-resolution imaging, and the need for high accuracy and efficiency. Existing systems often struggle to balance computational efficiency with fine-grained vehicle attribute detection. This paper proposes FYNet (Fast Yolo Net), a YOLOv5-based architecture for real-time vehicle localization and attribute recognition on six-lane highways. FYNet introduces a novel Path Aggregation Network (PANSum) to enhance multi-scale feature extraction, reduce computational overhead, and improve detection accuracy for objects ranging from license plates to long vehicles. With five output branches at different scales, FYNet achieves a robust detection with an inference speed of 16.3ms per 4K image (60 FPS), and reduces computations to 0.6 GFLOPs. For tracking, it employs StrongSORT to assign unique vehicle IDs. An effective strategy of separating front and rear views of vehicles into distinct classes also improved the model's mean average precision (mAP) by 1.8% while reducing model parameters. The model was evaluated on IRVA, a private dataset with 300K+ labels across 11 classes, offering new features compared to existing ITS datasets. FYNet outperforms standard YOLOv5 in both inference speed and accuracy, improving object recognition accuracy by 0.6% while maintaining real-time 4K processing.

Keywords: YOLOv5; FYNet; Real-time localization; DeepSORT; Intelligent Transportation System

1. Introduction

The development of smart transportation and traffic management systems is a key area of interest in the current era. These systems aim to improve the efficiency, safety and sustainability of transportation networks using artificial intelligence, cloud computing, and Internet of Things.

ITS offers a solution to the issues of traffic congestion and carbon emissions brought on by the sharp rise in vehicle populations [1]. Governments have released policies to support smart transportation, integrating traditional infrastructure with advancements in ITS, communications, and AI. These systems have improved productivity, reduced environmental impact, and enhanced traffic safety and accessibility. However, some drawbacks need immediate attention. [2].

Key issues in intelligent traffic control include identifying vehicle types, detecting the vehicle's position, reading the license plate, and estimating the vehicle's speed [3, 4]. The vehicle localization aims to identify the car bounding box from the stationary and moving objects. The purpose of vehicle type identification on the other hand is to specify the type of vehicles, including cars, vans, trucks, buses, and motorcycles.

The license plate number is an essntial inforamtion for any vehicle. To read the license plate, it is necessary to find its rectangle accurately and send it for license plate recognition (LPR) [5]. Numerous applications rely heavily on automatic LPR [6], and several methods have been proposed. However, most of them operated in confined spaces with adjusted lighting, constrained vehicle speed and static background. In this study almost no restrictions on the working environment are considered.

Currently, no dataset contains the whole attributes of vehicles, such as license plate and brand logo. Most of the existing datasets only include the class of the vehicle [7-9]. Accordingly, there is a good dataset that is used for self-driving cars. This dataset is expensive, includes images of cars, traffic signs, and lights, and does not have license plate labels. The license plates on a highway are read using a 2MP camera covering about 4 meters of the roadway. In order to read license plates on a 6-lane highway, six cameras are required, which is not economic [10].

In this paper we use a 4K camera installed on a six-lane highway instead of installing 6 different cameras. Figure 1 shows the block diagram of the vehicle attributes recognition system based on vehicle detection and tracking.

Since each car is in the camera's field of view for about 3 seconds, if the tracking is not used, a single vehicle will be processed and reported about 90 times (3 * 30 FPS) which is not reasonable. So, to record only one witness image for each car, the object tracking is used to track several cars simultaneously.

Since the aim here is to detect and track cars in real-time on a 6-lane highway, object detection, and tracking methods are considered. Accordingly, after reviewing the previous works and

having some experiments with tracking algorithms, the Strong SORT[1] [11] which is an improved version of the Deep SORT [12] algorithm, was selected to track cars. DeepSORT is based on the SORT and designed based on the Kalman filter and the Hungarian algorithm.

Figure 1.

Using an end-to-end neural network to forecast bounding boxes and class probabilities simultaneously is what YOLO (You Only Look Once) suggests [13]. It is distinct from the strategy used by earlier object identification systems that used classifiers as detectors.

YOLO is widely favored in object detection and segmentation due to its unique combination of speed, accuracy, and simplicity. Unlike traditional methods, YOLO processes images in a single forward pass, enabling real-time performance even on high-resolution inputs. Its single-stage architecture simplifies implementation and reduces computational overhead, making it highly efficient for both training and deployment on edge devices.

YOLO achieves competitive accuracy through continuous improvements, such as advanced backbones (e.g., CSPDarknet), feature extraction techniques (e.g., PANet, FPN), and optimized loss functions (e.g., CIoU). It is highly scalable and flexible, with models ranging from lightweight versions (e.g., YOLOv5n) for resource-constrained devices to larger versions (e.g., YOLOv5x) for high-accuracy tasks.

YOLOv5n6 is a light-weigth structure for recognizing objects in the street. However, it needs to be improved because it has challenges in recognizing the small objects in a 4k image of the six-lane highway. Besides, we want our algorithm to be real-time, but YOLOv5n6 has a computing requirement of about 1.7 GFLOPs[2] and an inference time of 40ms. This inference time will be increased after adding the StrongSORT algorithm for object tracking. So we propose an improved model, called FYNet, that can process 4K images in real-time. The details of the proposed model are given in Section 3.4. This model achieves the highest detection accuracy compared to the previous works [14, 15]. It detects various attributes of the vehicles, including bounding box, vehicle type, license plate, and manufacturer brand.

Almost all available datasets of vehicles have not distinguished between front and rear view of the vehicles. Separating the front and rear views as distinct classes is considered in our work. This technique has improved the mean average precision of the model (mAP) by about 1.8%.

[1] Simple Online and Realtime Tracking

[2] **Fl**oating point **op**eration**s**

The innovations of this paper are as follows:

- A new architecture, called FYNet is proposed for object detection and classification.
- PANSum block introduced to replace PANet and improve the feature representation.
- Five output branches are used in FyNet to better recognize objects of varying sizes from bus to license plate
- Several techniques used in FYNet, to reduce the number of FLOPs from 1.2 GFLOPs down to 0.6 GFLOPs and significantly increase the inference speed (60 FPS on 4K images).
- StrongSort is used for tracking vehicles and helps the classifier for better decisions.
- A private dataset, IRVA, with more than 300K labels from 11 classes, is presented for evaluation.

The rest of the paper is organized as follows: in Section 2 we reviewd related works. Section 3 presents the methodology in detail including the proposed architecture based on YOLOv5 and the tracking algorithm based on StrongSORT. Section 4 is dedicated to the experimental results and discussion. Finally, Section 5 concludes the paper.

2. Literature review

Most early studies and recent works focus on LPR, considering the YOLO as their base model [16-18]. Depending on the cutting-edge YOLO object detector, the ALPR system presented in the study by Laroca et al. [19] was reliable and effective. For each ALPR step, the Convolutional Neural Networks (CNNs) were trained and adjusted to be resilient over various circumstances. They trained a two-stage model for character segmentation and recognition using data augmentation. Their method is evaluated on SSIG dataset, which consists of 2,000 frames from 101 vehicle videos. It achieved a recognition rate of 93.53% and a performance of 47FPS.

In another study an end-to-end generic ALPR pipeline was considered based on YOLO for the detection and recognition of license plates, vehicles, and objects [20]. The authors implemented the entire ALPR pipeline from vehicle detection to the LP identification stage. YOLOv2 was used in the first stage, and the YOLOv4 detector was used in the latter steps.

Object recognition using deep neural networks is divided into two general categories; two-step methods like RCNN [21] that finds some bounding boxes using selective search algorithm and then classifies the object inside the bounding box using a convolutional network, and one-step methods that performs detection and classification of objects at the same time. In the two-step methods, the input image to must have a fixed size, and to solve this, the idea of matching the

spatial pyramid [22] or SPP is introduced. Using feature extraction methods such as FPN [23], the classification accuracy of these networks was improved.

The most important one-stage methods are Single Shot Multi-Box Detector (SSD) [24] and YOLO [25]. In SSD the region proposal network (RPN) [26] and multi-scale representation method are used. Unlike SSD, the YOLO divides the image into several cells. YOLO frames object detection as a single regression problem. This approach allows YOLO to simultaneously predict bounding boxes and class probabilities directly from full images in one evaluation, making it exceptionally fast.

The YOLO model along with a series of post-processing rules has been used in automatic license plate recognition (ALPR) system [27]. This system included a unified approach for license plate (LP) detection and layout classification. In order to achieve the optimal speed-accuracy trade-off, the system was developed by analyzing and optimizing various models.

Deep learning-based object detection algorithms may achieve autonomous learning and have robust detection; therefore, they are often not constrained by the application scenario [28]. These algorithms are currently more developed and can perform better in some circumstances, including pedestrian detection [29], face detection [30], etc. These methods may be applied broadly in intelligent monitoring systems [31], intelligent transportation systems [32], object detection in the military, object detection in medicine, etc. In more specialized settings, however, there is more significant potential for improvement when there are issues with occlusion, too small size objects and distortion [32].

Wang et al. [33] achieved high identification and tracking accuracy for small target vehicles considering a unique attention-based vehicle detection and tracking strategy. By incorporating the normalization-based attention module (NAM) into the traditional YOLOv5s model, a new vehicle detection model (YOLOv5-NAM) was designed which performs identification and tracking in real-time.

A vehicle tracking solution for traffic scenes was provided by Yang et al. [34] based on a detection-based tracking (DBT) framework since deep learning and correlation filter (CF) tracking are time-consuming. The YOLO model was utilized to create the vehicle detection model, and the object attribute information and intersection over union (IOU) constraints were merged to change the vehicle detection box. A simple feature extraction network model was built for vehicle tracking. This model used an inception module to lighten the computational

burden and improve the network scale's adaptability. In order to improve feature learning, a squeeze-and-excitation channel attention technique was used.

In 2019, Hossain and Lee [35] employed a specified GPU-based embedded computing modules to examine the most advanced deep learning-based multi-object detection algorithms. The authors combined Deep SORT with a deep learning-based association metric and presented a powerful method for tracking moving objects.

Shi et.al., introduced a lightweight vehicle detection algorithm based on YOLOv7-tiny, achieving significant reductions in computational complexity (9.2 GFLOPs) and improvements in detection speed and accuracy [36]. Li and Zhang proposed a fast vehicle detection algorithm using ShuffleNetv2 and a multi-attention mechanism, reducing computational complexity by 55.70%. However it loses 1% in accuracy and 2.6% in mAP [37]. Another study worked on nighttime vehicle detection using Deformable Convolutional Network combinded with Faster R-CNN and Soft-NMS [38], addressing challenges like low illumination and sample imbalance. For more comprehensive review of the existing methods for vehicle detection considering their performances and applications, readers are referred to the review paper provided by Liang et.al., [39].

From the literature review, we found that some important issues related to ITS have been less addressed. In this paper, we tried to tackle some of these gaps. The first is the need for real-time vehicle detection on multi-lane highways using a single 4K camera, a capability of immense practical and operational importance. Leveraging a single camera instead of multiple cameras, not only reduces hardware costs but also simplifies deployment, making it economically and logistically cost-effective.

The second gap addressed in this research is the lack of a lightweight neural network capable of processing 4K images in real-time with a computational load of less than 1 GFLOPs. This limitation was a major barrier to achieving efficient, real-time performance in high-resolution scenarios. To overcome this, we designed the FYNet architecture, which will be discussed in Section 3.4.

The third gap is the absence of a functional dataset tailored to Iranian vehicles, including vehicle types and license plates. To address this, we created the IRVA dataset, which provides a valuable resource for training and evaluating models in the context of Iranian traffic conditions. By addressing these gaps, this research not only advances the technical capabilities of ITS but also provides cost-effective and scalable solutions for real-world applications.

In this paper, we present an algorithm to detect all vehicles and their attributes simultaneously on a six-lane highway from a high-resolution 4k camera. We aim to automatically and accurately detect various vehicle characteristics, including bounding box, vehicle type, license plate, logo, and movement direction (forward or backward).

3. Methodology

This section gives details about the prepared dataset and the methodology for detecting and tracking vehicles and their attributes on a six-lane highway with a 4K camera.

3.1 IRVA dataset

In [40] a specific Berkeley DeepDrive (BDD) dataset was used for detection of the self-driving cars using the YOLOv4 network. Inspired by that work, we created an exclusive dataset called IRVA (Iranian Vehicles with Attributes) by collecting data from cameras installed on Iranian highways. To achieve an effective method for detecting and tracking vehicles on a six-lane highway, it was not possible to use the existing datasets for several reasons. In some datasets, such as COCO [41], cars like pickups are entirely different in appearance from those used in Iran, and hence they would have problems if used for recognition of Iranian pickups. On the other hand, we considered separating the rear and front sides of the vehicles which is not considered in existing datasets. This idea has been considered for two reasons: First, due to the similarity of the front views and the difference between front and back views, this separation somehow simplifies the classification so that a model with fewer parameters can be used. Second, separating front and rear views, helps us to understand the vehicle is coming or going.

While preparing the dataset, the rear and front views of all types of common Iranian vehicles, from different angles have been used. The images were taken at different hours of the day and night and in different weather conditions, such as sunny, cloudy, and rainy. The shooting locations are at different heights up to 6 meters, with a field of view of one, two, three, and six lanes. Furthermore, day, night, and twilight images are placed in different folders by appropriate abbreviations. In order to be able to report the strengths and weaknesses of the data collection and finally report the results more accurately, images were put into separate folders with codenames. Abbreviated codenames of the folders are shown in Table 1.

Table 1

The 4K images of the six-lane highway have a resolution of 2160 × 3840 pixels, but due to the memory limitations of RAM and GPU, the resolution is reduced to 960 × 1280 pixels before being fed to the model.

Number of labels in each folder is shown in Table 2. According to this table, 95,951 images with 278,954 labels were collected during the day, and 15,302 images with 34,430 labels were collected during the night. The final dataset has three folders including TRAIN that will be used for training the model, VAL for validation, and TEST for evaluation of the model.

Table 2

Table 3 shows the number of labels of each class for the training, validation, and test datasets. Here we used 70% of the data for the train, 15% for validation and 15% for the test. Because there is more than one label in some images of the IRVA dataset, such as highway images, the number of labels is not equal to the number of images.

Table 3

Sample images of the IRVA dataset is shown in Figure 2. It includes photos from six-lane highway with a camera height of 6-meters with different magnifications and wide and telephoto lens, images of heavy vehicles at the entrance of mines and parking lots with, and some other images in different conditions.

Figure 2

The IRVA images, around 111K samples, were labeled using the VOTT software[1]. In this application we can create bounding box for each object and assign the appropriate label. Figure 3, shows the VOTT environment and sample images with their labels.

Figure 3

3.2 Base architecture of the proposed model

After building the dataset, the YOLOv5[42] is selected for localization and classification of the objects. YOLOv5 is a state-of-the-art object detection model that uses a convolutional neural network to perform fast and accurate detection of multiple objects in an image [42]. The architecture of YOLOv5 consists of four main components: backbone, neck, head, and auxiliary branch (Figure 4).

Figure 4

YOLOv5 is an improved version of YOLOv4 [43]. It uses CSPNet as the backbone which is more efficient and lightweight than Darknet53, backbone of YOLOv4, reducing the number of parameters and increasing the processing speed. It also uses PANet as the neck which is more

[1] https://github.com/microsoft/VoTT

effective than SPP and SAM, enhancing the feature aggregation and reducing the complexity. Furthermore, it uses FPN as the head, that is more scalable and flexible than YOLOv4 head, allowing users to adjust the model size and input resolution. Finally, YOLOv5 has an optional auxiliary branch for segmentation which can improve the performance of the model by providing additional information and supervision for the detection task.

3.3 Separating the front and rear view of the vehicles

At first, we conducted an experiment to find whether it's better to separate the front and rear view of the vehicles. So, we trained the model once on the dataset with separated classes for rear and frontal view (Table 4) and once without separation of them (Table 5). The empirical findings emphasized the striking performance of the model when the front and rear view of the vehicles have separated classes. This simple technique increased the performance of the YOLO by 1.8% in mAP metric, as indicated in Tables 4 and 5.

Table 4

Table 5

3.4 The proposed model

Choosing a suitable deep network structure is very important to have a good performance for object detection in highway images. The YOLOv5 is fast, efficient, and scalable which makes it suitable for this purpose. However we need the model to be real-time, so some improvements should be made in this structure to have the minimum inference time while maintaining accuracy. To achieve these requirements, we made some modification on the original YOLOv5. The major changes made on YOLOv5n is as follows:

- Combining feature maps at different scales for better accuracy
- Introducing PANSum block instead of PANet block to improve the quality of feature representation
- Using SPPF block to improve the receptive field without imposing computational cost
- Introducing five output branches to better recognize objects of varying sizes from bus to license plate
- Reducing the number of convolution channels in some layers to reduce the computational cost while maintaining the accuracy

Figure 5 shows the details of the proposed model, called Flexible and Fast Yolo (FYNet). The computational performance of this model is 0.6GFLOPs which is suitable for real-time processing. It only takes 16.3ms to process a 4K image of the highway.

Figure 5

3.4.1 The idea behind the FYNet model

In the proposed FYNet model, the primary purpose is well detection and classification of small targets such as license plates along with regular and large objects like cars and long-vehicles. This was achieved by increasing the number of output heads at different resolutions and combining features from the backbone network with those from the PANSum block. This led to a 0.3% improvement in mAP50 and a 0.8% improvement in mAP50-95 at 1280 pixels. Additionally, we optimized the model's depth and width through extensive trials, achieving a balance between parameter efficiency and performance.

The next goal is to reduce the computational cost to have a real-time response. To achieve real-time performance, we decreased the number of convolutional channels in each layer by a factor of 2, resulting in a 50% reduction in FLOPs (from 1.2GFLOPs to 0.6GFLOPs) and significant improvements in inference speed (38% faster on GPU and 35% faster on CPU at 3840 pixels).

To have better accuracy, we combined features of the backbone that contain details about small objects with those features of the PANSum block enriching the final features used for classification. In this way, small objects like license plates were detected more accurately.

In order to effectively use the depth and width parameters, several structures with different combinations of the layers and the size of the kernels were tried, and finally, we achieved the structure of Figure 5.

3.4.2 PANSum block in FYNet model

Feature maps at different depths of the model contain different information. Initial layers contain low-level details such as the edges and corners of the objects, while the last layers contain high-level semantic information such as the location and class of the objects. Therefore, combining feature maps from different layers improves the segmentation accuracy even in complex backgrounds[44].

Recently two blocks have been used for the connection of feature maps at different scales, Feature Pyramid Network (FPN) and Path Aggregation Network (PANet). FPNs use a top-

down pathway and lateral connections to combine low-resolution, semantically strong features with high-resolution, semantically weak features. PANet consists of a FPN and a path aggregation module (PAN) that enhances the feature representation by aggregating information across different scales.

In the proposed model we introduced a modified PAN block, called PANSum (Figure 6). In this figure, nodes P3, P4, P5, P6, and P7 come from C3 blocks and go to the intermediate layers as shown in Figure 5, while CNVs come from ConvBNSILU modules of the backbone and directly go to the final layers (see red lines in Figure 5 and Figure 6). In fact, these new connections bypass the middle layer in a way similar to the skip connections of the UNET++ model [45]. These connections allow the features of the initial layers, which have a larger scale, to be used in the last layers, making the final feature map richer. These feature maps serve as important inputs for subsequent stages, allowing for more accurate object detection and classification. The structure of PANSum consists of four components: top-down path, bottom-up path, skip connections and adders. In the top-down path, the scale of the feature maps is reduced using the backbone convolutional network. In this path, the semantic and contextual information is increased. In the bottom-up path, the resolution of the feature maps is increased, and the feature maps are enriched by combining the feature maps of the top-down path using skip connections. Semantic information at lower scales is combined with detailed information on higher scales.

Adding feature maps of the Conv section in the backbone with feature maps of the same level in the top-down path before the head of the network enriches the semantic features. This block, increased the object recognition accuracy by 0.6%. The new sum block introduced in PANSum has very low computational overhead and doesn't affect the real-time performance.

Figure 6

Before training the model, it is helpful to have a statistical view of the distribution of labels and their dimensions. Figure 7 shows the chronogram of the labels and how they are distributed. The following information can be extracted from these graphs at a glance: the most significant number of samples belongs to the license plate class, and then passenger cars from the front, and the lowest number of samples belongs to the pickup class from the back. It is clear that the model will more learn those classes that have more samples. In most examples, the center of the Bbox is around the middle square in the image, and therefore, we are facing a dataset with complete images of cars, which the use of data augmentation or mosaic data augmentation can increase the generalization power of the model in detecting incomplete views of vehicles. There

is an accumulation of samples in the dataset for small objects, shown in bold red in the anchor box and dark navy. Therefore, increasing the number of the model heads with higher resolutions can significantly help recognize small objects such as car brands, models, and license plates.

Figure 7

The aim is to change YOLOv5 architecture to reduce the number of parameters down to 1M parameters and the approximate floating operations of 0.6 GFLOPs, so that a 4K image can be processed in less than 20ms. The tiniest version of YOLOv5 is the YOLOv5n6 network, with 281 layers, 3.2M parameters, 4.6GFLOPs, and inference time of 37.8ms for 4K images. Using this model, the real-time processing of the cars on highway is not reachable. Specially, when adding the tracking algorithm, the inference time for each frame exceeds 60ms, which means 16FPS and is far from real-time requirement.

At the first step, we removed some layers and reduced the number and width of filters in several layers of the YOLOv5n6 model. The new model named YOLOv5p6 has 786K parameters, 206 layers and 1.2GFLOPs and reached the inference time of 21.3ms. Adding NMS[1] time of 1.1ms for 4K images total inference time reached 22.4 ms. After many modifications to the YOLOv5 model, we managed to introduce a very light and fast model called FYNet[2]. This model has changes in the head part of the network and combines the feature maps of the backbone part using PANSum block, as shown in Figure 5 block diagram.

The proposed model has 244 layers, 1.7M parameters, and a calculation volume of 0.6GFLOPs, which has an inference time of 15.43ms and NMS time of 0.9ms. The main innovation of the proposed model is replacing the PAN block with the PANSum proposed block, as well as increasing network depth while reducing the width of the convolution channels, which leads to a reduction in the number of FLOPs from 1.2 GFLOPs to 0.6 GFLOPs or 50% reduction in computing volume. The increase in the number of network headers from 4 to 5 caused the parallelization of the NMS algorithm calculations and a significant reduction in the time of the NMS algorithm from 1.1ms to 0.9ms. It is noteworthy that convolution, deviation, and batch normalization parameters are included in the total number of parameters for each structure. Table 6 presents the upgraded network architecture.

Table 6

[1] Non-Maximum Suppression
[2] Code is available at: https://github.com/SAAHosseini/FYNet

3.5 Model training

Several tests have been performed considering different hyperparameters to train the FYNet architecture on the IRVA dataset, and the values that lead to the best detection results have been considered for model training.

In table 7, the various hyperparameters utilized during model training are presented, along with their corresponding values and explanations. These hyperparameters, such as learning rate, batch size, and number of epochs, play a crucial role in the training process and can significantly impact the model's performance. The learning rate determines the step size at which the model's weights are updated during training, while the batch size specifies the number of samples used to update the weights in each iteration. The number of epochs refers to the total number of passes through the training data during training. By carefully selecting these hyperparameters, it is possible to achieve optimal model performance.

Table 7

4. Results and discussion

The performance of the adopted methodology is illustrated and compared with the state-of-the-art architectures. The findings emphasize the superiority of the proposed model over the rest. The IRVA dataset is used to train both YOLOv5p6 [46] and FYNet.

As shown in Table 8, the YOLOv5 [47] has five different sizes, separated by the suffixes n, s, m, l, and x. The YOLOv5n6 has fewer parameters than small, medium, large, and very large. The more complicated the network, the higher the efficiency, the longer the inference time, and the longer the training time. This table indicates the different sizes of the YOLOv5 network in terms of the number of parameters, the inference time on the Tesla 100 graphics card, and the mAP on the COCO dataset. Nano and small sizes are recommended for real-time purposes.

Table 8

The comparison between the proposed FYNet model and YOLOv5p6, shown in table 9, demonstrates FYNet's superior performance across multiple metrics. FYNet achieves a 0.3% improvement in mAP50 and a 0.8% improvement in mAP50-95, indicating better detection accuracy. Notably, FYNet reduces FLOPs by 50%, highlighting its computational efficiency. This efficiency translates into faster inference times, with 38% improvement on GPU and 35% improvement on CPU, making FYNet more suitable for real-time applications. Additionally, FYNet shows an 18% reduction in NMS (Non-Maximum Suppression) time, further enhancing its speed. These improvements are consistent across various input resolutions (640p to 3840p),

with FYNet maintaining its advantage in both speed and accuracy. The results underscore FYNet's ability to balance performance and efficiency, making it a robust choice for high-resolution, real-time object detection tasks such as vehicle detection on multi-lane highways.

Table 9

Table 10 shows the system used for training and testing the aforementioned networks. It has an RTX3060 graphics card with 12GB of graphics memory and a 2.59GHz Core i5-11400 processor with 32GB of memory.

Table 10

Figure 8 shows the training graphs and all metrics including detection loss, classification loss, mAP, precision and recall during 300 epochs.

Figure 8

Figure 9 shows F1-Score graphs and confusion matrix. The confusion matrix displays the classification results based on the available information. This matrix is a performance measurement method for ML classification problems where the output can be two or more classes. The confusion matrix is necessary to calculate the evaluation criteria of Accuracy, Precision, Recall, and F1. As can be seen in this matrix, the network has more error in recognizing some classes such as R_PICKUP. The main reason of this error is the lack of images and training samples in the IRVA dataset. The F1-Score criterion obtains the precision and recall criteria and is one of the effective criteria for evaluating the trained model[48].

The F-measure is the weighted harmonic mean of precision (P) and recall (R) of a classifier, taking $\alpha=1$ (F1 score). It means that both metrics have the same importance. Our graph's confidence value that optimizes the precision and recall is 0.416, corresponding to the maximum F1 value (0.88). In most cases, a higher confidence value and F1 score are desirable. F1-Score is equal to one at best and zero at worst performance.

Figure 9

The evaluation results of the proposed model on the IRVA dataset, demonstrate its strong performance across various classes (Table 11). FYNet achieves an overall mAP50 of 94.0% and a mAP50-95 of 72.7%, indicating high detection accuracy across different Intersection over Union (IoU) thresholds. The model excels in detecting specific classes, such as front buses (F-BUS) with a precision of 97.3% and a recall of 98.1%, and front trucks (F-TRUCK) with a mAP50 of 98.6%. Notably, license plates (PLATE) are detected with a high

recall of 93.5% and a mAP50 of 96.7%, showcasing FYNet's effectiveness in critical tasks like license plate recognition.

However, the study has some limitations, primarily due to insufficient samples in certain classes. For example, classes like pickup (F-PICKUP, R-PICKUP), truck (F-TRUCK, R-TRUCK), and motorcycle have fewer samples compared to passenger cars, with a ratio of 1:20 in a single highway image. This imbalance leads to reduced accuracy in these classes and, consequently, a decrease in the overall average accuracy.

The training graphs confirm that the model has not encountered overfitting and is capable of further improvement. Over time, as more samples are collected and the dataset is enriched, especially for underrepresented classes, the accuracy is expected to increase, and errors will decrease. For example, the license plate class, which has a large number of samples, already achieves near-perfect mAP (close to 1), demonstrating the impact of dataset size on model performance.

In conclusion, while FYNet shows robust performance overall, enriching the dataset for underrepresented classes like pickups, trucks, and motorcycles will further enhance its accuracy and reliability, making it even more suitable for real-world ITS applications.

Table 11

Figure 10 shows the output images of the FYNet model, in recognizing the vehicle attributes on the highway. The algorithm's detection speed is 60 FPS in the hardware environment depicted in Table 10. The FYNet vehicle detection algorithm is capable of detecting and accurately classifying vehicles inside this window. A processing speed of 25 frames per second is achieved by the FYNet + StrongSORT algorithm in the hardware setup depicted in Table 10.

Figure 10

In Figure 11, the output images of tracking and identifying cars on a six-lane highway can be seen. The StrongSORT algorithm, along with the lightweight, fast, and accurate object detection model FYNet, has been able to detect the location, class, ID, and movement path of the vehicles in real-time with a 40ms inference time for highway images with 4K resolution at a rate of 25FPS.

Figure 11

5. Conclusion

In this paper, a novel model was proposed to recognize and track vehicles and their attributes on highways. Using the proposed method, a six-lane highway with only one camera with 4K resolution was examined. Various vehicle attributes, including bounding box, vehicle type, license plate, and manufacturer, were detected. According to the real-time requirements of the algorithm, several optimizations made on YOLOv5 and a new model called FYNet was designed. These include intelligently reducing the width of the filters, increasing the number of layers, replacing the PAN block with the proposed PANSum block, and replacing the head-5D block in the head section of the model. It achieved mAP of 94% on 4K images with an inference time of 16.33ms which means 60FPS. A new dataset, IRVA, was prepared with images taken in different conditions from day and night and from different angles. Also, considering the importance of extracting the characteristics of vehicles on the highway, 24 hours a day, the FYNet model was trained to be resistant to ambient light and different weather conditions. It should be noted that combining all these features on a comprehensive control and monitoring system using only one 4K camera and one processor simultaneously for a six-lane highway is a convenient and cost-effective idea for the users of the ITS industry. Combining the FYNet with the StrongSORT object tracking technique led to outstanding results that made it suitable for real-time applications like traffic control and law enforcement.

Consent for publication

Not applicable.

Data availability statement

The data used in this study will be made available for research purposes from the corresponding author upon reasonable request.

Competing interests

The authors did not declare any conflict of interest.

Funding

Not applicable.

References

1. Lv, Z. and Shang, W., "Impacts of intelligent transportation systems on energy conservation and emission reduction of transport systems: A comprehensive review", *Green Technologies and Sustainability*. **1**(1), pp. 100002 (2023). DOI: https://doi.org/10.1016/j.grets.2022.100002.
2. Chandra Shit, R., "Crowd intelligence for sustainable futuristic intelligent transportation system: a review", *Iet intelligent transport systems*. **14**(6), pp. 480-494 (2020). DOI: https://doi.org/10.1049/iet-its.2019.0321.
3. Venkatesan, K., Chavan, D., and Saxena, G., "An Innovative Deep Learning Model for Detecting Automobiles and Potholes", *International Journal of Scientific Methods in Intelligence Engineering Networks*. **01**, pp. 13-22 (2023). DOI: https://doi.org/10.58599/IJSMIEN.2023.1702.
4. Khosravi, H., Asgarian Dehkordi, R., and Ahmadyfard, A., "Vehicle speed and dimensions estimation using on-road cameras by identifying popular vehicles", *Scientia Iranica*. **29**(5), pp. 2515-2525 (2022). DOI: https://doi.org/10.24200/sci.2020.55331.4174.
5. Ibrahim, N.K., Kasmuri, E., Jalil, N.A., et al., "License plate recognition (LPR): a review with experiments for Malaysia case study", *arXiv preprint arXiv:1401.5559*, (2014). DOI: https://doi.org/10.7321/jscse.v3.n3.15.
6. Shyang-Lih, C., Li-Shien, C., Yun-Chung, C., et al., "Automatic license plate recognition", *IEEE Transactions on Intelligent Transportation Systems*. **5**(1), pp. 42-53 (2004). DOI: https://doi.org/10.1109/TITS.2004.825086.
7. Asgarian Dehkordi, R. and Khosravi, H., "Vehicle Type Recognition based on Dimension Estimation and Bag of Word Classification", *Journal of AI and Data Mining*. **8**(3), pp. 427-438 (2020). DOI: https://doi.org/10.22044/jadm.2020.8375.1975.
8. De Oliveira, I.O., Laroca, R., Menotti, D., et al., "Vehicle-Rear: A new dataset to explore feature fusion for vehicle identification using convolutional neural networks", *IEEE Access*. **9**, pp. 101065-101077 (2021). DOI: https://doi.org/10.1109/ACCESS.2021.3097964.
9. Gholamalinejad, H. and Khosravi, H., "IRVD: A Large-Scale Dataset for Classification of Iranian Vehicles in Urban Streets", *Journal of AI and Data Mining*. **9**(1), pp. 1-9 (2021). DOI: https://doi.org/10.22044/jadm.2020.8438.1982.
10. Wang, L., Li, L., Wang, H., et al., "Real-time vehicle identification and tracking during agricultural master-slave follow-up operation using improved YOLO v4 and binocular positioning", *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of*

Mechanical Engineering Science. **237**(6), pp. 1393-1404 (2022). DOI: https://doi.org/10.1177/09544062221130928.
11. Du, Y., Zhao, Z., Song, Y., et al., "Strongsort: Make deepsort great again", *IEEE Transactions on Multimedia*. **25**, pp. 8725-8737 (2023). DOI: https://doi.org/10.1109/TMM.2023.3240881.
12. Wojke, N., Bewley, A., and Paulus, D. "Simple online and realtime tracking with a deep association metric", in *2017 IEEE international conference on image processing (ICIP)*. IEEE, pp. 3645-3649.(2017). DOI: https://doi.org/10.1109/ICIP.2017.8296962.
13. Shashidhar, R., Manjunath, A., Kumar, R.S., et al. "Vehicle Number Plate Detection and Recognition using YOLO-V3 and OCR Method", in *2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)*. IEEE, pp. 1-5.(2021). DOI: https://doi.org/10.1109/ICMNWC52512.2021.9688407.
14. Jain, A., Gupta, J., Khandelwal, S., et al., "Vehicle license plate recognition", *Fusion: Practice and Applications*. **4**(1), pp. 15-21 (2021). DOI: https://doi.org/10.54216/FPA.040102.
15. Hsu, G.-S., Ambikapathi, A., Chung, S.-L., et al. "Robust license plate detection in the wild", in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, pp. 1-6.(2017). DOI: https://doi.org/10.1109/AVSS.2017.8078493.
16. Zhang, J., Yang, X., Wang, W., et al., "Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering", *Automation in Construction*. **146**, pp. 104699 (2023). DOI: https://doi.org/10.1016/j.autcon.2022.104699.
17. Farid, A., Hussain, F., Khan, K., et al., "A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments", *Applied Sciences*. **13**(5), pp. 3059 (2023). DOI: https://doi.org/10.3390/app13053059.
18. Fan, J., Wei, J., Huang, H., et al., "IRSDT: A Framework for Infrared Small Target Tracking with Enhanced Detection", *Sensors*. **23**(9), pp. 4240 (2023). DOI: https://doi.org/10.3390/s23094240.
19. Laroca, R., Severo, E., Zanlorensi, L.A., et al. "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector", in *2018 International Joint Conference on Neural Networks (IJCNN)*. pp. 1-10.(2018). DOI: https://doi.org/10.1109/IJCNN.2018.8489629.
20. Al-Batat, R., Angelopoulou, A., Premkumar, S., et al., "An end-to-end automated license plate recognition system using YOLO based vehicle and license plate detection with vehicle classification", *Sensors*. **22**(23), pp. 9477 (2022). DOI: https://doi.org/10.3390/s22239477.
21. Girshick, R., Donahue, J., Darrell, T., et al. "Rich feature hierarchies for accurate object detection and semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 580-587.(2014). DOI: https://doi.org/10.1109/CVPR.2014.81.
22. He, K., Zhang, X., Ren, S., et al., "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE transactions on pattern analysis and machine intelligence*. **37**(9), pp. 1904-1916 (2015). DOI: https://doi.org/10.1109/TPAMI.2015.2389824.
23. Lin, T.-Y., Dollár, P., Girshick, R., et al. "Feature pyramid networks for object detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2117-2125.(2017). DOI: https://doi.org/10.1109/CVPR.2017.106.
24. Redmon, J., Divvala, S., Girshick, R., et al. "You only look once: Unified, real-time object detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 779-788.(2016). DOI: https://doi.org/10.48550/arXiv.1506.02640.
25. Liu, W., Anguelov, D., Erhan, D., et al. "Ssd: Single shot multibox detector", in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, pp. 21-37.(2016). DOI: https://doi.org/10.1007/978-3-319-46448-0_2.
26. Li, B., Yan, J., Wu, W., et al. "High performance visual tracking with siamese region proposal network", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 8971-8980.(2018). DOI: https://doi.org/10.1109/CVPR.2018.00935.

27. Laroca, R., Zanlorensi, L.A., Gonçalves, G.R., et al., "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector", *IET Intelligent Transport Systems*. **15**(4), pp. 483-503 (2021). DOI: https://doi.org/10.1049/itr2.12030.
28. Shlezinger, N., Farsad, N., Eldar, Y.C., et al., "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection", *IEEE Transactions on Wireless Communications*. **19**(5), pp. 3319-3331 (2020). DOI: https://doi.org/10.48550/arXiv.1905.10750.
29. Dollár, P., Wojek, C., Schiele, B., et al. "Pedestrian detection: A benchmark", in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, pp. 304-311.(2009). DOI: https://doi.org/10.1109/CVPR.2009.5206631.
30. Diba, M. and Khosravi, H., "SNResNet: A New Architecture Based on SqNxt Blocks and Rish Activation for Efficient Face Recognition", *Traitement du Signal*. **41**(2), pp. 949-959 (2024). DOI: https://doi.org/10.18280/ts.410235.
31. Chen, B.-H. and Huang, S.-C., "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks", *IEEE transactions on multimedia*. **16**(3), pp. 837-847 (2014). DOI: https://doi.org/10.1109/TMM.2014.2298377.
32. Hua, X., Wang, X., Wang, D., et al., "Military object real-time detection technology combined with visual salience and psychology", *Electronics*. **7**(10), pp. 216 (2018). DOI: https://doi.org/10.3390/electronics7100216.
33. Wang, J., Dong, Y., Zhao, S., et al., "A High-Precision Vehicle Detection and Tracking Method Based on the Attention Mechanism", *Sensors*. **23**(2), pp. 724 (2023). DOI: http://dx.doi.org/10.3390/s23020724.
34. Yang, B., Tang, M., Chen, S., et al., "A vehicle tracking algorithm combining detector and tracker", *EURASIP Journal on Image and Video Processing*. **2020**(1), pp. 1-20 (2020). DOI: https://doi.org/10.1186/s13640-020-00505-7.
35. Hossain, S. and Lee, D.-j., "Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices", *Sensors*. **19**(15), pp. 3371 (2019). DOI: https://doi.org/10.3390/s19153371.
36. Shi, Q., Zhong, F., Li, B., et al. "Fast vehicle detection algorithm based on lightweight YOLO7-tiny", in *5th International Conference on Computer Vision and Data Mining (ICCVDM 2024)*. SPIE.(2024). DOI: https://doi.org/10.1117/12.3048399.
37. Li, J. and Zhang, J. "Fast vehicle detection method based on improved YOLOv5s", in *International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024)*. SPIE.(2024). DOI: https://doi.org/10.1117/12.3033636.
38. Xu, Y., Chu, K., and Zhang, J., "Nighttime vehicle detection algorithm based on improved faster-rcnn", *IEEE Access*. **12**, pp. 19299-19306 (2023).
39. Liang, L., Ma, H., Zhao, L., et al., "Vehicle Detection Algorithms for Autonomous Driving: A Review", *Sensors*. **24**(10), pp. 3088 (2024). DOI: https://doi.org/10.3390/s24103088.
40. Wu, F., Wang, D., Hwang, M., et al., "Decentralized Vehicle Coordination: The Berkeley DeepDrive Drone Dataset", *arXiv preprint arXiv:2209.08763*, (2022). DOI: https://doi.org/10.48550/arXiv.2209.08763.
41. Lin, T.-Y., Maire, M., Belongie, S., et al. "Microsoft coco: Common objects in context", in *13th European Conference of Computer Vision*. Zurich: Springer, pp. 740-755.(2014). DOI: https://doi.org/10.48550/arXiv.1405.0312.
42. Fang, Y., Guo, X., Chen, K., et al., "Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model", *BioResources*. **16**(3), pp. 5390 (2021). DOI: https://doi.org/10.15376/biores.16.3.5390-5406.
43. Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y.M., "YOLOv4: Optimal Speed and Accuracy of Object Detection", *arXiv preprint arXiv:2004.10934*, (2020). DOI: https://doi.org/10.48550/arXiv.2004.10934.
44. Tan, M., Pang, R., and Le, Q.V. "Efficientdet: Scalable and efficient object detection", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 10781-10790.(2020). DOI: https://doi.org/10.48550/arXiv.1911.09070.

45. Hoorali, F., Khosravi, H., and Moradi, B., "Automatic microscopic diagnosis of diseases using an improved UNet++ architecture", *Tissue and Cell*. **76**, pp. 101816 (2022). DOI: https://doi.org/10.1016/j.tice.2022.101816.
46. Zaman, F.H.K., Tahir, N.M., Yusoff, Y.M., et al., "Human Detection from Drone using You Only Look Once (YOLOv5) for Search and Rescue Operation", *Journal of Advanced Research in Applied Sciences and Engineering Technology*. **30**(3), pp. 222-235 (2023). DOI: http://dx.doi.org/10.37934/araset.30.3.222235.
47. Jia, W., Xu, S., Liang, Z., et al., "Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector", *IET Image Processing*, (2021). DOI: https://doi.org/10.1049/ipr2.12295.
48. Huang, H., Xu, H., Wang, X., et al., "Maximum F1-score discriminative training criterion for automatic mispronunciation detection", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. **23**(4), pp. 787-797 (2015). DOI: https://doi.org/10.1109/TASLP.2015.2409733.

Table 11. The mAP, recall, and precision of the FYNet on the IRVA dataset

Figure 1. A block diagram of vehicle attributes recognition system on the highway

Figure 2. Sample vehicle images from highway, mine and parking lots used in IRVA dataset

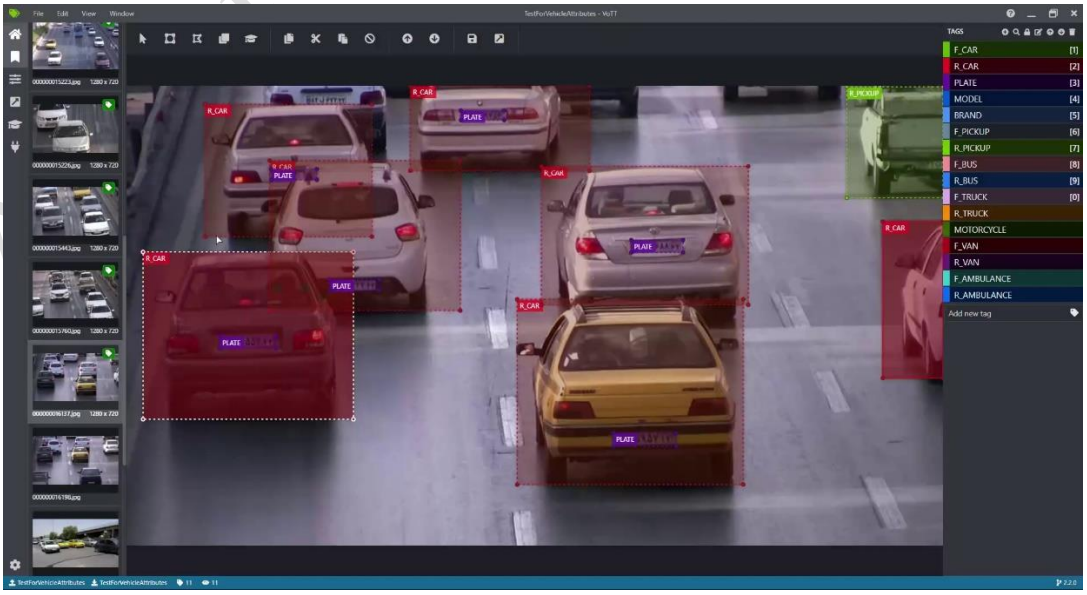

Figure 3. Labeling of images in IRVA dataset using VOTT software

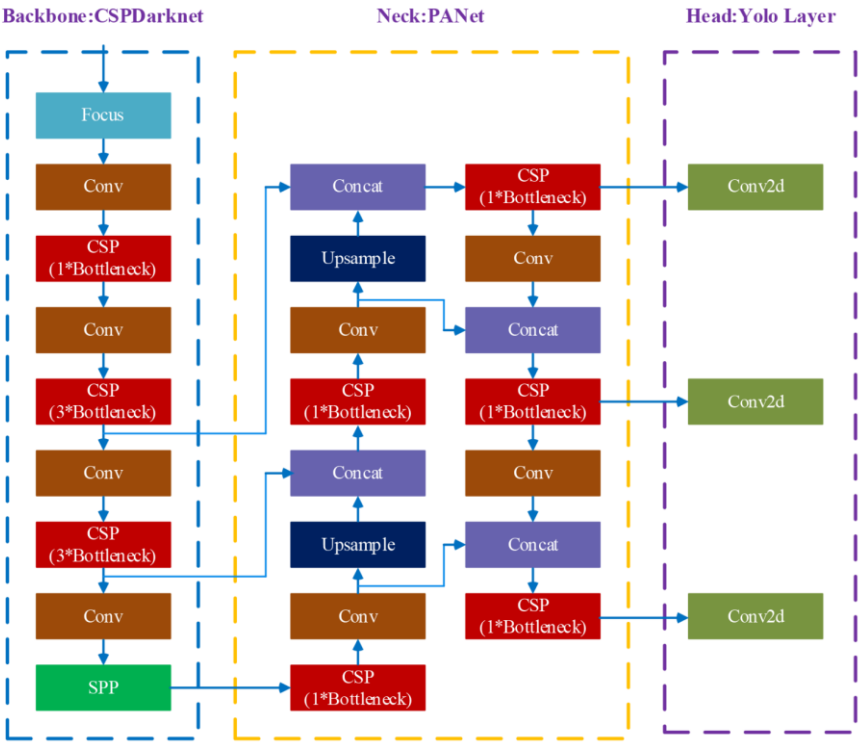

Figure 4. Abstract block diagram of YOLOv5 model [42]

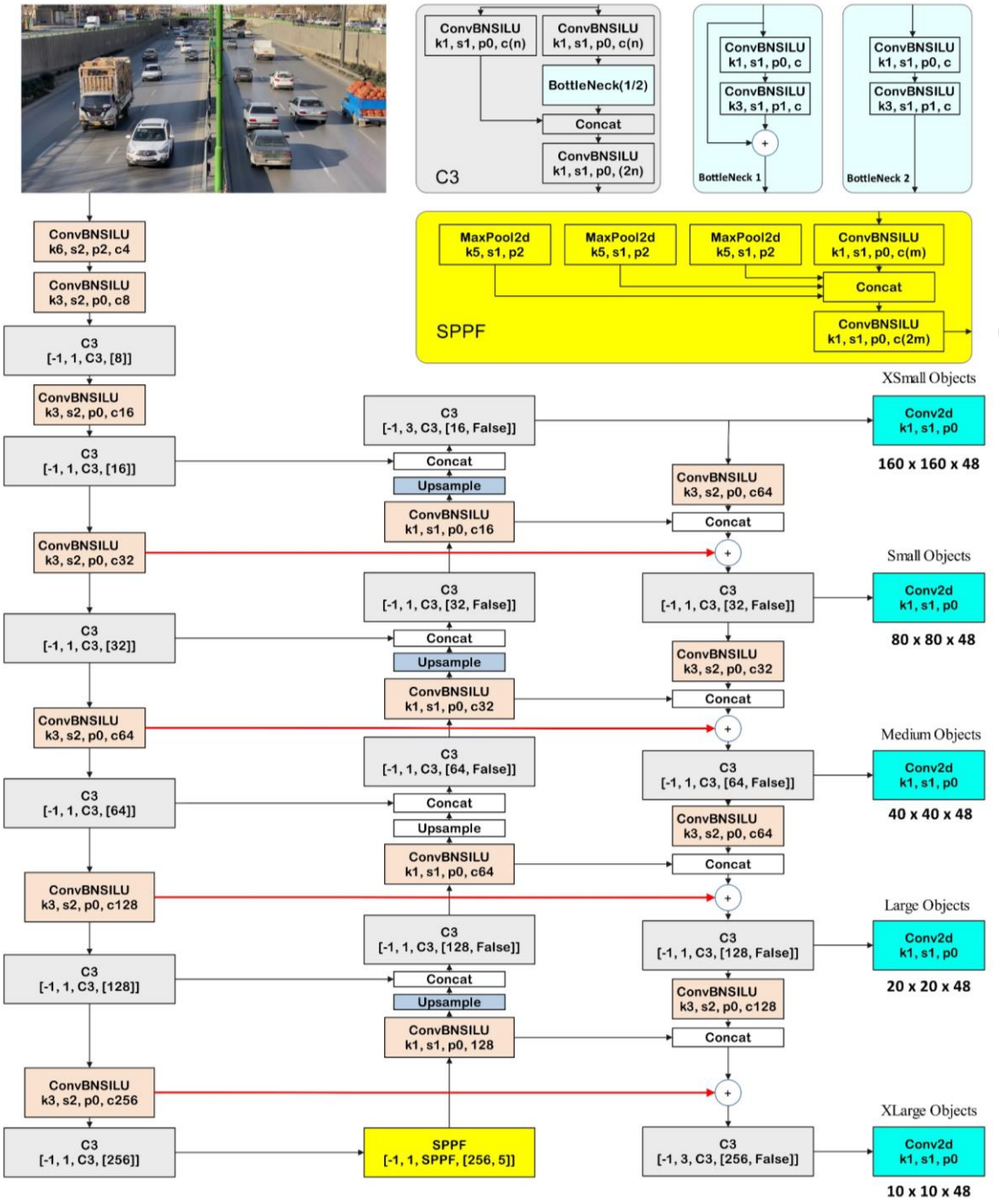

Figure 5. The structure of the proposed FYNet structure with fast inference time

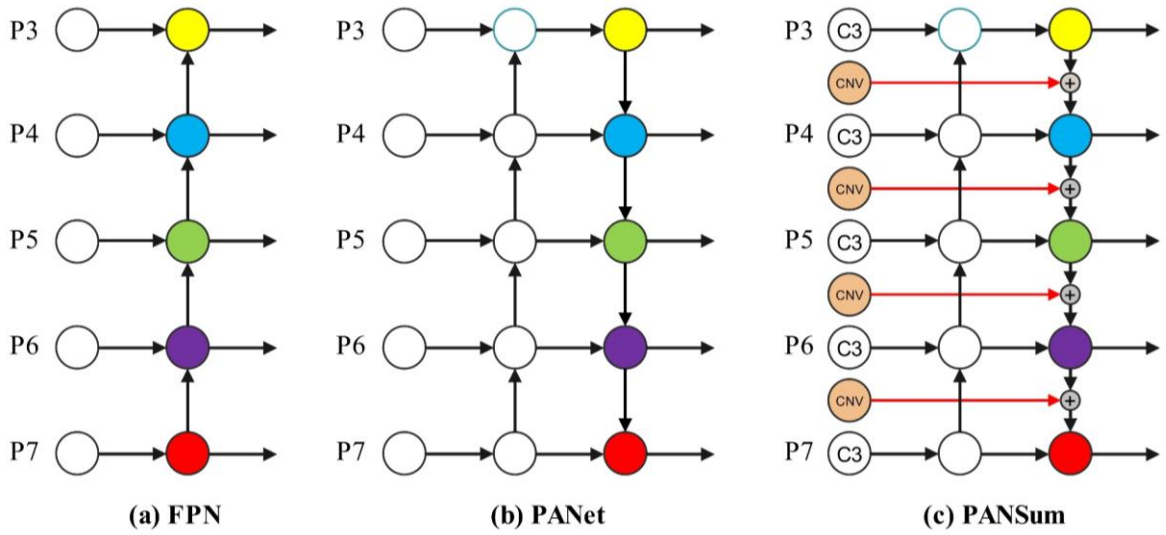

Figure 6. The structure FPN, PANet and PANSum blocks

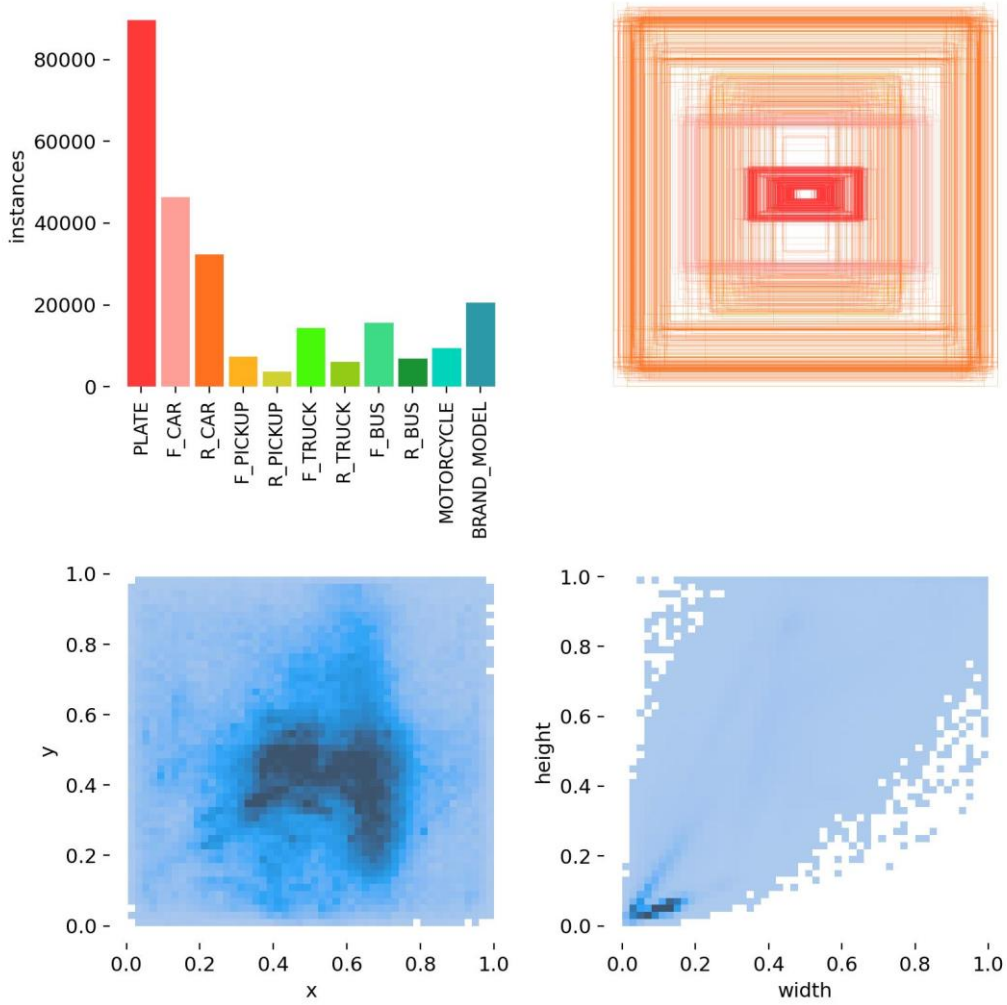

Figure 7. The chronogram graphs and the distribution of the labels in IRVA dataset

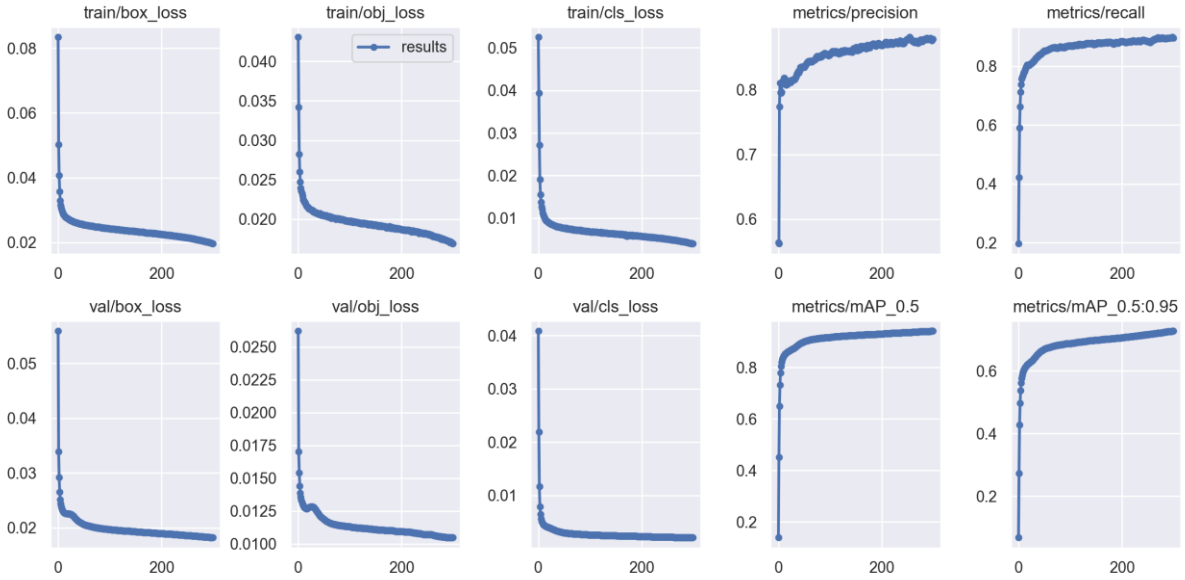

Figure 8. FYNet training graphs including mAP, precision, recall and train/val loss during 300 epochs

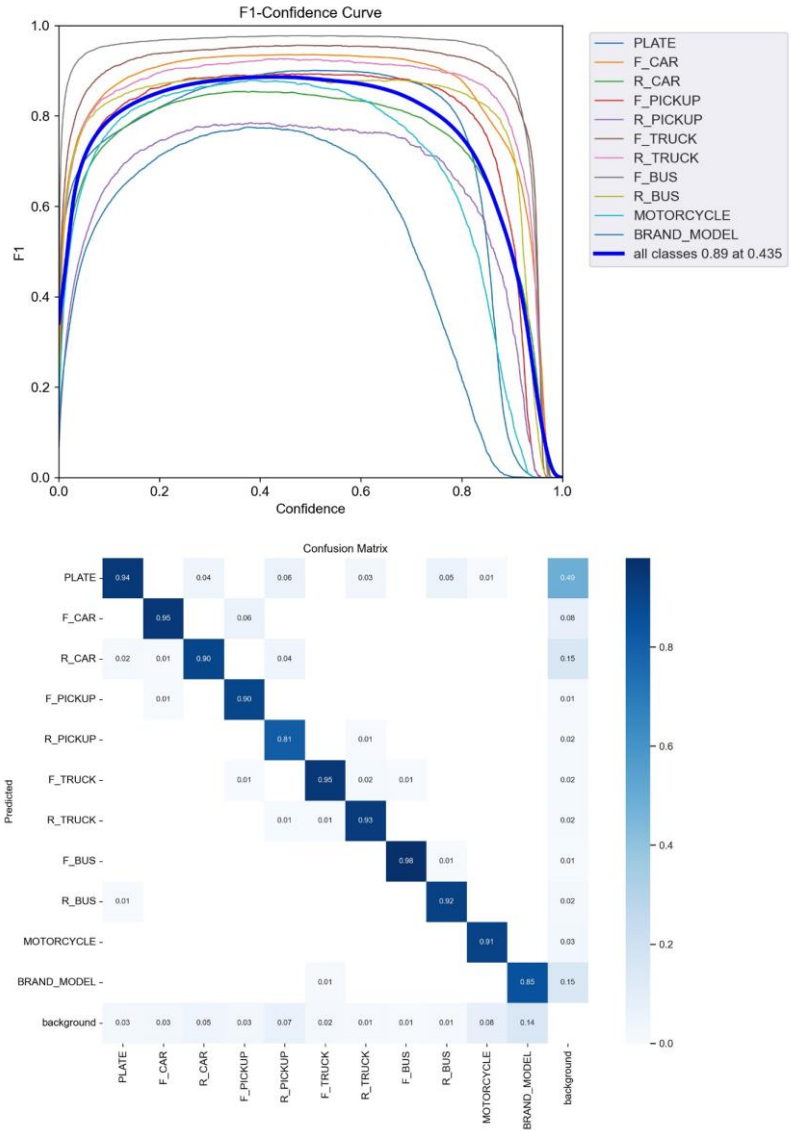

Figure 9. F1-Score graphs and Confusion Matrix after training the FYNet model

Figure 10. Output images of FYNet model on the highway

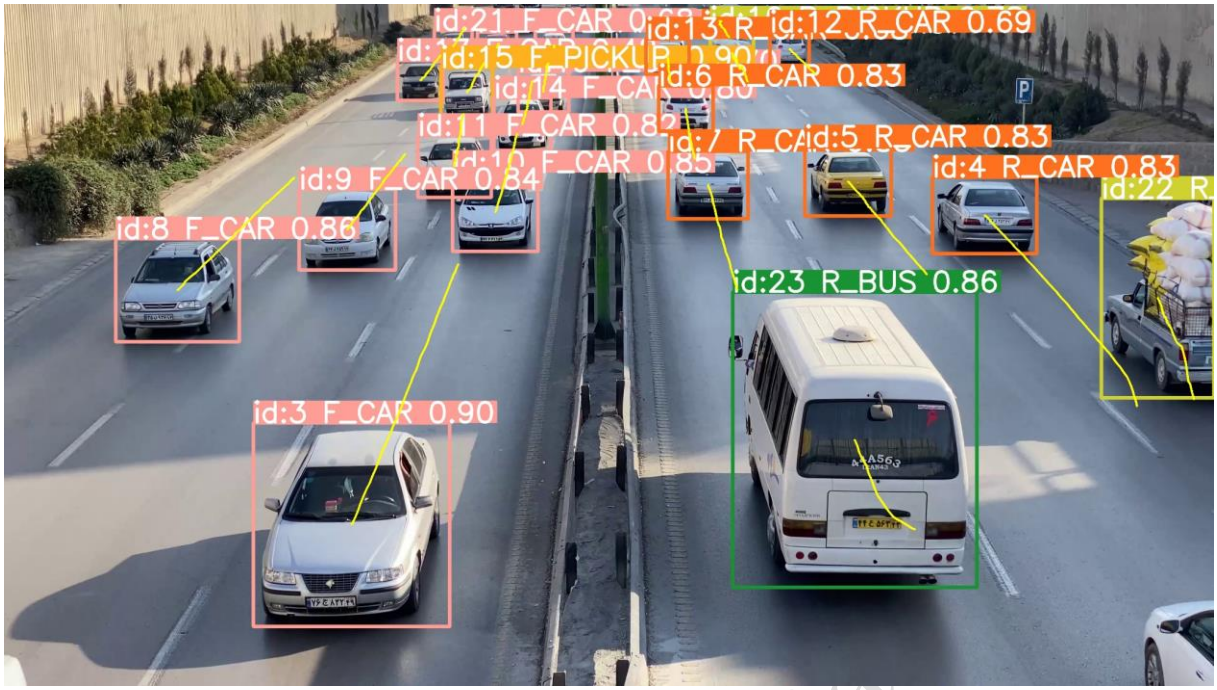

Figure 11. Tracking cars and assigning a unique ID to each car and visualizing the trajectory of each car on a six-lane highway.

Table 1. Abbreviations used in naming the IRVA dataset

Word	**Acronym**	**Word**	**Acronym**
Front	F	Ambulance	A
Rear	R	Bus	B
Front-Rear	FR	Van	V
Color	C	Truck	T
Grey	G	Pickup	P
Day	D	Motorcycle	M
Night	N	Car in Parking	CP
Day-Night	DN	Bus in Parking	BP
Highway	H	Bus in Highway	BH
Four-Way intersection	FW	Truck on Weighbridge	TW

Table 2. IRVA dataset statistics in day and night conditions

Class/Label	**Day**	**Night**	**Total**
Plate	99996	12002	111998
F-CAR	45803	12080	57883
R-CAR	36703	3465	40168
F_PICKUP	7994	1038	9032
R-PICKUP	4613	56	4369
F-TRUCK	16314	1416	17730
R-TRUCK	7310	127	7437
F-BUS	16293	2939	19232
R-BUS	7440	921	8379
MOTORCYCLE	11567	38	11605
BRAND_MODEL	25203	348	25551
Total label	278954	34430	313384
Image Count	95951	15302	111253

Table 3. IRVA dataset statistics for 11 classes of vehicle attributes

Class/Label	Train	Val	Test	Total
Plate	78453	16828	16717	111998
F-CAR	40385	8765	8733	57883
R-CAR	27975	6101	6092	40168
F_PICKUP	6373	1341	1318	9032
R-PICKUP	3052	651	666	4369
F-TRUCK	12433	2646	2651	17730
R-TRUCK	5190	1129	1118	7437
F-BUS	13479	2904	2922	19305
R-BUS	5929	1228	1222	8379
MOTORCYCLE	8155	1739	1711	11605
BRAND_MODEL	17935	3771	3845	25551
Total label	219319	47103	46962	313384
Image Count	77912	16661	16680	111253

Table 4. Metrics of YOLOv5n when front and rear views have separated classes

Class	Samples	Precision	Recall	mAP50	mAP50-95
PLATE	9181	0.861	0.900	0.956	0.667
F-CAR	4588	0.924	0.923	0.967	0.805
R-CAR	3135	0.791	0.856	0.902	0.671
F_PICKUP	839	0.841	0.864	0.889	0.713
R-PICKUP	395	0.727	0.722	0.747	0.513
F-TRUCK	1611	0.952	0.944	0.978	0.856
R-TRUCK	704	0.846	0.919	0.953	0.806
F-BUS	1647	0.964	0.971	0.991	0.845
R-BUS	742	0.785	0.941	0.943	0.77
MOTORCYCLE	1138	0.891	0.805	0.904	0.528
BRAND_MODEL	2379	0.744	0.669	0.767	0.412
Total	26359	**0.848**	**0.865**	**0.909**	**0.689**

Table 5. Metrics of YOLOv5n when front and rear views are in the same class

Class	Samples	Precision	Recall	mAP50	mAP50-95
PLATE	9181	0.853	0.912	0.958	0.674
CAR	7723	0.876	0.901	0.951	0.670
PICKUP	1234	0.823	0.822	0.868	0.762
TRUCK	2315	0.921	0.953	0.974	0.667
BUS	2389	0.906	0.966	0.983	0.849
MOTORCYCLE	1138	0.875	0.808	0.904	0.529
BRAND_MODEL	2379	0.713	0.720	0.767	0.415
Total	26359	**0.842**	**0.854**	**0.905**	**0.671**

Table 6. The network structure of the proposed FYNet model

Module	Params	C_{in}	C_{out}	Kernel	Stride	Padding
ConvBNSILU	880	3	8	6	2	2
ConvBNSILU	592	8	8	3	2	0
C3	336	8	8	-	-	-
ConvBNSILU	1184	8	16	3	2	0
C3	1248	16	16	-	-	-
ConvBNSILU	4672	16	32	3	2	0
C3	4800	32	32	-	-	-
ConvBNSILU	18560	32	64	3	2	0
C3	18816	64	64	-	-	-
ConvBNSILU	73984	64	128	3	2	0
C3	74496	128	128	-	-	-
ConvBNSILU	295424	128	256	3	2	0
C3	296448	256	256	-	-	-
SPPF	164608	256	256	-	-	-
ConvBNSILU	33024	256	128	1	1	0
Upsample	0	-	-	-	-	-
Concat	0	-	-	-	-	-
C3	90880	256	128	-	-	-
ConvBNSILU	8320	128	64	1	1	0
Upsample	0	-	-	-	-	-
Concat	0	-	-	-	-	-
C3	22912	128	64	-	-	-
ConvBNSILU	2112	64	32	1	1	0
Upsample	0	-	-	-	-	-
Concat	0	-	-	-	-	-
C3	5824	64	32	-	-	-
ConvBNSILU	544	32	16	1	1	0
Upsample	0	-	-	-	-	-
Concat	0	-	-	-	-	-
C3	1504	32	16	-	-	-
ConvBNSILU	2336	16	16	3	2	0
Concat	0	-	-	-	-	-
Sum	0	-	-	-	-	-
C3	4800	32	32	-	-	-
ConvBNSILU	9280	32	32	3	2	0
Concat	0	-	-	-	-	-
Sum	0	-	-	-	-	-
C3	18816	64	64	-	-	-
ConvBNSILU	36992	64	64	3	2	0
Concat	0	-	-	-	-	-
Sum	0	-	-	-	-	-
C3	74496	128	128	-	-	-
ConvBNSILU	147712	128	128	3	2	0

Concat	0	-	-	-	-	-
Sum	0	-	-	-	-	-
C3	296448	256	256	-	-	-
Detect	24048	-	-	-	-	-

Table 7. Hyperparameters used to train the FYNet on IRVA dataset

Parameter	Value	Description
epoch	300	The number of epochs to train the model
Batch size	9	Batch size in each iteration
Image size	1280	Sample image size
lr0	0.01	Initial learning rate (SGD=1E-2, Adam=1E-3)
lrf	0.01	Final learning rate
momentum	0.937	SGD momentum/Adam beta1
weight_decay	0.0005	optimizer weight decay
warmup_epochs	3	warmup epochs
warmup_momentum	0.8	warmup initial momentum
warmup_bias_lr	0.1	warmup initial bias lr
box	0.05	box loss gain
cls	0.5	cls loss gain
cls_pw	1	cls BCELoss positive_weight
obj	1	obj loss gain (scale with pixels)
obj_pw	1	obj BCELoss positive_weight
iou_t	0.2	IoU training threshold
anchor_t	4	anchor-multiple threshold
anchors	3	anchors per output layer
fl_gamma	0	focal loss gamma
hsv_h	0.015	image HSV-Hue augmentation (fraction)
hsv_s	0.7	image HSV-Saturation augmentation (fraction)
hsv_v	0.4	image HSV-Value augmentation (fraction)
degrees	0	image rotation (+/- deg)
translate	0.1	image translation (+/- fraction)
scale	0.5	image scale (+/- gain)
shear	0	image shear (+/- deg)
perspective	0	image perspective (+/- fraction)
flipud	0	image flip up-down (probability)
fliplr	0.5	image flip left-right (probability)
mosaic	1	image mosaic (probability)
mixup	0	image mixup (probability)
copy_paste	0	segment copy-paste (probability)

Table 8. Comparison of different types of YOLOv5 model on COCO dataset

Model	Size (pixels)	mAP 50	mAP 50-95	Params (M)	FLOPs (B)	Speed V100 (ms)	Speed CPU(ms)
YOLOv5n6	1280	54.4	36.0	3.2	4.6	8.1	153
YOLOv5s6	1280	63.7	44.8	12.6	16.8	8.2	385
YOLOv5m6	1280	69.3	51.3	35.7	50.0	11.1	887
YOLOv5l6	1280	71.3	53.7	76.8	111.4	15.8	1784
YOLOv5x6	1280	72.7	55.0	140.7	209.8	26.2	3136

Table 9. Comparison of the proposed FYNet model with the YOLOv5p6 model

Model	Size (pixels)	Layer (num)	mAP 50	mAP 50-95	Params (M)	FLOPs (B)	Inference (GPU-ms)	Inference (CPU-ms)	NMS (ms)
YOLOv5p6	640	206	-	-	0.786	1.2	6.4	25.42	1.0
FYNet(ours)	640	244	-	-	1.7	**0.6**	**6.3**	**17.78**	**0.9**
YOLOv5p6	1280	206	93.7	71.9	0.786	1.2	6.43	84.38	1.0
FYNet(ours)	1280	244	**94.0**	**72.7**	1.7	**0.6**	**6.28**	**59.71**	**0.9**
YOLOv5p6	1920	206	-	-	0.786	1.2	7.65	167.6	1.1
FYNet(ours)	1920	244	-	-	1.7	**0.6**	**7.55**	**123.3**	**0.9**
YOLOv5p6	3840	206	-	-	0.786	1.2	21.3	841.81	1.1
FYNet(ours)	3840	244	-	-	1.7	**0.6**	**15.43**	**555.1**8	**0.9**
Improvement (%)	-	-	**0.3**	**0.8**	-	**50%**	**38%**	**35%**	**18%**

Table 10. Hardware Specification

Configuration	**Model**
CPU	Intel i5-11400
GPU	NVIDIA GeForce RTX3060 12GB
RAM	32GB
CUDA	11.7
PyTorch	2.0.1
Python	3.11.5

Table 11. The mAP, recall, and precision of the FYNet on the IRVA dataset

Class	**Instances**	**P**	**R**	**mAP50**	**mAP50-95**
PLATE	16828	0.858	0.935	0.967	0.682
F-CAR	8765	0.935	0.934	0.975	0.812
R-CAR	6101	0.836	0.87	0.933	0.718
F_PICKUP	1341	0.892	0.892	0.933	0.759
R-PICKUP	651	0.78	0.783	0.858	0.638
F-TRUCK	2646	0.958	0.951	0.986	0.863
R-TRUCK	1129	0.894	0.961	0.986	0.838
F-BUS	2904	0.973	0.981	0.992	0.878

R-BUS	1228	0.845	0.94	0.961	0.798
MOTORCYCLE	1739	0.897	0.857	0.931	0.573
BRAND_MODEL	3771	0.792	0.754	0.832	0.441
All classes	47103	**0.878**	**0.896**	**0.94**	**0.727**

Hossein Khosravi earned his Bachelor's degree in Electronic Engineering from Sharif University of Technology in 2003. He went on to complete his Master's and Ph.D. degrees in Machine Learning at Tarbiat Modares University in 2005 and 2009, respectively. Since 2009, he has served as a faculty member at the Department of Electrical Engineering, Shahrood University of Technology, where he is actively involved in both teaching and research. In addition to his academic role, he is the CEO of Shahaab Co. (Shahaab-co.com), a technology company specializing in intelligent systems and automation solutions. His research interests span deep learning and image processing.

Seyyed Aliakbar Hosseini is currently pursuing a Ph.D. in Electronic Engineering at Shahrood University of Technology, Iran, with a research focus on deep learning applications in Intelligent Transportation Systems (ITS). He earned his M.Sc. in Mechatronics Engineering from the South Tehran Branch of Islamic Azad University in 2012. His Ph.D. thesis centers on the development of real-time computer vision algorithms for traffic analysis, with an emphasis on optimizing deep neural networks for vehicle detection and attribute recognition in high-resolution highway environments.