A COMPUTER VISION FRAME WORK FOR TRACKING & DETECT VEHICLES FROM ROAD SURVEILLANCE VIDEO

Vinod Kumar Yadav[1], Dr. Pritaj Yadav[2] and Dr. Shailja Sharma[3]

1 *Research Scholar, Computer Science & Engineering Department, RNTU, Bhopal, M.P., India*
[2] *Associate Professor, Computer Science & Engineering Department, RNTU, Bhopal, M.P., India*
[3]*Professor, Computer Science & Engineering Department, RNTU, Bhopal, M.P., India*

Abstract: *Use a deep learning-based computer vision method for vehicle detection and tracking, plays an integral role in intelligent transportation systems and traffic accident detection. Road Vehicle detection, recognition, and counting are an important part of road traffic monitoring and surveillance. Designing a road monitoring model with good performance is very complex. Traditional vehicle detection capabilities based on artificial intelligence algorithms have poor detection capabilities and robustness. This paper proposes an extended efficient single stage detection technique that is based on the YOLO to perform vehicle detection and multi-object tracking methods for tracking and counting vehicles that are based on a computer vision approach. The proposed deep learning model takes a moving vehicle's video, proposed detection technique vehicles detect and create a unique id for each of the initial detection after then tracking each of the vehicles as they move around frames in a given video, maintaining the id assignment. The proposed detection algorithm has great adaptability when viewed under a variety of conditions, such as heavy traffic, night environment, multiple vehicles overlapping, and part of a vehicle missing. Experimental results show that the algorithm can accurately detect and recognize vehicles according to their edge contours.*

***Keywords*: Yolo, Deep Sort, Deep Learning, Computer vision, Tracking, CNN.**

1. INTRODUCTION

In recent years, the number of vehicles has risen on the roads, and the traffic density has increased on the current roads, which is also caused by the frequent incidence of road traffic accidents. In order to ensure the safety of peoples' lives and property, the basic principle of vehicle detection, recognition, and tracking is proposed. [1]. It is a very significant and challenging task to detect, identify and track moving vehicles via video, real-time monitoring, and other data. For various complex situations in traffic monitoring on different roads and driving conditions, real-time monitoring of vehicles is very important and challenging [2]. The difficulty of this task lies in the accurate positioning and recognition of all vehicles in complex scenes and vehicle segmentation and tracking. When the road surveillance camera captures photos of the front and rear of the vehicle, it is sometimes unable to accurately give whether the front or rear of the vehicle is captured. When two vehicles pass by at the same time, the captured picture includes the head of one vehicle and the tail of another vehicle [3]. When the head and tail cannot be distinguished, it is generally based on the conventional left-in-right-out principle or user-defined rules to distinguish the license plate number of the head or tail that needs to be recognized at present [4]. However, the system cannot distinguish between the head and tail, causing more false positives. In the previous work, we used the convolution neural network method to directly identify the head and tail of extracted vehicles in the monitoring image. The recognition rate is not ideal, so the work of this paper is to solve or partially solve the problem that the recognition accuracy of the head and tail is not high [5]. To solve these problems, a convolution neural network method based on an extended efficient single stage yolov5 technique is proposed to detect the vehicles in the video. This work is conducive to the identification of the front and rear of the vehicle in the later stage. After the vehicle is detected, they track each of the vehicles as they move around frames in a given video, maintaining the id assignment. In tracking an element-based object, it extracts the features of the object from a single frame and compares the appearance information with consecutive frames based on the similarity scale. Easy Online and Real-Time Tracking with the Deep Association (Deep-SORT) metric makes it possible to track multiple items by combining appearance details with its tracking components [6].

2. LITERATURE SURVEY

Before than 2014, the main detection methods are the background difference method [7], the Inter-frame difference method [8], and the optical flow method [9]. After extensive use of deep learning methods, the YOLO algorithm proposed by Redmon et al. in 2015 he takes the whole image as the input of the network and directly obtains the output of the bounding box and detection result [10]. This method can detect 45 frames per second, which is faster than other algorithms. The YOLO algorithm no longer moves the window, but divides the original image directly into small non-conforming squares and then deforms and creates a map of objects of that size. Based on the above analysis it can be concluded that each element of the feature map is also a small square corresponding to the original image [11]. Each element can then be used to predict the target of those central points in the small square. There are three main improvements in YOLOv2 compared to YOLOv1: batch normalization; Refine the classification model using high-resolution images; Using an a priori box YOLO2 started using K-means clustering to obtain a priori box dimensions, and YOLO3 continues this method by setting three priority fields for each down sampling scale, resulting in a total of nine a priori sizes the box is clustered. The main improvements of YOLO3 are: modification of the network structure; Multilevel function is used for object detection [12]. Object classification replaces soft-max with Logistic classification [13]. The top-down PAN feature fusion method was first proposed in the YOLOv4 [14] paper. A model with most parameters in YOLOv5 uses the Focus module before inserting the image tensor into the backbone Sew. Focus - Subnetwork Sampling; SPP: function fusion; PAN: Fusion of functions from top to bottom. Although the authors have not put YOLOv5 into a direct test comparison with YOLOv4, the test results on the COCO dataset are impressive. In summary, YOLOv5 is not only very fast, but also very lightweight with a model size of, and it matches the YOLOv4 benchmark in terms of accuracy. Currently, prior to wheat testing on Kaggle, most participants are using his YOLOv5 framework. In general, both YOLOv4 and YOLOv5 have good accuracy in real-world testing and training, but the different network structure of YOLOv5 makes users more flexible. According to the needs of different projects, it integrates the strengths and weaknesses of YOLOv4 and YOLOv5 and in different detection, networks can take full advantage of the detection algorithm.

3. VEHICLE DETECTION AND TRACKING ARCHITECTURE

YOLOv5s is the network with the smallest feature map depth and width, and the other three are considered to be deepened and extended to base on it. YOLOv4 and his YOLOv5 are similar in structure but differ in some details. YOLOv5 (You only look once can process 140 frames per second for real-time video image recognition and has a smaller structure. The YOLOv5 version weight data files are 1/9th smaller than YOLOv4, which is 27 MB in size. YOLOv5 is divided into YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5 according to the network depth size and feature map width. YOLOv5s is used as the usage model in this paper. If the backbone uses mobilenetv3 small, the design of the channel-separable convolution actually increases the training speed. It takes about 6 hours to complete training for 300 epochs with yolov5s on my device, but only about 4.5 hours on mobilenetv3 small. Mobilenetv3 small also has advantages in terms of network parameters.

3.1.Network structure of yolov5

The YOLOv5 network structure is divided into two parts: input side, backbone part, and detection network: neck and prediction part. The main procedural function used on the input side is Mosaic Data Enhancement. This paper mainly introduces random scaling, random clipping and random image placement for stitching data to achieve relatively significant experimental results in detecting small targets. In the initial training of YOLO version, all picture data will be processed with Mosaic data so that the result of the picture is 416*416 or 608*608 and then detected. This will lead to problems in data processing during the experiment, resulting in smaller, partially obscured and blurred targets on the picture that cannot be detected before, which makes the results of the experiment uneven. Adaptive anchor frame calculation means that in different training, we can adjust the anchor frame, pass in the required training set based on COCO, adjust the effect of turning on the adaptive anchor frame algorithm according to the results we need, and also include adaptive picture zooming, zooming the picture to a uniform size, making it easier for the system to quantify processing and quick extraction of information. The main purpose of the Backbone structure is to enhance the learning ability of the convolution network and reduce the budget costs: Focus structure, CSP structure. Neck: The FPN+PAN structure is mainly used to adjust the number of layers to convey flat features and reduce the risk of data loss. Prediction uses CIOU_Loss can also be set to use

other IOU_Loss if desired. Loss use and closure (GIOU_Loss=False, DIOU_Loss=False, CIOU_Loss=False) [15]. YOLOv5 has a shorter network structure and deeper properties, resulting in faster operations. By analyzing various properties of the training parameters, the training map of the neural network model is approximated, greatly improving the convergence speed of the network. Of course, supervised learning in machine learning can be described by function approximation.

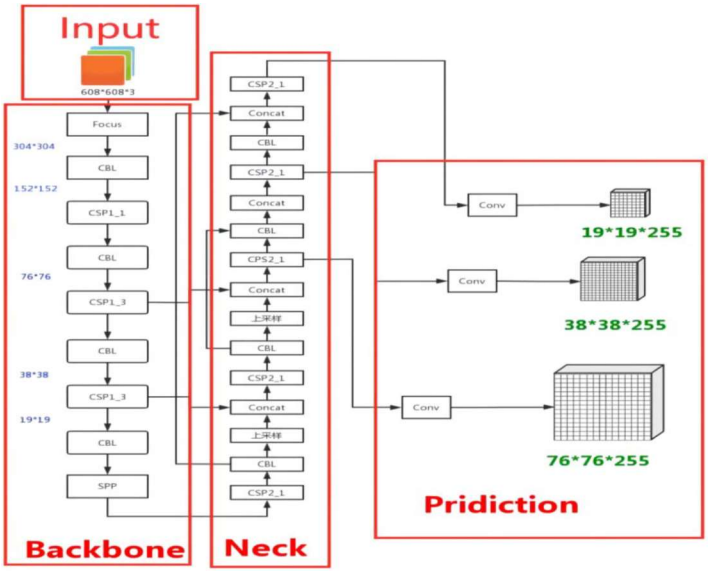

Figure 1: YOLOv5s Network structure.

It tries to approximate the properties represented by the data in order to minimize the error during training by calculating the error between predicted and expected outputs. Function approximation is a useful tool only when the underlying target mapping is unknown. Function approximation using neural networks because they are universal approximations. In theory you can use them to approximate any function. Since the basic structure of YOLOv5 is not yet complete, its creator has not published a scientific paper describing it, and the program code is constantly being updated. The basic network structure of yolov5 is shown in Figure 1.

3.1.1. Basic principles of Input side

The input side of mosaic data extension: Yolov5 uses the same mosaic data extension as Yolov4. The creator of the mosaic data improvement proposal, who is also a member of the Yolov5 team, uses the mosaic data improvement technique in the training model phase, which is an improvement of the cutmix data improvement technique. Cut-Mix stitches two images together, while Mosaic's data enrichment method takes four images and stitches them together according to random scaling, random clipping, and random ordering. This improved method combines multiple images into one. This not only enriches the dataset, but also greatly increases the training speed of the network and reduces the memory footprint of the model. The detection effect of small targets is greatly improved. The main advantage of this design is the enrichment of the dataset, the random use of four images, the random zooming, and the random stitching. This design greatly enhances the detection dataset. In particular, random zoom adds many small targets, making network more robust.

Adaptive anchor frame calculation: When training the network , the network outputs predicted boxes at based on the first anchor box, compares it with the actual box which is basically correct, computes the difference between them and puts them in Update in reverse and iterate over the network. Therefore, the initial anchor frame is also an important part, just like the YOLOv5 initial anchor frame in the COCO dataset [16]. Predefined Frames is a set of preset frames. During training, a training pattern is constructed with the actual offsets of the limit positions from the preset limits. Anchor boxes can be customized by border aspect ratio and border area (scale). To define a set of rules for generating preset borders, anchor boxes can be used to generate a set of borders anywhere in the image.

Anchor box does not need to specify a center position, as it usually creates a boundary at the center of point in the feature map extracted by CNN.

Adaptive picture scaling:
(a) Principle of adaptive image scaling: In the common target vehicle detection algorithm, the length and width of different frames and video frames are different. Therefore, to solve this kind of problem, a common way is to uniformly scale the original images to a standard size and send them to the detection network. Examples of sizes that are frequently utilized in yolov3, yolov4, and yolov5 algorithms include 416 * 416, 608 * 608, and others. [17]. You can see from the image that although proportional scaling has been done, you can still see that after scaling down the image and filling it, the sizes of the black borders are different at both ends. If the image is filled more, there is excess unnecessary information about the image, which greatly affects the running speed. Therefore, the letterbox function is modified in the YOLOv5 code to adaptively add as little black border as possible to the original image. Through this simple improvement, the speed of reasoning and the actual program are greatly improved the running effect is remarkable.

(b) Adaptive image scaling process:
(I) calculate the scale. For example, the length of the original movie length * width: 800 *600 and the original scale size is 416 * 416. After dividing by the size of the original image, you can get two scale factors of 0.52 and 0.69. Choose a small scale factor.
(II) Calculate the reduced size. Multiply the length and width of the original image by a small scale factor (800 * 0.52 = 416; 600 * 0.52 = 312) to get the reduced image size (416 * 312).
(III) Calculate the padding of the black border to be 416-312 = 104 to get the height to be filled initially. Then use np.mod the remainder in numpy to get 8 pixels and then divide by 2 to get the values to be filled at both ends image height. The size of the new image is 416 * 320 (312+8).

3.1.2. Back Bone

Both YOLOv5 a convolutional neural network that aggregates and forms image features with different image granularity, use CSP_Darknet as a backbone to extract rich information features from input images. (CSP_Net solves the problem of duplicating gradient information for network optimization in Backbone, a framework of other large convolutional neural networks. The gradient will change are integrated into the feature map end-to-end, reducing model parameters and FLOPS values, ensuring the speed and accuracy of reasoning and reducing the size of the model [20]. This can effectively mitigate the gradient vanishing problem (it is difficult to reverse the lossy signal over very deep networks), promote element propagation, promote reuse of network elements, and reduce the number of network parameters.

3.1.3. Neck

A series of network layers that interpolate and combine image features and transfer them to the Based prediction layer on Mask R-CNN and FPN framework PANET improves information propagation and has the ability to accurately preserve spatial information, which helps to correctly locate pixels to create a mask. YOLOv5 now uses FPN+PAN as in Neck and YOLOv4. FPN is top-down and uses an up-sampling method to transfer and combine information to obtain a predicted feature map and PAN uses a bottom-up function pyramid [18]. The difference between YOLOv5 and YOLOv4 is that the neck structure of yolov4 adopts a common convolution operation. In the neck structure of yolov5, the csp2 structure designed by cspnet is used to strengthen the fusion capability of network functions. yolov5's neck network still uses the FPN + pan structure, but some improvements have been made based on it. The following figure shows the specific details of the YOLOv4 and YOLOv5 neck nets. By comparison, we can find that yolov5 uses the csp2_1 structure, which replaces some CBL modules, and replaces the CBL module after the concate operation with the csp2_1 module.

3.2. Tracker–detector-integrated object tracking

In this work, considering navigation tasks in assistive platforms, an evaluation study of multi-object tracking by Deep-SORT using new data association metrics is proposed. Deep-SORT methods were proposed with a focus on

real-time object tracking tasks, both achieving state-of-the-art results with a high frame rate [19]. The SORT and Deep-SORT methods share the same overall architecture, divided into three main modules, as shown in Figure 2, The SORT method associates objects using bounding box detections to match measurements with predicted tracks, using the overlap of bounding boxes. On the other hand, to improve the bounding box association step, the Deep-SORT uses a CNN to extract appearance features from the object bounding box images.

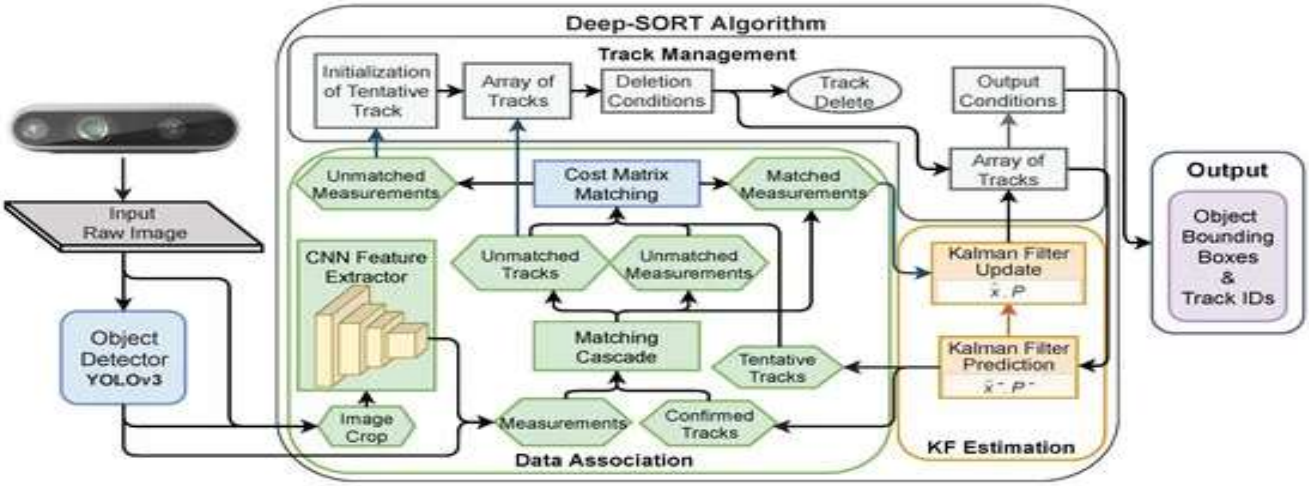

Figure 2: Overview of the vehicle tracking Deep-SORT algorithm.

4. RESULTS AND ANALYSIS

This paper mainly studies the YOLOv5s on the basis of the vehicle detection and recognition technology processing algorithm to do practical application work, such as training model creation, development, and testing interface creation, processing and preparation of model datasets, and training self-built datasets. Find the best parameters on the local dataset and test the data, and finally get the relevant results of the experiment, the general process of the experiment is shown in Figure 3.

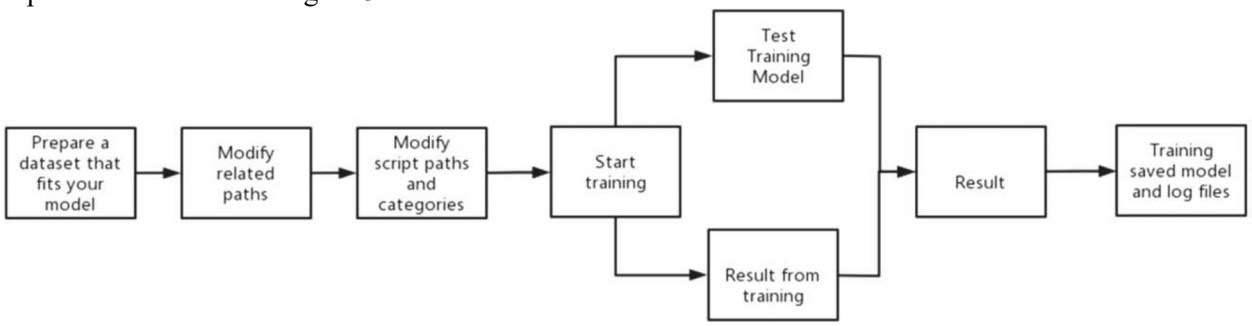

Figure 3: Training process

In this paper, the network structure and performance of the YOLOv5 network are tested using three training datasets, COCO128 dataset, COCO2017 dataset and custom ME_ COCO datasets. The reason for using the COCO128 dataset (including 128 images of different classes) is that it is uncertain whether the final result of the YOLOv5s network training model can be obtained with its own training set, so the network parameters and training scale are adjusted to achieve ideal experimental results. The COCO2017 dataset and the self-made me dataset are used to verify the accuracy of the test results to achieve high accuracy of vehicle identification and detection. This paper too describes a comparison with other complex networks to illustrate the significant improvements of our network model. Configuration is specific to. The yaml file format is saved and modified according to the specific trained information. The YOLOv5s configuration file format included in this document is shown in Figure 4 below.

```
# parameters
nc: 3  # number of classes
depth_multiple: 0.33  # model depth multiple
width_multiple: 0.50  # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, BottleneckCSP, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 9, BottleneckCSP, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, BottleneckCSP, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, BottleneckCSP, [1024, False]],  # 9
  ]
```

```
# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, BottleneckCSP, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, BottleneckCSP, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, BottleneckCSP, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, BottleneckCSP, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]
```

Figure 4: YOLOv5s file configuration

The first is the number of NC (number of classes) data set categories; the second parameter is depth_ a multiple that controls the depth of the model. Multiple that can be used to dynamically adjust the depth of the model. The third parameter is width_ multiple which controls the width of the model. Current output channel on each layer in the middle of the model = theoretical channel (parameter C2 for each layer) * width_ Multiple, which also serves as a dynamic adjustment of the model width. Yolov5 initialized nine anchors, Used in three Detect layers (three feature maps), grid_cell of each feature map has three anchors for prediction. The matching rules are as follows: the larger feature map, the further forward it is, the smaller the sampling rate and the smaller the perception field, so it is relatively possible to predict some smaller objects with a smaller scale, shrinking all the anchors assigned to it; the smaller the object map, the further back it is, the relative The higher the sampling rate of the original map, the larger the sensing field, so it is relatively possible to predict some larger-scale objects. With larger anchors assigned to them. This means you can use small object maps (object map) to detect large targets, or small targets on a large object map. The yolov5 backbone counts one row for each module and each row consists of four parameters. From represents the input of the current module from this layer, −1 represents the output from the previous layer, and number represents the theoretical and actual number of iterations of the current module. The depth_ Multiple parameter together specifies the depth of the mesh model; the module class name of the module through which the generic is removed. Find the corresponding classes in py and set up the mesh modularization; Args is a list, the parameters required for module generation, channel, kernel size, pitch, padding, bias etc. vary according to different layers during the mesh building process. Head is the same as Backbone and is constructed similarly to Backbone and consists of four parameters, i.e. from represents the output from this layer for the input of the current module and −1 represents the output from the previous layer. However, here is a list of where the input to this layer comes from the output connection of the layer. Number represents the theoretical number of repetitions of the current module, and the actual number of repetitions is determined by the depth above parameter. Multiple together determine the depth of the network model. The module class name of the module through which the generic is removed. Find the corresponding classes in py and set up the mesh modularization. Args is also a list, the parameters required to create the module, channel, kernel size, pitch, padding, distortion, etc.

4.1. Experimental results and analysis

The training results of the datasets are represented by labels as shown in Figure 5. To make the experimental results more intuitive, we group the datasets. It divides a data set into different classes or clusters according to specific criteria such as distance. Data objects in the same cluster are of the same type as much as possible, while data

objects that are not in the same cluster are quite different. First, standardize the features of the data and reduce the dimension, select the most effective features from the obtained features, store them in vectors, and measure the similarity based on different types of data. Finally, the distribution of data samples can be visually seen through the clustering results, which facilitates classification error analysis The first figure on the left shows the amount of data in the training set, the number of categories on the horizontal axis and the vertical axis represents the number of data frames detected in each category, which represents the sum of different vehicle detection results; The second one on the left shows the approximate vehicle position of the data set from the center and the distribution density distribution. Dense points are generated according to the position detected by the vehicle and integrated into this observation table; the third one on the left is the vehicle size distribution in the dataset.

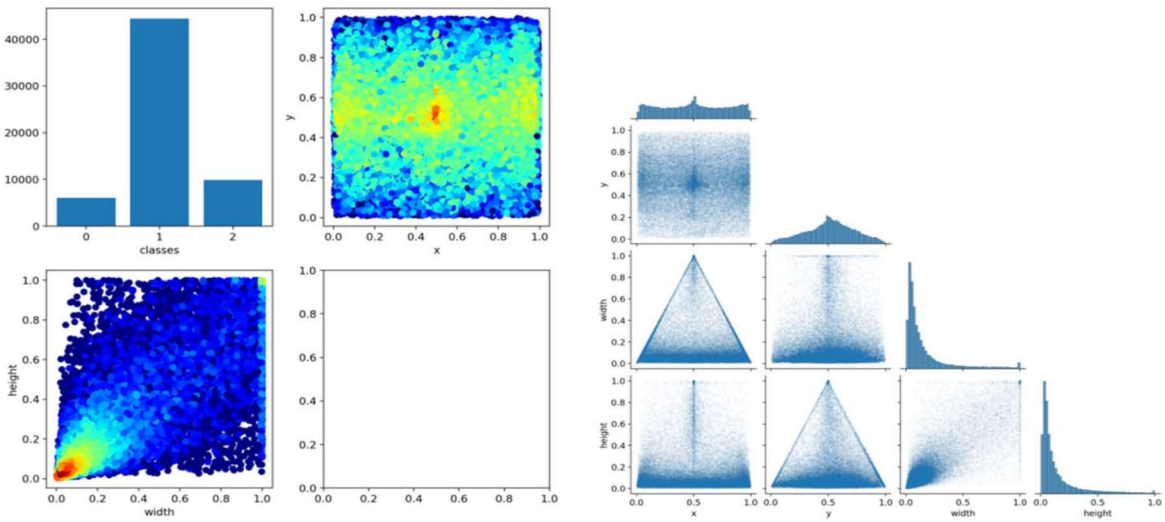

Figure 5: Distribution of training results of data set.

The larger the vehicle, the larger the width and height values and the closer the obtained dense points will be to this direction; The correct figure is a detailed statistic box diagram with markers. As in the image on the left, the macro results obtained in the image on the left are integrated into small ones points to create the same image to make the experimental data more clear and recognizable. At the same time, it can also display the approximate width and height of the vehicle in the data set as well as the position, image size etc. Box is GIOU_Loss used by YOLOv5 is the bounding box loss. The array is speculated to be the mean value function GIOU_Loss. The smaller the field, the more accurate the target detection. It can be obtained from visual data in the image. As time is subtracted on the line (x-axis, time_seconds), the mean value of the loss function also decreases and approaches the equilibrium point; Objectivity: speculated as the average target detection loss. The smaller the target, the more accurate the target detection; Classification: it is speculated as the average value of the classification loss. The smaller the value, the more accurate the classification; Accuracy: accuracy (positive classes found / all positive classes found); Recall: the true precision is positive, that is, how many positive samples were found (how many were recalled). Recall describes how many true positive examples in the test set are selected by two classifiers from the perspective of real results, that is, how many true positive examples are recalled by the two classifiers. Val box: bounding box loss validation set; Val objectless: mean target detection loss in the validation set; Val classification: the mean value of the classification loss of the validation set; The map is the area bounded after drawing with precision and recall as two axes, M represents the mean, @number after @ represents the threshold value to judge IOU as positive and negative samples, @0.5:0.95 represents the threshold value, and take the mean value after sampling 0.5:0.05:0.95. mAP@0.5: 0.95 represents the average map at different IOU thresholds (from 0.5 to 0.95, step 0.05) (0.5, 0.55, 0.6, 0.65, 0 .7, 0.75, 0.8, 0.85, 0.9, 0.95); mAP@0.5: indicates an average map with a threshold greater than 0.5.

5. CONCLUSION AND FUTURE WORK

In this paper, the vehicle detection method is based on yolov5s image recognition and processing technology is used to successfully realize tracking detection and vehicle counting recognition. From training As a result, the algorithm has high recognition speed and recognition speed in complex and bad environment weather. YOLOv5s

not only runs very fast, but also significantly reduces the storage space of the model. I believe, that with in-depth research, further improving the accuracy of the program and identifying other types of objects will be of great help to popularize artificial intelligence and promote the development of smart cities in the future. Currently, there is little research and innovation about yolov5 in relevant academic papers that requires us to calm down, to explore and perfect more and better methods, flexibly use them according to different scenarios and project needs, learn from each other and give full play to yolov5's fast, efficient and accurate tools detection advantages.

REFERENCES

[1]. Zhao, Z.-Q., Huang, D. S., & Sun, B.-Y. "Human face recognition based on multiple features using neural networks committee". Pattern Recognition Letters (2004), 25(12), 1351–1358.

[2]. Wang, X.-F., Huang, D. S., & Xu, H. "An efficient local Chan-Vese model for image segmentation". Pattern Recognition (2010). 43(3), 603–618.

[3]. Hu, R.-X., Jia, W., Ling, H., & Huang, D. S."Multi-scale distance matrix for fast plant leaf recognition". IEEE Transactions on Image Processing (2012), 21(11), 4667–4672.

[4]. Liang, X., Wu, D., & Huang, D. S. "Image co-segmentation via locally biased discriminative clustering". IEEE Transactions on Knowledge and Data Engineering (2019), 31(11), 2228–2233.

[5]. Wu, D., Wang, C., Wu, Y., Huang, DS, et al. "Attention deep model with multi-scale deep supervision for person re-identification". IEEE Transactions on Emerging Topics in Computational Intelligence (2021). 5(1), 70–78.

[6]. L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," arXiv preprint arXiv: 1511.04136, 2015.

[7]. Xin, S., Zhenkang, S., & Ping, W. P. "Application of mean shift in target tracking". Systems Engineering and Electronic Technology (2007). 29(9), 1405–1409.

[8]. Jifeng, S., & Chengqing, W. "Moving vehicle tracking based on feature point optical flow and Kalman filter". Journal of South China University of Technology (Natural Science Edition) (2013).

[9]. Mei, Y., Gangyi, J., & Bokang, Y. "Application of computer vision technology in intelligent transportation system". Computer Engineering and Application (2001). 190(9), 396–397.

[10]. Sun, Z.-L., Huang, D. S., & Cheung, Y.-M. "Extracting nonlinear features for multispectral images by FCMC and KPCA. Digital Signal Processing (2005). 15(4), 331–346.

[11]. Huang, D. S., & Du, J.-X. "A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks". IEEE Transactions on Neural Networks (2008). 19(12), 2099–2115.

[12]. Chen, W. S., Yuen, P. C., Huang, J., & Dai, D. Q. "Kernel machine-based one-parameter regularized fisher discriminant method for face recognition". IEEE Transactions on Systems, Manand Cybernetics, PartB (Cybernetics) (2005). 35(4), 659–669.

[13]. Zhao, Y., & Huang, D. S. "Completed local binary count for rotation invariant texture classification". IEEE Transactions on Image Processing (2012)., 21(10), 4492–4497

[14]. Linlin, M., Jianxin, M., Jiafang, H., & Yadi, L. "Research on target detection algorithm based on yolov5s". Computer Knowledge and Technology. (2021), 17(23), 100–103.

[15]. Kaijie Zhang, Chao Wang, Xiaoyong Yu, Aihua Zheng, Mingyue Gao, Zhenggao Pan, Guolong Chen & Zhiqi Shen. "Research on mine vehicle tracking and detection technology based on YOLOv5". Systems Science & Control Engineering (2022), 10:1, 347-366, DOI: 10.1080/21642583.2022.2057370.

[16]. Xiang, Y.; Zhao, B.; Zhao, K.; Wu L.; Wang X. "Improved Dual Attention for Anchor-Free Object Detection". Sensors 2022, 22, 4971. https://doi.org/10.3390/s22134971.

[17]. Jeonghun Lee, Kwang‑il Hwang1. "YOLO with adaptive frame control for real‑time object detection applications". Multimedia Tools and Applications (2022) 81:36375–36396.

[18]. Jinnan Lu and Yang Liu. "A real-time and accurate detection approach for bucket teeth falling off based on improved YOLOX". Mech. Sci., 13, 979–990, 2022. https://doi.org/10.5194/ms-13-979-2022.

[19]. iashuai Dai,Ling Wu and Pikun Wang. "A Video-Based Real-Time Tracking Method for Multiple UAVs in Foggy Weather". Electronics 2022, 11(21), 3576; https://doi.org/10.3390/electronics11213576.

[20]. Kaijie Zhang, Chao Wang, Xiaoyong Yu, Aihua Zheng, Mingyue Gao, Zhenggao Pan, Guolong Chen & Zhiqi Shen.. "Research on mine vehicle tracking and detection technology based on YOLOv5". Systems Science & Control Engineering, 10:1, 347-366, DOI: 10.1080/21642583.2022.2057370.