Research on mine vehicle tracking and detection technology based on YOLOv5

Kaijie Zhang a, b, Chao Wang a, b, Xiaoyong Yu a, Aihua Zheng c, Mingyue Gao a, Zhenggao Pan a, Guolong Chen d and Zhiqi Shen e

a School of Informatics and Engineering, Suzhou University, Suzhou, People's Republic of China; b Institute of Machine Learning and Systems Biology, School of Electronics and Information Engineering, Tongji University, Shanghai, People's Republic of China; c Information Materials and Intelligent Sensing Laboratory of Anhui Province, Anhui University, Hefei, People's Republic of China; d School of Computer Science and Information Engineering, Bengbu University, Bengbu, People's Republic of China; e School of Computer Science and Engineering, Nanyang Technological University, Nanyang Avenue, Singapore

ABSTRACT
Vehicle tracking detection, recognition and counting is an important part of vehicle analysis. Designing such a model with excellent performance is difficult. The traditional target detection algorithm based on artificial features has poor generalization ability and robustness. In order to take use the deep learning method for vehicle tracking detection, recognition and counting, this paper proposes a vehicle detection method based on yolov5. This method uses the deep learning technology, takes the running vehicles video as the research object, analysis the target detection algorithm, proposes a vehicle detection framework and platform. The relevant detection algorithm of the platform designed in this paper has great adaptability, when displayed under various conditions, such as heavy traffic, night environment, multiple vehicles overlap with each other, partial loss of vehicles, etc. it has good performance. The experimental results show that the algorithm can accurately segment and identify vehicles according to the edge contour of vehicles. It can take use the materials includes pictures, videos, and real-time monitoring, and has a high recognition rate in real-time performance.

1. Introduction
In recent years, with the accelerating process of China's modernization, the number of cars has increased sharply, and the traffic pressure of the original roads has increased, which also leads to the frequent occurrence of road traffic accidents 2009. In order to ensure the safety of citizens' lives and property, the basic principle of target vehicle detection, recognition and tracking is proposed (Zhao et al., 2004). It is a very important and challenging task to detect, identify and track stationary and moving vehicles in image, video, real-time monitoring and other data. For various complex situations in traffic monitoring under different roads and driving conditions, real-time monitoring of vehicles is very important and challenging (Wang et al., 2010). The difficulty of this task lies in the accurate positioning and recognition of all vehicles in complex scenes and vehicle segmentation and tracking (Zhang et al., 2006).

In the complex environment of the mine, some transport vehicles may escape after pulling the resources, that is, these vehicles escape the management of the monitoring platform after pulling the minerals, because the national tax on the resources in the mining area is calculated according to the mining volume, which will lead to tax evasion. In addition, the license plate recognition of the transport vehicles entering and leaving the mining area can effectively weigh the resources pulled by the transport vehicles and prevent the vehicles from not weighing or overloading, which is of great significance for the effective monitoring of resources and avoiding the loss of national taxes. Therefore, the fine identification of the license plate, front and rear of the transport vehicle in the mining area and the shape of the minerals carried is not only of great significance for the effective management of mine resources but also to avoid the illegal act of tax evasion (Laihu, 2012).

When the mine environmental supervision system captures the 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 (Hu et al., 2012). 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 (Liang et al., 2019). However, the system cannot distinguish the head and tail, Causing more false positives. In the previous work, we use the convolution neural network method to directly identify the head and tail of mine 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 mine head and tail is not high (Wu et al., 2021). To solve these problems, a convolution neural network method based on yolov5 deep learning is proposed to monitor and frame the vehicles in the monitoring pictures. This work is conducive to the identification of the front and rear of the vehicle in the later stage. Therefore, this paper mainly focuses on the detection of mining vehicles (Wang et al., 2021).

2. Related works

Vehicle detection technology, the main methods before 2014 are background difference method (Xin et al., 2007), inter-frame difference method (Jifeng & Chengqing, 2013), optical flow method (Mei et al., 2001). After the extensive use of in-depth learning methods, the YOLO algorithm proposed by Redmon et al. in 2015 takes the whole picture as the input of the network and directly obtains the output of the boundary box and the detection result (Sun et al., 2005). This method can detect 45 frames per second, which is faster than other algorithms. The YOLO algorithm no longer slides the window but divides the original picture directly into small non-coincident squares and then convolutes to produce a feature map of this size. Based on the analysis above, it can be concluded that each element of the feature map is also a small square corresponding to the original picture (Huang, 1999; Huang & Du, 2008; Zhao et al., 2004). Then, each element can be used to predict the target of those centre points in the small square. There are three main improvements made by yolov2 compared with its yolov1: batch normalization; Fine-tune the classification model using high-resolution images; Using a priori box yolov2 has started to use K-means clustering to obtain the dimensions of a priori box, and yolov3 continues this method by setting three priori boxes for each down-sampling scale, resulting in a total of nine sizes of a priori box being clustered. The main improvements of yolov3 are: adjusting the network structure; Multi-scale feature is used for object detection (Chen et al., 2005). Object classification replaces soft-max with Logistic (Zhao & Huang, 2012). The top-down feature fusion method of PAN was first proposed in the paper yolov4. The model with the most parameters in yolov5 uses the Focus module before the image tensor is entered into the backbone network. Focus - Sub-grid sampling; SPP: feature fusion; PAN: Top-Down Feature Fusion.

Although the author did not place YOLOv5 in direct test comparisons with yolov4, the test results on COCO datasets are impressive. Therefore, in summary, yolov5 is not only very fast but also has a very lightweight model size and is equivalent to the yolov4 benchmark in accuracy. Now most of the contestants in front of the wheat test on kaggle are using the framework of yolov5 (First known cv, 2021). Generally speaking, both yolov4 and yolov5 have good accuracy in the actual test and training, but the various network structures of yolov5 make users more flexible to use. We can integrate the advantages and disadvantages of yolov4 and yolov5 according to the needs of different projects and give full play to the advantages of algorithms under different detection networks.

Target tracking is a hot topic in the field of vision. Based on the huge practical value of visual tracking and the rapid development of computer technology, and the strong demand for visual tracking technology, developed countries such as Europe and America have made a lot of in-depth research on video tracking technology. In the area of intelligent monitoring, the real-time monitoring system W4 developed by Maryland University implements human tracking, can be used to monitor human behaviour, and can judge whether a person carries objects and other simple behaviours. The Vehicle Traffic Monitoring System at the University of Redding in the United Kingdom is a study of vehicle and pedestrian tracking and interaction identification. (Căleanu et al., 2007) In China, the visual tracking technology has been studied since 1986. The theory and technology in the field of visual tracking have also made considerable progress. With the development of technology and the improvement of related products and equipment, Chinese universities and scientific research institutes have made some breakthroughs in moving target tracking, detection and recognition (Haibo et al., 2017). This paper mainly studies the yolov5s on the basis of the mining vehicle tracking 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, training self-built datasets. Find the best parameters on the local dataset, test and test the data, and finally get the relevant results of the experiment, the general process of the experiment is shown in Figure 1.

3. Core content of yolov5 algorithm

3.1. YOLOv5 origin

YOLOv5 crossed the world when scholars were still immersed in the shock and regret of Joseph Redmon, the father of YOLO, announcing his retirement from CV.
Since YOLOv5 has not officially published a paper, it can only be studied from public code. In the code officially released by YOLOv5, there are four versions of the detection network: YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s. Among them, YOLOv5s is the network with the smallest depth and width of the feature map, and the other three can be considered to be based on it, which has been deepened and widened. YOLOv4 and YOLOv5 are similar in structure but slightly different in detail. YOLOv5 (You Only Look Once) was proposed by Ultralytics LLC in May 2020. It can process 140 frames per second for real-time video image detection, and has a smaller structure. The weight data file of YOLOv5s version is 1/9 of YOLOv4 with a size of 27 MB (Desda, 2021). YOLOv5 is divided into YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5 according to the network depth size and the width of the feature map. YOLOv5s is used as the usage model in this paper.

For the test results on the test set, here is a simple summary of the two experiments: when the backbone uses mobilenetv3 small, thanks to the design of its channel separable convolution, the training speed is indeed faster. It takes about 6 h to complete the training of 300 epochs with yolov5s on my device, while it only takes about 4.5 h to use mobilenetv3 small, in terms of network parameters, mobilenetv3 small also has an advantage. However, in terms of the performance in the test set, yolov5s is in precision, recall, mAP@0.5, mAP@0.5: 0.95 exceeds mobilenetv3 small in four indicators, especially in mAP@0.5 On this index, it is 0.103 higher (Mr. Hang, 2021). Of course, these are only two groups of experiments, and there is not much optimization work such as adjusting parameters. We can only generally think that the characteristics of the two backbones are fast and accurate, and which one to use should be determined according to the specific use scenario. The results of backbone using yolov5s on the test set and backbone using mobilenetv3 small on the test set are shown in Figure 2 (Linlin et al., 2021).

3.2. Basic network structure principles of YOLOv5

The YOLOv5 network structure diagram is divided into two parts: the main network: the input side, the Backbone part, and the detection network: Neck and Prediction part. The main method functions used in the input side are mosaic data enhancement. This paper mainly introduces random scaling, random clipping and random arrangement of pictures to stitch the data, so as to achieve a relatively significant experimental result on small target detection, which changes the problem that YOLO is own method has not responded sufficiently to small target detection from the beginning to now. YOLO is compared with MTC. MTC detection transfers the data source in and then detects and processes the data. If a smaller target is detected but still does not affect the detection results obtained from its experiments, MTC detects all the target areas limited to 12*12, and MTC also trains 12*12 targets during training. 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 can not 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: FPN + PAN structure is mainly to adjust the number of layers to transfer shallow features and reduce the risk of data loss. Prediction uses CILOU _ Loss can also be set to use other IOU_Loss as needed Use and
Figure 2. Comparison between the results of mobile netv3 and yolov5s on the test set.

| Experimental method | Yolov5s results | Mobilenetv3 results |
|---------------------|-----------------|---------------------|
|                     | Precision | Recall | mAP@0.5 | mAP@0.7 | mAP@0.95 | Precision | Recall | mAP@0.5 | mAP@0.7 | mAP@0.95 |
| all                 | 0.822     | 0.645  | 0.752   | 0.475   | 0.652    | 0.607    | 0.649   | 0.386   | 0.430   | 0.430   |
| aeroplane           | 0.804     | 0.825  | 0.962   | 0.507   | 0.883    | 0.600    | 0.649   | 0.430   | 0.500   | 0.283   |
| bicycle             | 0.963     | 0.875  | 0.949   | 0.643   | 0.775    | 0.889    | 0.885   | 0.500   | 0.283   | 0.283   |
| bird                | 0.852     | 0.375  | 0.556   | 0.380   | 0.833    | 0.667    | 0.665   | 0.283   | 0.283   | 0.283   |
| boat                | 0.953     | 0.500  | 0.535   | 0.317   | 0.778    | 0.429    | 0.434   | 0.252   | 0.252   | 0.252   |
| bottle              | 0.930     | 0.500  | 0.840   | 0.511   | 0.692    | 0.556    | 0.543   | 0.332   | 0.332   | 0.332   |
| bus                 | 0.676     | 0.750  | 0.769   | 0.568   | 0.767    | 0.595    | 0.797   |          |          |          |
| car                 | 0.765     | 0.710  | 0.850   | 0.631   | 0.760    | 0.846    | 0.949   | 0.726   | 0.726   | 0.726   |
| cat                 | 1         | 0.578  | 0.827   | 0.586   | 0.718    | 0.600    | 0.670   | 0.295   | 0.295   | 0.295   |
| chair               | 0.675     | 0.684  | 0.803   | 0.470   | 0.681    | 0.467    | 0.672   | 0.305   | 0.305   | 0.305   |
| cow                 | 0.848     | 1      | 0.995   | 0.472   | 1        | 0.218    | 0.53    | 0.275   | 0.275   | 0.275   |
| Dining-table        | 0.729     | 0.500  | 0.745   | 0.489   | 0.455    | 0.500    | 0.484   | 0.155   | 0.155   | 0.155   |
| dog                 | 0.829     | 0.417  | 0.588   | 0.286   | 0.730    | 0.583    | 0.671   | 0.412   | 0.412   | 0.412   |
| horse               | 0.879     | 0.811  | 0.887   | 0.585   | 0.569    | 0.500    | 0.662   | 0.463   | 0.463   | 0.463   |
| motorbike           | 0.980     | 1      | 0.995   | 0.771   | 0.639    | 0.600    | 0.658   | 0.336   | 0.336   | 0.336   |
| person              | 0.924     | 0.737  | 0.827   | 0.481   | 0.688    | 0.633    | 0.670   | 0.335   | 0.335   | 0.335   |

The basic network structure of Focus is shown below. The technical principle used by Focus is derived from the pass-through of YOLOv2, that is, the pass-through layer is made up of four slice phases Concat and convoluted again to change the 26*26*64 structure to 13*13*256 structure, which makes the experimental results more accurate. Convolution CBL consists of Conv + BN + Leaky relu activation function consists of three components. CSP1_X uses the CSPNet network structure for reference on the principle of YOLO, consisting of three convolution layers and X Res_unit modules consist of each other’s Concates. Similar CSP2_X no longer uses Res_. The unit module is changed to CBL, and its basic principle is based on the basic routing structure of YOLOv4 and YOLOv3. SPP uses 1*1, 5*5, 9*9, 13*13. The maximum pooling method of 13 is multi-scale fusion, as shown in the following figure. YOLOv5 has a shorter network structure with deep characteristics, which results in faster operation. By analyzing the different characteristics of learning parameters, the learning map in the neural network model is approximated, and the convergence speed of the network is greatly improved (Du et al., 2007; Han & Huang, 2006). Of course, supervised learning in machine learning can be described by function approximation. By calculating the error between predicted output and expected output, an attempt is made to approximate the characteristics represented by the data so as to minimize the error during training (Han & Huang, 2008).
Functional approximation is a useful tool only when the basic target mapping is unknown (Han et al., 2008). Functional approximation using neural networks is because they are universal approximators (Han et al., 2010). Theoretically, they can be used to approximate any function. As the basic structure of YOLOv5 is still perfected, its authors have not published academic papers to explain it, and the program code is continuously updated. The basic network structure of yolov5 is shown in Figure 3.

### 3.3. Input end basic principles

(1) Mosaic Data Enhancement

The input side of Yolov5 uses the same Mosaic data enhancement as Yolov4. The author of the Mosaic data enhancement proposal, who is also a member of the Yolov5 team, uses the Mosaic data enhancement method during its training model phase, which is an improvement on the Cut-mix data enhancement method. Cut-mix stitches two pictures, while Mosaic’s data enhancement method uses four pictures and stitches them together according to random scaling, random clipping and random arrangement. This enhancement method combines several pictures into one, which not only enriches the dataset but also greatly improves the training speed of the network and reduces the memory requirements of the model. The detection effect for small targets is greatly improved. The main advantages of this design are to enrich the dataset, use four pictures randomly, zoom randomly, and stitch randomly. This design greatly enriches the detection dataset, especially random zoom adds many small targets, which makes the network more robust. Second, reduce the use of GPU, considering that many people may only have one GPU, Mosaic can directly calculate the data of four pictures during enhanced training so that the Mini-batch size does not need to be large and a GPU can achieve better results (Deep eye, 2020) as shown in Figure 4.

(2) Adaptive anchor frame calculation

In the YOLOv5 algorithm, there are anchor frames with initial lengths and widths for different datasets. In network training, the network outputs a prediction box on the basis of the initial anchor box and then compares it with the real box ground-true, calculates the difference between them, updates them in reverse, and iterates over the network parameters (Jiang Dabai, 2020a, 2020b, 2020c). Therefore, the initial anchor frame is also an important part, such as the YOLOv5 initial anchor frame on the COCO data-set:

A predefined border is a set of preset borders. During training, training samples are constructed with the actual offset of the border position from the preset border. An Anchor Box can be adjusted by the aspect ratio of the border and the area (scale) of the border. To define, which is equivalent to a series of rules for generating preset borders, Anchor Box allows you to generate a series of borders anywhere in the image. Since Anchor box usually generates a border at the centre of the point of the Feature Map extracted by the CNN, an Anchor box does not need to specify a central location. Faster R-CNN defines three sets of aspect ratio $= [0.5, 1, 2]$ and three scales $= [8, 16, 32]$, which can be combined to place borders of nine different shapes and sizes. Ratio defines the aspect ratio and scale is the area of the border. AnchorBox is generated by centring on the point on the Feature Map that was last generated by the CNN network (coordinates mapped back to the original map). In the case of Faster R-CNN, the VGG network is used to sample the input image 16 times, that is, a point on the Feature Map corresponds to a 16 times on the input image $* \text{A square area of 16}$ (the field of perception). Depending on the predefined Anchor, a point on the Feature Map is centred to generate nine borders of different shapes and sizes on the original map, as shown in Figure 5.

The points on a feature map correspond to the original figure $16 \times 16$ square area, only using the frame of the area for target positioning, its accuracy will undoubtedly be very poor, and even the ‘frame’ can not reach the target at all. After adding anchor, the points on a feature map can generate 9 boxes of different shapes and sizes, so that the probability of ‘box’ living the target will be great, which greatly improves the recall rate of inspection; Then through the subsequent network to...
adjust these borders, its accuracy can also be greatly improved.

In YOLOv3 and YOLOv4, when training different datasets, calculating the initial anchor frame value is run through a separate program. However, this function is embedded in the code in YOLOv5, and the optimal anchor frame values in different training sets are calculated adaptively for each training. Of course, if you don’t think the calculated anchor box works very well, you can also turn off the automatic calculation anchor box function in your code.

```
parser.add_argument('noautoanchor', action='store_true', help='disable auto-anchor check')
```

The code for control is train. The above line of code in python, set to False, is not automatically calculated for each training session.

\[
b_x = \sigma(t_x) + c_x \cdots \quad (t_x = \log[0.543/(1 - 0.543)] = 0.172) \\
b_y = \sigma(t_y) + c_y \cdots \quad (t_y = \log[0.463/(1 - 0.463)] = -0.148)
\]
bw = Pwew \ldots \ldots \ldots (t_w = \log(224/315) = -0.340) \\
bh = Pheth \ldots \ldots \ldots (t_w = \log(202/280) = -0.326) \\
(1)

For COCO datasets, YOLOv5 configuration file*. 640 is preset in yaml. The size of the anchor box under 640 image sizes, but for custom datasets, YOLOv5 will automatically learn the size of the anchor box again because the target recognition frame often needs to scale the original picture size, and the size of the target object in the dataset may also be different from the COCO data-set. As shown in the figure above, YOLOv5 is learning the dimensions of the automatic anchor box. For the BDD100K data-set, when the picture in the model is scaled to 512, the best anchoring frame is shown in Figure 6.

(3) Adaptive picture scaling

(a) Principle of adaptive picture scaling

In the common target vehicle detection algorithms, the length and width of different pictures and video frames are different. Therefore, in order to solve this kind of problem, the common way is to uniformly scale the original pictures to a standard size and then send them to the detection network. For example, 416 \times 416, 608 \times 608 and other sizes are commonly used in yolov3, yolov4 and yolov5 algorithms, such as scaling the following 800 \times 600 truck image (using 416 \times 416 size format) (Jiang Dabai, 2020a, 2020b, 2020c) (Figure 7).

It can be seen from the picture that although the proportional scaling has been carried out, it can still be seen that after the picture is scaled and filled, the sizes of the black edges at both ends are different. If the picture is filled more, there is redundant useless information of the picture, which greatly affects the running speed. Therefore, the letterbox function is modified in YOLOv5 code to adaptively add the least black edge to the original image. Through this simple improvement, the reasoning speed has been greatly improved, and the actual program running effect is remarkable.

(b) Adaptive picture scaling process

As shown in Figure 8 above, the three groups of pictures from the top are I, II and III respectively.

(I) Calculate the scaling scale. For example, the original film length \times width: 800 \times 600, and the original scaling size is 416 \times 416. After dividing by the size of the original image, you can get two scaling factors of 0.52 and 0.69. Select the small scaling factor.

(II) Calculate the scaled size. Multiply the length and width of the original image by the small scaling factor (800 \times 0.52 = 416; 600 \times 0.52 = 312) to obtain the scaled image size (416 \times 312).

(III) Calculate the black edge filling value will be 416-312 = 104 to obtain the height to be filled originally. Then use np.mod remainder in numpy to get 8 pixels, and
then divide by 2 to get the values to be filled at both ends of the picture height. The size of the new picture is $416 \times 320 (312 + 8)$.

Note: the filling colour in yolov5 is grey, i.e. (114,114,114). During training, the method of reducing the black edge is not adopted, or the traditional filling method is adopted, i.e. reduced to the size of $416 \times 416$. Only when testing and using model reasoning, the way of reducing the black edge is adopted to improve the speed of target detection and reasoning. Since yolov5’s network undergoes 5 times of down sampling, the 5th power of 2 is equal to 32. Therefore, at least remove the multiple of 32 and then take the remainder (Jiang Dabai, 2020a, 2020b, 2020c).

### 3.4. Backbone

A convolution neural network that aggregates and forms image features at different image granularities, both YOLOv5 and YOLOv4 use CSP_Darknet as backbone to extract rich information features from input images. (Cross Stage Partial Networks) is a cross-stage local network. CSP.Net is actually based on Densnet’s idea to copy the feature map of the underlying layer and send a copy through dense block to the next stage to separate the feature map of the underlying layer. CSP.Net solves the problem of duplicating gradient information for network optimization in Backbone, a framework of other large convolution neural networks. The gradient changes are integrated into the feature map from beginning to end, thus reducing the model parameters and FLOPS values, ensuring both the speed and accuracy of reasoning, and reducing the model size. This can effectively alleviate the problem of gradient disappearance (it is difficult to reverse the loss signal through very deep networks), support feature propagation, encourage network reuse of features, and reduce the number of network parameters (Du et al., 2006).

1. Focus structure, as shown in Figure 9 (Backbone Layer 1 focus convolutes pixel un-shuffling instead of...
Mosaic’s data enhancement method is used on the input side. Four pictures are randomly invoked, randomly sized and distributed, stacked, enriching the data, adding many small targets, and improving the recognition ability of small objects (Gan et al., 2021). Four pictures can be calculated at the same time, which increases the Mini-batch size and reduces the consumption of GPU memory. Yolov5 can also first set anchor size by clustering, and then it can calculate the anchor values in different training sets during the training process, during each training session. Then an adaptive picture size zoom mode is used to improve the prediction speed by reducing the black edges (Haitao et al., 2014). The Focus structure does not exist with versions Yolov3 and Yolov4, and the key step is the slicing operation, as shown in the following figure. For example, by inserting the original image 416×416×3 into the Focus structure, the feature map will be changed to 208×208×12 by slicing, and then 32 convolution kernels will be processed to 208×208×32. Using the structure of Yolov5s as an example (Redmon et al., 2016), the original 608×608×3 image is input into the Focus structure, which is sliced into a 304×304×12 feature map, then convoluted into a 304×304×32 feature map after 32 convolution cores (First acquaintance with cv.yolov5 paper, 2020) (Figure 10).

(2) CSP structure. Its basic network structure is shown in Figure 11 (a cross-stage local fusion network, following Dense-net’s dense cross-layer pickup connection idea)

In YOLOv4 network structure, the CSP structure is designed in the main network by referring to the design ideas of CSP_Net. Only CSP structures are used in Backbone in YOLOv4, whereas two different CSP are used in Backbone and Neck in YOLOv5. In Backbone, use CSP1 with residual structure X, because the Backbone network is deep, the addition of residual structure makes the gradients worthwhile to enhance when the layers and layers propagate back, effectively preventing the disappearance of gradients caused by network deepening, resulting in finer feature granularity. Using CSP2 in Neck X, the output of the backbone network is divided into two branches compared to the simple CBL, and then the Concat is used to enhance the network’s ability to fuse the features and retain more rich feature information. The basic network structure of Yolov4 is shown in Figure 12.

The difference between YOLOv5 and YOLOv4 is that only the backbone network in YOLOv4 uses CSP structure. Two CSP structures are designed in YOLOv5. Taking YOLOv5s network as an example, CSP1_X structure applied to Backbone backbone network, another CSP2_X structure applied to (Deep eye, 2020) in Neck.

3.5. Neck

A series of network layers that blend and combine image features and transfer them to the prediction layer, Based on the Mask R-CNN and FPN framework, PANET enhances information dissemination and has the ability to accurately preserve spatial information, which helps to locate pixels appropriately to form a mask. YOLOv5 now uses FPN + PAN as it does in Neck and YOLOv4. The FPN is top-down, and uses the up-sampling method to transfer and fuse information to obtain a predicted feature map. The PAN uses a bottom-up feature pyramid (Jiang Danbai, 2020a, 2020b, 2020c). The FPN + pan network structure model is shown in Figure 13.

The difference between YOLOv5 and YOLOv4 is that the neck structure of yolov4 adopts ordinary convolution operation. In the neck structure of yolov5, the csp2 structure designed by cspnet is used to strengthen the ability
of network feature fusion. The neck network of yolov5 still uses FPN + pan structure, but some improvements have been made based on it. The following figure shows the specific details of the neck networks of YOLOv4 and YOLOv5. Through comparison, we can find that yolov5 uses csp2_1 structure replaces some CBL modules and replaces the CBL module after concat operation with csp2_1 module (Technology digger, 2021). The difference of neck between yolov5 and yolov4 network structures is shown in Figure 14.

### 3.6. Output terminal

(1) Bounding box loss function
Using CIROU_Loss in Yolov5 does the loss function of the Bounding box. Loss function uses CIROU_Loss. Prediction includes the Bounding box loss function and non-maximum suppression (NMS), which effectively solves the problem when the bounding boxes do not coincide. In the processing phase of target detection prediction results, a weighted NMS operation is used to obtain the optimal target frame for the filtering of many target

---

**Figure 12.** Yolov4 structure.

**Figure 13.** FPN + pan structure (Jiang Dabai, 2020a, 2020b, 2020c).
frames that appear. The difference between YOLOv5 and YOLOv4 is that only the backbone network in YOLOv4 uses CSP structure. Two CSP architectures are designed in Yolov5 (Huang & Chau, 2008). Taking YOLOv5s network as an example, CSP1_X structure applied to Backbone backbone network, another CSP2_X structure applies to in Neck (Jiang Dabai, 2020a, 2020b, 2020c). Equation 2 is as follows.

\[
\text{CIOU}_\text{Loss} = 1 - \text{CIOU} = 1 - \left( \frac{\text{IOU} - \frac{\text{Distance}_C^2}{\text{Distance}_C^2}}{1 - \text{IOU} + \nu} \right) + V^2
\]

\[
\nu = \frac{4}{\pi^2} \left( \arctan \frac{W^t_{gt}}{h^t_{gt}} - \arctan \frac{W^p}{h^p} \right)^2
\]

(2) NMS non maximum suppression

In the post-processing of target detection, NMS operation is usually required for the filtering of many target frames. Because CIOU_Loss contains the influence factor V, which involves the information of ground truth, but there is no ground truth when testing reasoning. So yolov4 is in DIOU_Loss is adopted on the basis of DIOU_NMS, while YOLOv5 uses weighted NMS (Wu et al., 2021). It can be seen that DIOU_NMS is adopted, the red part of the lower middle arrow, can also be detected for vehicles that were originally blocked (Jiang Dabai, 2020a, 2020b, 2020c). The picture difference is shown in Figure 15.

4. Experimental results and analysis

In this paper, YOLOv5s network structure and network performance are tested through three training datasets, COCO128 data-set, COCO2017 data-set, and self-made ME_COCO datasets. The reason for using COCO128 data-set (including 128 pictures of various classes) is that it is not sure if the final result of training YOLOv5s network model with self-made training set can be obtained, so the network parameters and training scale are adjusted to achieve the ideal experimental results. COCO2017 data-set and self-made me data-set are used to verify the accuracy of test results, so as to achieve a high accuracy of vehicle identification and detection. This paper also discusses comparisons with other complex networks to illustrate significant improvements in our network model. The configuration is specific to the yaml file format is saved and modified according to the specific information trained. The YOLOv5s file configuration format involved in this paper is shown in the following Figure 16.

The first is the number of NC (number of classes) dataset categories; the second parameter is depth_multiple, which controls the depth of the model. Multiple, which can be used to dynamically adjust the model depth. The third parameter is width_multiple, which controls the width of the model. Actual output channel at 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, Use in three Detect layers (three feature maps) are used, and each
grid_cell of each feature map has three anchors for prediction. The rules of assignment are as follows: the larger the feature map, the farther ahead it is, the smaller the down-sampling rate and the smaller the field of perception, so it is relatively possible to predict some smaller objects with smaller scales, the smaller all the anchors assigned to it; the smaller the feature map, the farther back it is, the relative The larger the down-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. That is, you can use small feature maps (feature map) to detect large targets, or small targets on a large feature map. yolov5s backbone counts onelineforeachmodule, and each line consistsofourparameters. from representsthe input of the current module from that layer, −1 represents the output from the previous layer, and number representsthe theoretical and actual number of repetitions of the current module. Parameter depth_Multiple jointly determines the depth of the network model; Module module
class name, through which common is removed. Find the corresponding classes in py and set up the network modularization. Args is a list, the parameters required for module building, channel, kernel_ Size, stride, padding, bias, and so on, change according to different layers during the network building process. Head is the same as Backbone and is constructed similarly to Backbone and consists of four parameters, that is, from represents the output from that layer for the input of the current module and $-1$ represents the output from the previous layer. However, here is a list, where the input to this layer comes from the output concat of the layer. Number represents the theoretical number of repetitions of the current module, and the actual number of repetitions is determined by the parameter depth_above. Multiple jointly determines the depth of the network model. Module module class name, through which common is removed. Find the corresponding classes in py and set up the network modularization. Args is also a list, the parameters required for module building, channel, kernel_ Size, stride, padding, bias, etc.

### 4.1. Experimental dataset construction

In the experiment, both COCO128 and COCO2017 datasets come from the COCO data-set official website. The full name of COCO is Common Objects in Context, which is a data set provided by Microsoft team for image recognition. COCO collects data by using Amazon Mechanical Turk extensively. COCO datasets now have three label types: object instances, object key-points, and image captions, which are stored using JSON files. The COCO2017 data-set used in this paper is divided into three parts: training, validation and testing. Each part contains 118,287, 5000 and 40,670 pictures, and the total storage size is about 25G. The test dataset has no label information, and the training and validation datasetshave annotations. The dataset currently contains 80 categories, which mainly contain a variety of target detection data. For example, people, trucks, buses, cars, etc.

The self-made me includes 4456 pictures of the front, 8914 pictures of the roof, 1108 pictures of the rear, and 20,880 pictures of miscellaneous items (strong light, night, multi front and rear, etc.). The proportion of experimental data set is shown in Table 1.

### 4.2. Environmental variable parameter model

The research on tracking, detection, recognition and counting technology of mining vehicles based on yolov5 designed in this paper is based on yolov5s network structure. During model reasoning, we can obviously feel that the speed of yolov5s loading model is very fast, whether it is the speed of loading model or the reasoning speed of test pictures. At the same time, it also has higher version requirements for computer and related network model training configuration. The specific test environment is shown in Table 2.

The parameter settings during network training are shown in Table 3. As an important hyper parameter in supervised learning and deep learning, lr0 learning rate determines whether and when the objective function converges to the local minimum. An appropriate learning rate can make the objective function converge to the local minimum in an appropriate time. Momentum is a commonly used acceleration technique in gradient descent method to accelerate convergence. Momentum method is mainly to solve the ill conditioned condition of Hessian matrix; Weight_decay to prevent over fitting. In the loss function, weight decay is a coefficient placed in front of the regularization term. The regularization term generally indicates the complexity of the model. Therefore, the function of weight decay is to adjust the impact of model complexity on the loss function. If the weight decay is large, the value of the complex model loss function will be large. Box is a language specially designed to simplify vector graphics. In addition, there is an integrated development environment, which promotes the use of the language by interactively displaying graphics output. Box combines the convenience of drawing graphics with the mouse and the convenience of describing graphics with a language tailored to vector graphics. Yolov5 uses GIOU_Loss is the loss of the bounding box, and the box is speculated as the mean value of the GIOU_Loss function. The smaller the box, the more accurate it is; Objectness: it is speculated as the average loss of target detection. The smaller the target, the more accurate the target detection is; Classification: it is speculated as the average value of classification loss. The smaller the value, the more accurate the classification is; IOU is a standard for measuring the accuracy of detecting corresponding objects in a specific data set. IOU is a simple measurement standard. As long as it is a task to obtain a bounding box in the output, IOU can be used for measurement; Anchor has anchor point or anchor box in computer vision. Anchor box often appears in target detection is anchor box, which represents a fixed reference frame.

| Table 2. Training environment. |
|--------------------------------|
| CPU : i7-8750H                 |
| GPU: NVIDIA Ge-force GTX1050   |
| Memory : 8G                   |
| Python : 3.8                  |
| Torch-vision : 0.7            |
| Compiler : Anaconda, PyCharm  |
| OS : Window10                 |
| PyTorch : 1.6                 |
| CUDA : 10.1                   |
| CUDNN : 7.4                   |
| YOLO : YOLOv5s                |
| Numpy : 1.18                  |

| Table 3. Training environment. |
|--------------------------------|
| CPU : i7-8750H                 |
| GPU: NVIDIA Ge-force GTX1050   |
| Memory : 8G                   |
| Python : 3.8                  |
| Torch-vision : 0.7            |
| Compiler : Anaconda, PyCharm  |
| OS : Window10                 |
| PyTorch : 1.6                 |
| CUDA : 10.1                   |
| CUDNN : 7.4                   |
| YOLO : YOLOv5s                |
| Numpy : 1.18                  |
Table 3. Training environment.

| Category       | Set value |
|----------------|-----------|
| Learning rate  | 0.0032    |
| momentum       | 0.843     |
| weight_decay   | 0.00036   |
| Box            | 0.0296    |
| iou_t          | 0.2       |
| Anchor_t       | 0.91      |

4.3. Modify the parameters according to your own hardware configuration

The trained model will be saved in runs / exp0 / weights / last.pt and best.pt in yolov5 directory, and the detailed training data will be saved in runs / exp0 / results.txt file. Precautions for training self-built data sets include making the data sets into VOC format. Note that the paths in the two data sets will change in different data sets and different computers. Note that it is modified to the path applicable to the operating PC, and the corresponding TXT generated by Python make-txt.py in the training model. Note that the path in the data will be modified, or the program Python VOC_Label.py generates the path of the corresponding file. Pay attention to modifying the path and category in the data. Data / me.yaml modifies the number of paths and categories in the data. When you choose to use yolov5s as the training model, you need to modify the number of categories in the generated model / yolov5s.yaml file (select the appropriate categories according to your own research direction) or modify model / yolov5s.yaml file to apply the generated results to other files. The choice depends on what model you want to train. python train.py –batch 2 –epochs 80 –data data/me.yaml –cfg models/yolov5s.yaml –weights weights/yolov5s.pt.

Through the testing of data sets and the matching use of relevant required environmental variables, it is not difficult to see that the model has great advantages for vehicle detection. Users can easily solve the preliminary preparations according to the training requirements and the construction of required models. With the rapid development of the Internet, our pace of life is accelerating, accompanied by efficient and convenient data processing technology will also be rapidly improved. At the same time, at the end of the training, the label file is automatically generated to the specified folder, the category is automatically identified and converted to the network configuration, and the training instructions are automatically provided for you. The user only needs to open a CMD window and enter the command to start the training. It greatly saves the time of program redundancy and slow program running speed in previous years, so that you can have more time to learn other knowledge or learn more relevant knowledge, improve the running speed and accuracy of the current code, and the speed increases accordingly. However, there is no good breakthrough in accuracy. While controlling the learning rate, you also try to reduce the learning rate of the machine, with the decline of the learning rate, it can effectively...
ensure that the training model will not rebound in the continuous increase of data set and the extension of time, so as to be closer to the optimal solution. With the continuous training, the increase of the amount of data and the number of training, the learning rate will gradually tend to a stable value.

4.4. Lab page display

Advanced project address block, then open the specified project command address, activate the torch under the system, call the page display module, its label interface is shown in the diagram. The page obtained by running the program is shown in Figure 17.

The main.py file stores test pictures, videos, and monitored specifications, background colour designs, settings for related variables, and the design of the project UI interface as shown in the diagram. The design details of UI interface are shown in Figure 18.

Save the videos and pictures needed for the experiment under the images folder, and the computer can call the videos and pictures directly under the folder. The camera module monitors and processes real-time according to the privileges of using the device. The real-time results of the experiment are displayed on the system page and used for data storage. Picture detection, video detection and real-time monitoring detection are shown in Figures 19 and 20.

4.5. Experimental results and analysis

(1) Data set training results are represented by labels, as shown in Figure 21.

To make the experimental results more intuitive, we cluster the datasets. Divides a data set into different classes or clusters according to specific criteria, such as distance (Li et al., 2021). Data objects in the same cluster
are of the same type as possible, while data objects not in the same cluster differ greatly. First, standardize the data features and reduce the dimension, select the most effective features from the obtained features (Pei et al., 2006), store them in vectors, and measure the similarity based on different types of data (Shun & Huang, 2006; Wang & Huang, 2009). Finally, the distribution of data samples can be visually seen through the clustering results, which makes it easier to analyze the classification error (Wen et al., 2007). The first figure on the left shows the amount of data in the training set, the number of horizontal axis categories, and the vertical axis represents the number of data frames detected in each category, representing the sum of different vehicle detection results; The second in the left figure shows the approximate position of the data set vehicle from the centre point and the distribution density distribution. Dense points are generated according to the position detected by the vehicle and integrated into this table for observation; the third in the left figure is the distribution of vehicle size in the dataset. The larger the vehicle, the larger the values of width and height, and the closer the dense points obtained will be to this direction; The right figure is a detailed statistical diagram of the marker box. Like the left figure, the macro
results obtained in the left figure are integrated into small points to generate the same image, so as to make the experimental data clearer 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 location, picture size, etc.

(2) Mean comparison of all kinds of target detection AP with different parameters, as shown in Figure 22.

(a) Precision (single class accuracy)

From the perspective of prediction results, precision describes how many positive examples predicted by the two classifiers are real positive examples, that is, how many positive examples predicted by the two classifiers are accurate; The proportion of the part that the classifier considers to be a positive class and is indeed a positive class in the proportion that all classifiers consider to be a positive class' measures the probability that the positive class separated by a classifier is a positive class. The two extreme cases are that if the accuracy is 100%, it means that none of all classes is a positive class.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (3)
\]

notes:

① True positive (TP): the real category of the sample is positive, and the result predicted by the model is also positive, and the prediction is correct.

② True negative (TN): the real category of the sample is negative, and the model predicts it as negative, and the prediction is correct.

③ False positive (FP): the real category of the sample is negative, but the model predicts it as positive, and the prediction is wrong.

④ False negative (FN): the real category of the sample is positive, but the model predicts it as negative, and the prediction is wrong.

(b) Recall (regression rate)

Recall describes how many real positive examples in the test set are selected by the two classifiers from the perspective of real results, that is, how many real positive examples are recalled by the two classifiers. The proportion of the part that the classifier thinks is a positive class and is indeed a positive class in all really positive classes' measures the ability of a classifier to find all positive classes. In two extreme cases, if the recall rate is 100%, it means that all positive classes are divided into positive classes by the classifier. If the recall rate is 0%, it means that no positive category is divided into positive categories.

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (4)
\]

Precision and recall are often contradictory performance metrics;

Improve precision = improve the threshold of positive cases predicted by two classifiers = make the positive cases predicted by two classifiers as real as possible;

Improve recall = reduce the threshold of positive cases predicted by the two classifiers = make the two classifiers pick out the real positive cases as much as possible (Huang, & Jiang, 2012);

(3) Accurate result memory curve, as shown in Figure 23.
Box is GIOU_Loss used by YOLOv5 is the loss of the bounding box. The box is speculated as the mean value of the GIOU_Loss function. The smaller the box, the more accurate the target detection is. It can be obtained from the visual data in the figure. With the deduction of the time of the abscissa (x-axis, time_seconds), the mean value of the loss function is also decreasing and tends to an equilibrium point; Objectness: it is speculated as the average loss of target detection. The smaller the target, the more accurate the target detection is; Classification: it is speculated as the average value of classification loss. The smaller the value, the more accurate the classification is; Precision: precision (found positive classes / all found positive classes); Recall: the true accuracy is positive, that is, how many positive samples have been found (how many have been recalled). Recall describes how many real positive examples in the test set are selected by the two classifiers from the perspective of real results, that is, how many real positive examples are recalled by the two classifiers. Val box: verification set bounding box loss; Val Objectness: mean value of target detection loss in verification set; Val classification: the mean value of the classification loss of the verification set; Map is the area enclosed after drawing with precision and recall as two axes, M represents the average, @ the number after @ represents the threshold value for judging IOU as positive and negative samples, @ 0.5:0.95 represents the threshold value, and take the mean value after taking 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. Experimental summary

In this paper, the vehicle detection method based on yolov5s image recognition and processing technology is used to successfully realize the tracking detection and recognition counting of vehicles. From the training results, the algorithm has high recognition rate and recognition speed in both complex environment and bad weather. YOLOv5s not only runs very fast but also greatly reduces the storage space of the model. I believe that with the in-depth research, further improving the accuracy of the program and the identification of other object types will be of great help to the popularization of artificial intelligence and promote the development of smart cities in the future. At present, there is little research and innovation on yolov5s in relevant academic papers, which requires us to calm down to explore and improve more and better methods, flexibly use them according to different scenarios and project needs, learn from each other, and give full play to yolov5s rapid, efficient and accurate detection advantages.

Disclosure statement

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Funding

This work was supported by the Key teaching and research project of Suzhou University [grant number szxy2020]jyxm11;
Baohua, Z., Xiaobo, L., & Junfei, Y. (2009). Vehicle segmentation. Pattern Recognition, 42(8), 950–956. https://doi.org/10.1109/TSMCB.2008.932185

Haibo, L., Lingyun, X., Bin, H., et al. (2017). Research status and prospects of target tracking methods based on in-depth learning. Infrared and Laser Engineering, 46(5), 7. https://doi.org/10.3788/IRLA201746.0502002

Haitao, Z., Xue, Y., & Wanjun, L. (2014). Segmentation algorithm of occluded vehicle curve using vehicle edge. Computer Application Research, 31(7), 2191–2194. https://doi.org/10.3969/j.issn.1001-3695.2014.07.066

Han, F., & Huang, D. S. (2006). Improved extreme learning machine for function approximation by encoding a priori information. Neurocomputing, 69(16-18), 2369–2373. https://doi.org/10.1016/j.neucom.2006.02.013

Han, F., & Huang, D. S. (2008). A new constrained learning algorithm for function approximation by encoding a priori information into feedforward neural networks. Neural Computing and Applications, 17(5-6), 433–439. https://doi.org/10.1007/s00521-007-0135-5

Hu, R.-X., Jia, W., Ling, H., & Huang, D. S. (2012). Multiscale distance matrix for fast plant leaf recognition. IEEE Transactions on Image Processing, 21(11), 4667–4672. https://doi.org/10.1109/TIP.2012.2207391

Huang, D. S. (1999). Radial basis probabilistic neural networks: Model and application. International Journal of Pattern Recognition and Artificial Intelligence, 13(7), 1083–1101. https://doi.org/10.1142/S0218001499000604

Huang, D. S., & Du, J.-X. (2008). A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Transactions on Neural Networks, 19(12), 2099–2115. https://doi.org/10.1109/TNN.2008.2004370

Huang, D. S., & Jiang, W. (2012). A general CPL-AdS methodology for fixing dynamic parameters in dual environments. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(5), 1489–1500. https://doi.org/10.1109/TSMCB.2012.2192475

Huang, Z.-K., & Chau, K.-W. (2008). A new image thresholding method based on Gaussian mixture model. Applied Mathematics and Computation, 205(2), 899–907. https://doi.org/10.1016/j.amc.2008.05.130

Jiang Dabai. (2020a, August 10). A complete explanation of yolov5 core basic knowledge of Yolo Series in simple terms [OL]. Zhihu, https://zhihu.com/computer-vision/172121380

Jiang Dabai. (2020b, May 21). A complete explanation of yolov3 & yolov4 & yolov5 core basic knowledge of Yolo series [OL], Zhihu, https://blog.csdn.net/nan355655600/article/details/106246625

Jiang Dabai. (2020c, May 27). A complete explanation of yolov3 & yolov4 & yolov5 & yolox core basic knowledge of Yolo series [OL], Zhihu, https://zhuhanlan.com/p/143747206

First known cv. (2021, April 6). Yolov5 algorithm for target detection [OL]. Zhihu. https://zhuhanlan.com/p/362765525?ivk_sa=1024320u

Gan, Z., Wenju, L., & Yaoxing, Z. (2021). Traffic sign recognition based on improved yolov5 algorithm [a]. Simulation application branch of China Computer Users Association. Proceedings of the ‘21 national academic conference on simulation technology [C]. Simulation application branch of China Computer Users Association: computer simulation magazine: 5.

References

Baohua, Z., Xiaobo, L., & Junfei, Y. (2009). Vehicle segmentation method based on incremental scanning. http://www.paper.edu.cn/releasepaper/content/200910-358.

Câleanu, C., Huang, D. S., Gui, V., Tiponu, V., & Maranescu, V. (2007). Interest operator versus Gabor filtering for facial depth after the attack [OL], Zhihu, https://zhuanlan.zhihu.com/p/362765525?

First acquaintance with cv.yolov5 paper. (2020, October 6). The most complete overview in the history of Yolo algorithm: from yolov1 to yolov5 [OL]. Polar market platform, https://zhuhanlan.com/p/136382095

First known cv. (2021, April 6). Yolov5 algorithm for target detection [OL]. Zhihu. https://zhuhanlan.com/p/362765525?ivk_sa=1024320u

Gan, Z., Wenju, L., & Yaoxing, Z. (2021). Traffic sign recognition based on improved yolov5 algorithm [a]. Simulation application branch of China Computer Users Association. Proceedings of the ’21 national academic conference on simulation technology [C]. Simulation application branch of China Computer Users Association: computer simulation magazine: 5.
Jifeng, S., & Chengqing, W. (2013). Moving vehicle tracking based on feature point optical flow and Kalman filter. *Journal of South China University of Technology (Natural Science Edition)* (10). https://kns.cnki.net/kcms/detail/detail.aspx?dbname=CJFD2005&filename=HNLG20051005&uniplatform

Laihu, L. (2012). Business tax planning in commercial mineral resources exploration projects. China’s land and resources economy.

Li, Z.-P., Su, H.-L., Zhu, X.-B., Wei, X.-M., Jiang, X.-S., Gribova, V., Filaretov, V. F., & Huang, D. S. (2021). Hierarchical graph pooling with self-adaptive cluster aggregation. *IEEE Transactions on Cognitive and Developmental Systems*, PP(99), 1–1. https://doi.org/10.1109/TCDS.2021.3100883

Liang, X., Wu, D., & Huang, D. S. (2019). Image co-segmentation via locally biased discriminative clustering. *IEEE Transactions on Knowledge and Data Engineering*, 31(11), 2228–2233. https://doi.org/10.1109/TKDE.2019.2911942

Linlin, M., Jianxin, M., Jiafang, H., & Yadi, L. (2021). Research on target detection algorithm based on yolov5s. *Computer Knowledge and Technology*, 17(23), 100–103. https://doi.org/10.14004/j.cnki.cjkt.2021.2402

Mei, Y., Gangyi, J., & Bokang, Y. (2001). Application of computer vision technology in intelligent transportation system. *Computer Engineering and Application*, 190(9), 396–397. http://en.cnki.com.cn/Article_en/CJFDTOTAL-JMSJ201809255.htm

Mr. Hang. (2021, December 24). Yolov5 summary document (theory, code, experimental results) [OL], Zhihu, https://zhu anlan.zhihu.com/p/449257337

Pei, S., Huang, D. S., Li, K., & Irwin, G. W. (2006). “Gene selection by cooperative competition clustering,” Lecture Notes in Bioinformatics, Springer-Verlag, Vol. 4115, pp. 464–474, (International Conference on Intelligent Computing, Kunming, August 16–19, China).

Redmon, J., Divvala, S., Girshick, R., et al. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 20–27 June 2016.

Shun, P., & Huang, D. S. (2006). Cooperative competition clustering for gene selection. *Journal of Cluster Science*, 17(4), 637–651. https://doi.org/10.1007/s10876-006-0077-6

Sun, Z.-L., Huang, D. S., & Cheung, Y.-M. (2005). Extracting nonlinear features for multispectral images by FCMC and KPCA. *Digital Signal Processing*, 15(4), 331–346. https://doi.org/10.1016/j.dsp.2004.12.004

Technology digger. (2021, February 12). Yolov5 algorithm details [OL]. CSDN, https://blog.csdn.net/WZZ18191171661/article/details/113789486

Wang, C., Pan Z. G., & Li X. Z. (2021). Multilevel metric rank match for person re-identification. *Cognitive Systems Research*, 65(3), 98–106. https://doi.org/10.1016/j.cogsys.2020.10.002

Wang, X.-F., & Huang, D. S. (2009). A novel density-based clustering framework by using level set method. *IEEE Transactions on Knowledge and Data Engineering*, 21(11), 1515–1531. https://doi.org/10.1109/TKDE.2009.21

Wang, X.-F., Huang, D. S., & Xu, H. (2010). An efficient local Chan-Vese model for image segmentation. *Pattern Recognition*, 43(3), 603–618. https://doi.org/10.1016/j.patcog.2009.08.002

Wen, G.-Z., Guo, X.-Y., Huang, D. S., & Liu, K.-H. (2007, August 12–17). “Application of self-organizing map in aerosol single particles data clustering”, The 2007 International Joint Conference on Neural Networks (IJCNN), Orlando, Florida, USA, pp. 991–996.

Wu, D., Wang, C., Wu, Y., Huang, DS, et al. (2021). Attention deep model with multi-scale deep supervision for person re-identification. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(1), 70–78. https://doi.org/10.1109/TETCI.2020.303460

Xin, S., Zhenkang, S., & Ping, W. P. (2007). Application of mean shift in target tracking. *Systems Engineering and Electronic Technology*, 29(9), 1405–1409. http://en.cnki.com.cn/Article_en/CJFDTOTAL-XTYD200709000.htm

Zhang, G.-J., Du, J.-X., Huang, D. S., Lok, T.-M., & Lyu, M. R. (2006). Adaptive nearest neighbor classifier based on supervised ellipsoid clustering. Lecture Notes in Artificial Intelligence (LNAI), Springer-Verlag, Vol. 4223, pp. 582–585 (The 2nd International Conference on Natural Computation (ICNC’06) and the 3rd International Conference on Fuzzy Systems and Knowledge Discovery (FSKD’06), Xi’an, September 24-28, 2006, China).

Zhao, Y., & Huang, D. S. (2012). Completed local binary count model. *IEEE Transactions on Image Processing*, 21(10), 4492–4497. https://doi.org/10.1109/TIP.2012.2204271

Zhao, Z.-Q., Huang, D. S., & Sun, B.-Y. (2004). Human face recognition based on multiple features using neural networks committee. *Pattern Recognition Letters*, 25(12), 1351–1358. https://doi.org/10.1016/j.patrec.2004.05.008