Traitement du Signal
Vol. 41, No. 5, October, 2024, pp. 2449-2457
Journal homepage: http://iieta.org/journals/ts

Passenger Vehicle Driver Detection Based on YOLOv5+Stacking

Ying He

Department of Academic Affairs, Guizhou University of Finance and Economics, Guiyang 550025, China

Corresponding Author Email: HeYing@mail.gufe.edu.cn

https://doi.org/10.18280/ts.410519

Received: 23 February 2024
Revised: 10 August 2024
Accepted: 25 September 2024
Available online: 31 October 2024

Keywords:
object detection, integrated learning, non-maximum suppression, attentional mechanisms, YOLO v5

ABSTRACT

The intelligent detection of driver distracted driving and mask wearing can strengthen the safety management of passenger vehicle operators, the correct detection of drivers in monitoring images by regulatory authorities is the basis for the implementation of these two tasks. Although the traditional object detection model can detect the driver and the passenger accurately, there are still many wrong classification between the driver and the passenger. In order to increase the detection rate, this paper uses CBAM attention mechanism and the idea of non-maximum suppression in YOLO v5, and proposes a combination model of YOLOv5+Stacking integrated learning, which can effectively reduce false detection between drivers and passengers and ultimately increase the detection rate of YOLO v5 model. In this paper, this model is used to verify and detect the collected data set, and its evaluation indexes are not only better than the original YOLO v5 model but also better than other similar detection models.

1. INTRODUCTION

Distracted driving is the main cause of passenger car traffic accidents [1-3], and the correct wearing of masks during the outbreak of COVID-19, influenza and the post-epidemic era can effectively protect passenger car drivers from virus infection [4]. Intelligent supervision of passenger vehicle drivers' driving behavior and mask wearing behavior can effectively reduce the safety risks of passenger vehicle enterprises. In the past, the monitoring process was realized by manual sampling of the monitoring images sent back by passenger vehicles every minute, but this management method has disadvantages such as low efficiency, poor effect, limited rationality and discontinuity of management. Real-time detection of driver behavior requires each vehicle to be equipped with high-performance edge computing equipment. For passenger vehicle enterprises, one-time investment is too high and can not be immediately converted into profits. If the surveillance video is sent back to the server for processing, there are problems such as poor signal, high delay, high traffic cost, and high server load. This paper discusses how to use artificial intelligence based on the existing digital system of passenger vehicles to carry out real-time detection and locate the driver in the image of the minute back monitoring image, so as to prepare the technology for the subsequent mask wearing detection and distracted driving detection.

In fact, whether it is mask wearing detection or distracted driving detection, the driver is the subject of detection. The monitoring images transmitted back to the server from the passenger vehicle include external and internal monitoring, while the internal monitoring has both specialized passenger monitoring and driver monitoring, and the driver monitoring comes from the camera in front of the driver's right side. However, it is possible that there is only the driver in the driver monitoring image, or there is both the driver and the passenger, or there is no driver in the image. The driver and the passenger are both human beings and cannot be distinguished from each other only by their physical appearance. The difference lies in whether they are in the driver's seat in front of the steering wheel, i.e., the person sitting in front of the steering wheel is the driver and the other person is the passenger. Thus steering wheel detection is a must. The research in this paper is to detect and localize the driver in front of the steering wheel in the image from all the uploaded images.

2. RELATED WORK

Based on CNN, detection models have been widely applied in the past decade due to their outstanding performance in image recognition and processing. These models can be divided into two-stage models [5-7] and single-stage models [8-10] by design, both of which need to first detect candidate bounding boxes and then perform classification. However, whether single-stage or two-stage models, they usually treat object detection as independent and parallel classification and regression subtasks, operating on each candidate bounding box separately, without explicitly modeling the spatial relationships between objects. Hence, they are deficient in representing spatial relationships between objects.

Compared with CNN-based models, spatial relation networks can effectively capture relational information and context information in object detection, thereby improving detection accuracy and performance [11, 12]. These models perform better for detecting small objects or occluded objects, providing an effective way to enhance the performance of small object detection [13]. Such models can adopt various methods to model the spatial relationships between detected objects. For example, Hu et al. [11] incorporated relation network modules into CNN detection models like Faster R-CNN and FPN, inferring the relationships between candidate bounding boxes through weighted inner products of object features; Xu et al. [14] employed graph neural networks to graph-embed candidate bounding boxes and calculate spatial positional relationships between regions.

Although transformer-based object detection models like DETR [15] do not explicitly model spatial positional relationships between objects, its encoder and decoder can implicitly understand the relative positions and interactions between objects through self-attention layers. However, DETR is not as good as CNN models in capturing local details because it focuses more on global context, while CNNs can learn multi-scale local features through receptive field mechanisms [16-18]. Additionally, the positional encoding in DETR is directly added in the semantic space, unlike word vectors in natural language processing that can interact with each other, so its positional encoding has limitations and may fail to accurately convey spatial positional information, and the inference speed is also slower [15].

Stacking, also known as stacked generalization [19], is an ensemble learning technique that combines the outputs of multiple base models to build a more powerful prediction model. This technique leverages the strengths of different models and typically provides more accurate results than a single model [20-23]. Current research focuses on improving the generalization ability of stacking models, such as through more effective base model selection, adaptive model fusion methods, and the use of complex features extracted by deep learning models [24]. Stacking models have shown significant performance improvements in various fields, including financial risk analysis [25], medical diagnosis [23], image and speech recognition [26].

The YOLO v5+stacking ensemble learning model proposed in this study improves upon YOLO v5 by explicitly modeling only the spatial information of faces and steering wheels due to the characteristics of the data, and enhances the understanding of spatial relationships between objects through a stacking model, thereby significantly improving the detection accuracy by enhancing YOLO v5's spatial representation capability.

3. MODEL ARCHITECTURE AND PRINCIPLE

The goal of this paper is to detect the driver in the image for further safety monitoring, in order to accomplish this goal this paper needs to detect three types of objects: steering wheel, driver and passenger. Misdetection and omission of these three types of objects have different impacts on realizing the goal of this paper. According to the subsequent experimental analysis, these three types of objects are misdetected or omitted on the goal of this paper, as well as the frequency of occurrence of the situation shown in Table1.

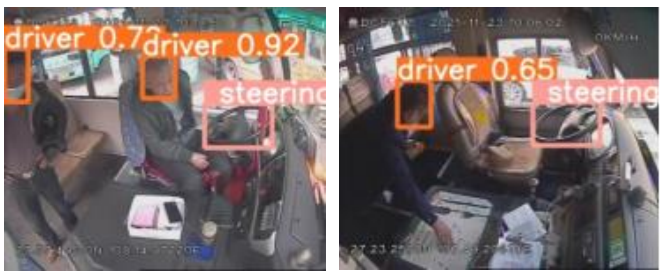
Table 1. Analysis of the impact of misdetections and omissions on targets for the three categories of objects

No.	Types of False or Missed Detections	Possible Impact of Missed or False Detection	Implications	Frequency
1	Driver misclassified as passenger	the image may be discarded	Yes	Often
2	Passenger misclassified as driver (Image without driver)	The driver detected was a passenger	Yes	Often
3	Passenger misclassified as driver (Image with driver)	Two or more drivers detected	Yes	Often
4	Missed steering wheel detection	Potential driver missed detection and image discard	yes	no
5	Background misclassified as steering wheel	Potential incorrect driver detection	yes	Infrequently
6	Background misclassified as passenger	Passenger increase barely affects driver detection	no	Often
7	Background misclassified as driver (Image without driver)	detected the wrong driver	yes	Infrequently
8	Background misclassified as driver (Image with driver)	may detected the wrong driver	yes	Infrequently

In all the aforementioned scenarios, the fourth situation is almost nonexistent; the fifth, seventh, and eighth scenarios occur very rarely; whereas the first, second, third, and sixth scenarios are more frequent. The sixth scenario has a minor impact on the objectives of this paper. Reducing the false positives of types one, two, and three would not only improve the overall detection performance of the model but also greatly assist in achieving the goals of this paper.

Figure 1 shows example images where passengers were incorrectly classified as drivers. In Figure 1 (a), two drivers appear in one image, with the driver on the left being farther away from the steering wheel. Although the passenger in Figure 1 (b) is farther from the steering wheel, they were still classified as a driver. This indicates that, while the YOLO v5 model can classify drivers and passengers with relative accuracy, it also shows insensitivity to certain geometric features such as distance.

Although the false detection of the steering wheel was very rare in the YOLO v5 detection process, the false detection as shown in Figure 2 also occurred, and two steering wheels appeared in one image.



(a) (b)

Figure 1. Example of YOLO v5 detection error (a) The model detects multiple drivers; (b) Passengers misclassified as driver

For the problems shown in Figure 1, the output of the YOLO v5 model was modified in this paper. The output results were divided into pre-classification results and final classification results. A stacking model was inserted between the two classification results to improve the classification accuracy of drivers and passengers. To solve the problem shown in Figure 2, this paper adopts the idea similar to non-maximum suppression to reduce the false detection of the steering wheel class. In addition, the CBAM module used in YOLO v5 not only enhances the model's ability to represent the local area of the image, but also enhances the model's ability to perceive the location and scale of the target object. The overall structure of the model is shown in Figure 3.

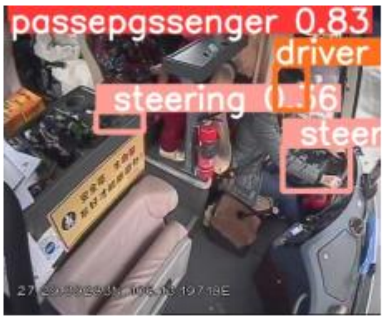


Figure 2. Multiple steering wheels detected by YOLO v5

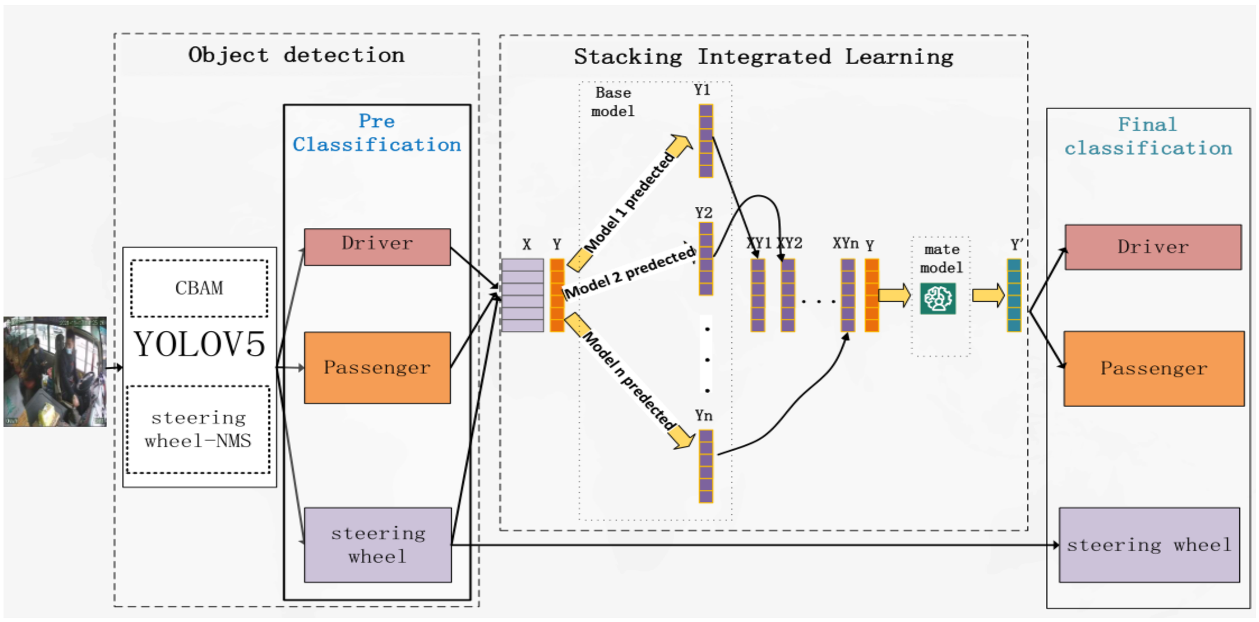


Figure 3. Overall structure of the model

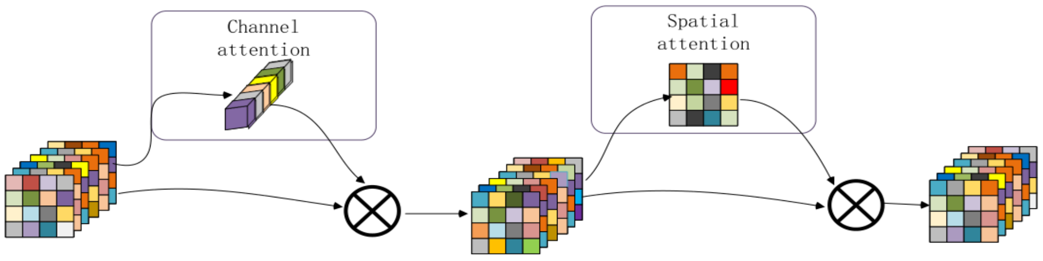


Figure 4. CBAM architecture

3.1 YOLO v5+CBAM

YOLO v5 model is a single-stage object detection model with high precision and fast speed, which can detect a variety of objects accurately and stably. This article uses the YOLO v5x version with the largest number of parameters.

CBAM module [27] is composed of channel attention mechanism and spatial attention mechanism in sequence, and its structure is shown in Figure 4. The channel attention mechanism calculates the importance of each channel through global pooling and multi-layer perceptron, and the spatial attention mechanism calculates the importance of each position through channel pooling and convolutional layer. CBAM can enhance the ability of local image representation, so that the network can accurately capture the location and scale of the target object.

CBAM modules can be inserted in multiple parts of YOLO v5, and their functions vary depending on the insertion location. The 17th, 20th and 23rd layers of YOLO v5 contain rich high-level semantic information and are connected with the detect output of the model. Therefore, by inserting CBAM modules into these positions, the model can not only recognize and distinguish the interaction and context information between different objects more effectively, but also enhance its feature extraction and expression ability for objects of various scales. The YOLOVx5 model and CBAM module structure used in this paper are shown in Figure 5.

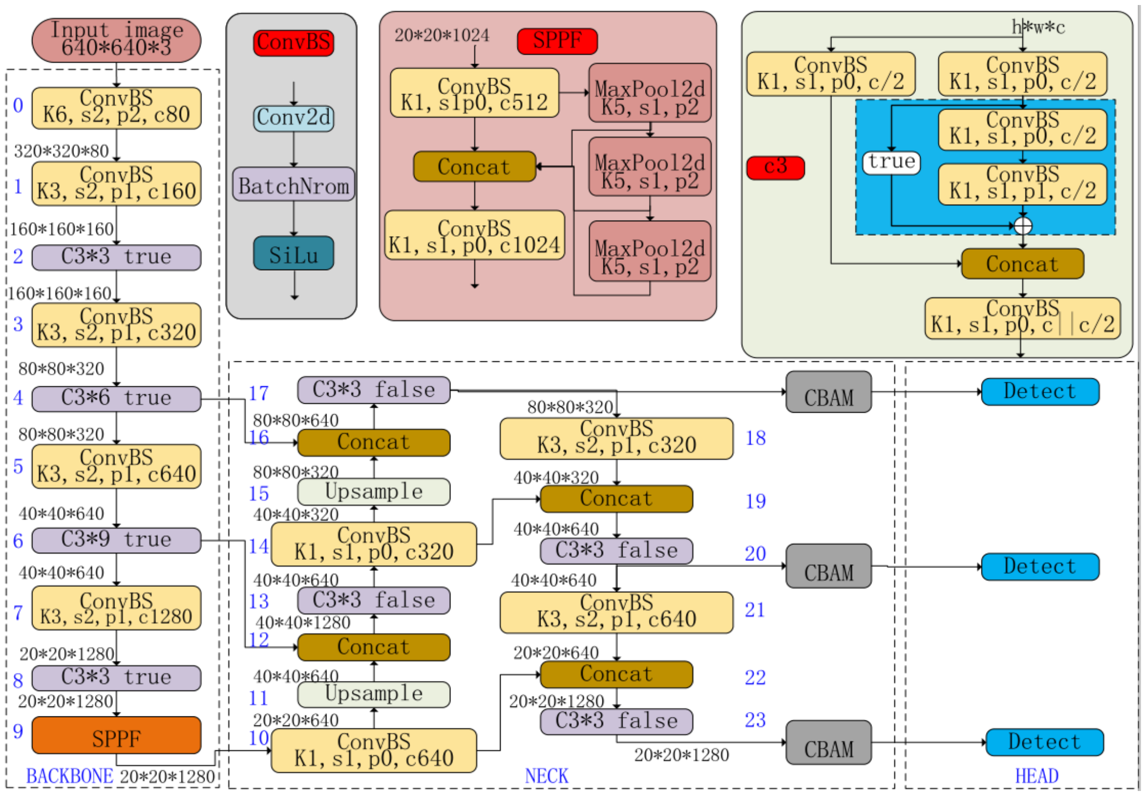


Figure 5. YOLO v5x + CBAM architecture

The loss function of the YOLO v5x model in this paper mainly consists of three parts, namely $L_{cls}, L_{obj}, L_{loc}$, whose values are shown in Eq. (1).

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \tag{1}$$

$\lambda_1, \lambda_2, \lambda_3$ are 0.5, 1 and 0.05, respectively.

Classes loss: BCE loss is used to calculate only the classification loss of the positive sample, and its function is shown in Eq. (2).

$$\mathrm{L} = -\mathrm{ylogp} - (1-\mathrm{y})\log(1-\mathrm{p}) = \begin{cases} -\log \mathrm{p} & \mathrm{y}=1 \\ -\log(1-\mathrm{p}) & \mathrm{y}=0 \end{cases} \tag{2}$$

Objectness loss: The BCE loss is still used, which refers to the CIoU of the target bounding box and GT Box of the network prediction. here the loss is calculated for all samples, and the confidence loss function is shown in Eq. (3).

$$L_{obj} = 4.0 \cdot L_{obj}^{\mathrm{small}} + 1.0 \cdot L_{pkj}^{\mathrm{medium}} + 0.4 L_{\mathrm{large}}^{\mathrm{large}} \tag{3}$$

Location loss: CIoU loss is used to calculate the location loss for positive samples only, and its function is shown in Eq. (4).

$$\mathrm{L_{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2(\mathrm{b}, \mathrm{b^{gt}})}{\mathrm{c}^2} + \alpha v \tag{4}$$

3.2 Steering wheel type non-maximum suppression

Although the probability of steering wheel misdetections occurring is extremely low, misdetections have occurred as shown in Figure 2. Multiple steering wheels in an image will have a large impact on the subsequent integration learning, so in this paper the idea of non-maximum suppression similar to the anchor frame selection in object detection is adopted. When multiple steering wheels are detected in an image, the steering wheel with the highest reliability in the image is selected, and the basis for selection is shown in Eq. (5).

$$Anchor_{steer} = max\,(confidence_i * score_i, confidence_j * score_j) \tag{5}$$

$confidence_i$ represents the confidence level of the ith detected steering wheel in the current image, $score_i$ represents the classification score of the steering wheel in that detection frame.

3.3 Stacking integrated learning model

Table 2. Template for base model dataset in stacking ensemble model

Field Name	Notes	Field Name	Notes
width	Image width	Steering_x	Top-left x-coordinate of the steering wheel
height	Image height	Steering y	Top-left y-coordinate of the steering wheel
face_x	Top-left x-coordinate of the face bounding box	Steering w	Width of the steering wheel
face_y	Top-left y-coordinate of the face bounding box	Steering_h	Height of the steering wheel
face_w	Width of the face bounding box detected by YOLO v5	TrueClass	The ground truth's true class for the face bounding box
face_h	Height of the face bounding box		

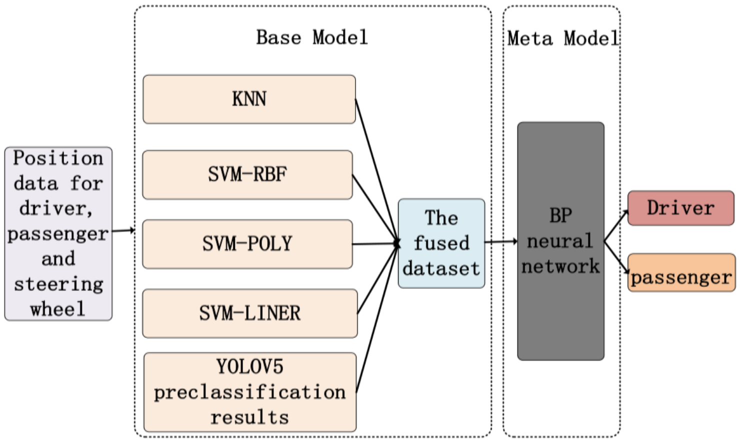


Figure 6. The stacking model proposed in this paper

The learning data of the stacking model comes from the pre-detection results obtained from the YOLO v5 model, and its learning data is the geometric information between the objects, which is shown in Table 2.

As shown in Figure 6, the stacking model proposed in this paper includes base models consisting of Support Vector Machine (SVM) models [28] with three types of kernel functions (poly, rbf, linear) and the K-Nearest Neighbors (KNN) model, as well as the preliminary classification results from the YOLO v5 model itself, forming a total of five groups of base learning models. These models are responsible for classifying the faces in the images as either drivers or passengers. Building upon this, the secondary meta-learner model, a Back Propagation (BP) neural network [29], will learn from the outputs of the five groups of base models and their corresponding true labels. Ultimately, the meta-model's classification results are used to update and refine the preliminary categorization results of drivers and passengers to obtain the final classification outcome. This approach significantly improves the accuracy of the detection.

In the SVM model, when the dataset is not linearly separable, kernel functions can be employed to map the feature data into a higher dimensional space, where it becomes linearly separable. The three types of kernel functions utilized in the SVM [28] in this article are as follows:

Linear kernel function, Liner:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j \tag{6}$$

Polynomial kernel function, Poly:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j + 1)^d \tag{7}$$

Gaussian kernel function, RBF:

$$k(x_i, x_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right) \tag{8}$$

4. EXPERIMENT AND RESULT ANALYSIS

4.1 Experimental data

The training and testing approach for the models used in this paper is primarily divided into three stages: The first stage focuses on the YOLO v5 object detection model; the second stage centers on the base models (SVM and KNN) of the ensemble model; and the third stage revolves around the meta-model (BP neural network) of the ensemble model. Since YOLO v5 not only exhibits misclassifications between drivers and passengers but also suffers from missed detections or misdetections of faces, leading to inconsistencies between the detected face data and the face data in the image annotations, and YOLO v5 itself is one of the base models, the detection boxes used during the training and testing in the second and third stages correspond to the detection results of YOLO v5, while the classes correspond to the ground truth classes. The training, validation, and test data for the second stage correspond to the training, validation, and test data from the first stage. The training, validation, and test data for the third stage correspond to the training, validation, and test data from the second stage, respectively. The specific training and test data for each stage are as follows.

The first stage involves using the YOLO v5 model for object detection, with the dataset being PCSI - Passenger Car Surveillance Images. This is an image dataset containing 20,000 image files, 10,000 of which are surveillance images of the driver's seat. In other words, these images include 10,000 with a steering wheel present and 10,000 without a steering wheel. Of these images, 7,864 contain a driver in the driver's seat, while 2,136 do not have a driver present, and the total number of passengers is 7,049. This dataset is primarily used for training and testing the YOLO v5 model for object detection. We have annotated the drivers, passengers, and steering wheels in this dataset using VOTT, and different file names are used to distinguish whether a given image depicts a driver actively driving. In this paper, we randomly selected 70% of the images from PCSI as the training set for YOLO v5, 15% as the validation set, and 15% as the test set. The test set consists of 3,000 images, of which 1,500 contain a steering wheel, 1,126 images include a driver, 363 images do not contain a driver, and there are a total of 1,063 passengers.

The second stage uses the base models (SVM and KNN) of the ensemble model to classify drivers and passengers. The training, validation, and test data for this stage correspond to the training, validation, and test data from the first stage. As the detection results for the test set in the first stage missed 25 passengers (detected as background), and no drivers were detected as background, the test set for this stage includes 1,126 driver records and 1,038 passenger records, for a total of 2,164 data points.

The third stage takes the outputs of the individual base models as a feature vector and inputs it to the BP neural network meta-model. The training, validation, and test sets for this stage correspond to those from the second stage. The test data for this stage also consists of 2,164 data points, divided into 6 columns, with the first 5 columns representing the feature vector corresponding to the classification results of the 5 base learning models for the current face bounding box, and the last column representing the ground truth class for the face bounding box.

4.2 Experiments and results analysis of the YOLO v5 model

For object detection experiments, the hardware and software configuration utilized was dual E5-2666 CPUs, 64GB RAM, two Tesla M40 24GB GPUs, CUDA 10.1, PyTorch 1.7.1, and Python 3.7. The YOLO v5 model was optimized using SGD with a momentum of 0.937, an initial learning rate of 0.01, a weight decay of 0.0005, a training threshold of 0.2, image size normalized to 640x640, 300 epochs, and a batch size of 16.

4.2.1 Experiments with the original YOLO v5x model

In this paper, we first used the original YOLO v5 model to conduct 3-class training and testing for drivers, passengers, and steering wheels on the images. The test results are shown in the confusion matrix in Figure 7, where P represents the precision for detecting that class of objects, R represents the recall rate for that class of objects, and C represents the count. Although the accuracy rates for drivers and passengers are relatively high, there are many instances of drivers being detected as passengers, and vice versa. The confusion matrix includes an additional 'background' class. The detection results show that many passengers were not detected, and many background regions were detected as passengers. There were also 3 cases where the background was detected as a driver. Furthermore, the recall rate for steering wheels is 1, although two background regions were predicted as steering wheels, the precision for the steering wheel class is approximately 1.

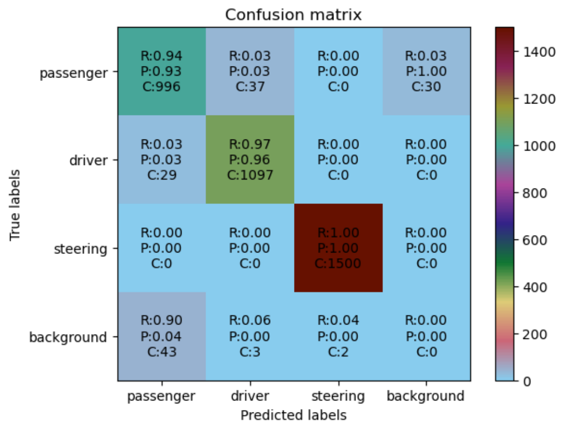


Figure 7. Detections by original YOLO v5 model

4.2.2 Experiments with the CBAM module

After incorporating the CBAM module into the YOLO v5 model, the model implemented a sequential attention structure from channel to spatial dimensions. Figure 8 shows the detection results after using the CBAM module. As can be seen from the figure, adding the CBAM module significantly improved the recall rate and precision for the passenger class. The recall rate and precision for the driver class also increased by approximately 1% each.

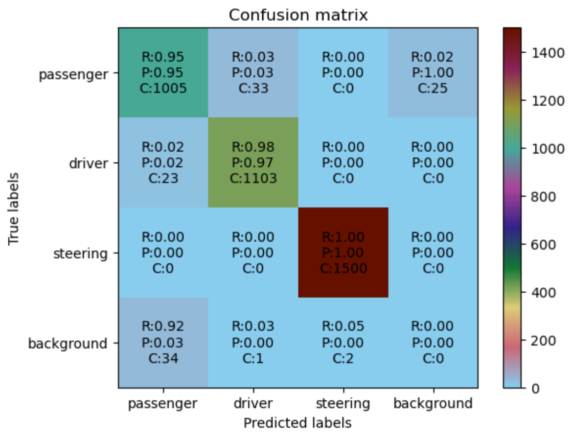


Figure 8. YOLO v5+CBAM detection results

4.2.3 Experiments with non-maximum suppression for the steering wheel class

After integrating CBAM into YOLO v5, we applied the concept of non-maximum suppression to the detection results for the steering wheel class. Figure 9 shows the detection results. From the results, we can see that the two previous instances of background being detected as a steering wheel have been resolved using this method.

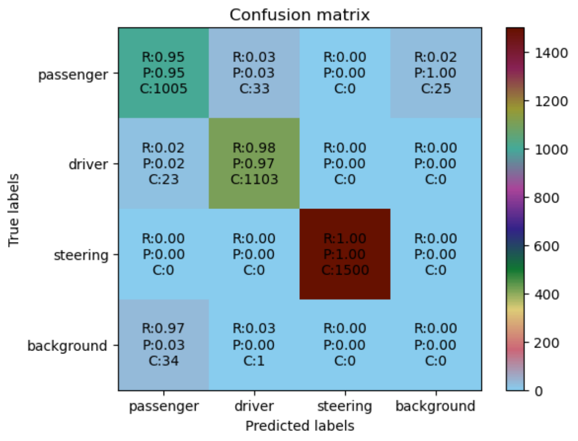


Figure 9. YOLO v5+CBAM + steering-wheel-type NMS detection results

4.3 Experiments and results analysis of ensemble learning models

The experiments in this section continue to use the confusion matrix for evaluation. Additionally, among the four base models in the ensemble learning approach, the parameters for the three SVM models were obtained through a grid search for the optimal values on the dataset employed in this paper.

4.3.1 Experiments on the SVM-RBF base model

For the SVM model based on the RBF kernel function, the search range for the C parameter was 0.1-5, with an increment of 0.25 each time, and the optimal parameter found was

4.849999999999999. The search range for gamma was 1-101, with an increment of 1 each time, and the optimal value for this dataset was 22.

Figure 10 shows the test results of the SVM model based on the RBF kernel function. While this model achieved high classification accuracy for drivers and passengers on the manually labeled test data, there were still many classification errors between drivers and passengers from a quantitative perspective, with 44 out of 2,164 classification objects being misclassified. Notably, 2 of the bounding boxes detected as passengers by YOLO v5 in the background were incorrectly classified as Drivers by the SVM-RBF model.

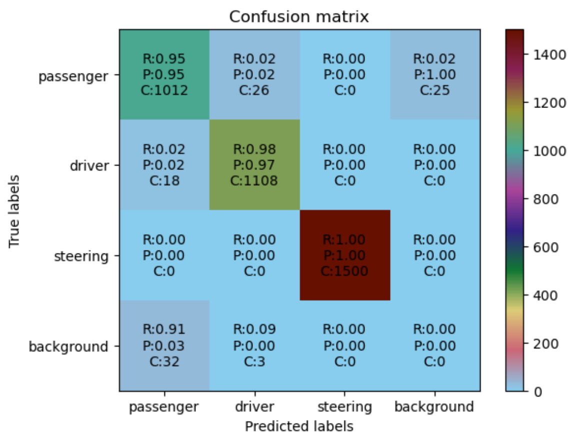


Figure 10. Test results for the SVM-RBF model

4.3.2 Experiments on the SVM-POLY base model

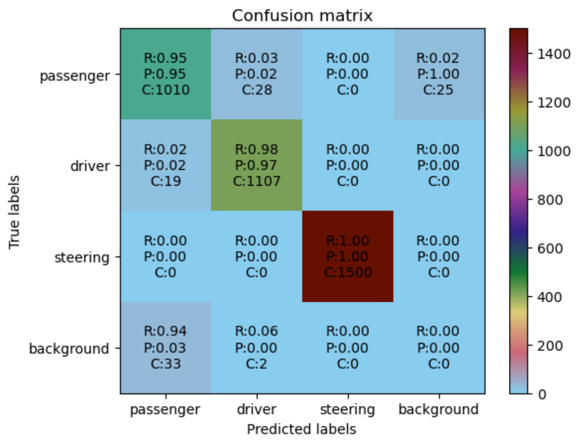


Figure 11. Test results for the SVM-POLY model

For the SVM model based on the polynomial kernel function, the parameter search range for parameter C was 0.1-5, with an increment of 0.25. The optimal result found was 0.8499999999999999. The search range for the coef0 parameter was 2-6, with an increment of 0.5, and the optimal result found was 3.5. The search range for the degree parameter was 0-5, with an increment of 0.5, and the optimal result found was 5. Figure 11 shows the test results for the SVM model based on the polynomial kernel function. From the results, it can be seen that using the polynomial kernel SVM model results in a slight performance decrease compared to using the RBF kernel SVM model for both test methods. However, among the background bounding boxes detected as passenger by YOLO v5, only 2 were misclassified as Driver by the SVM-RBF model.

4.3.3 Experiments on the SVM-linear base model

For the SVM model based on the linear kernel function, the search range for the C parameter was 0.1-5, with an increment of 0.25. The optimal parameter value found was C': 3.5999999999999996.

Figure 12 shows the test results for the SVM model based on the linear kernel function. On both test datasets, the error rate of the SVM-linear classifier was significantly higher.

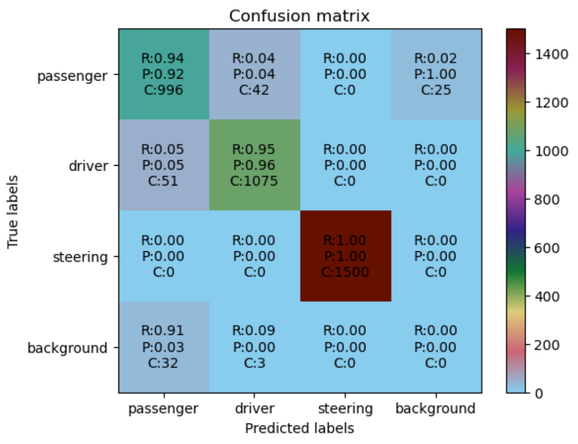


Figure 12. Test results for the SVM-linear model

4.3.4 Experiments on the KNN base model

In this study, the KNN model employed the KNeighborsClassifier classifier, with the default parameters of n_neighbors=5, algorithm=auto, leaf_size=30, and p=2 [30].

Figure 13 illustrates the test results obtained using the KNN model. The results demonstrate that the accuracy of the KNN model is comparable to that of the SVM-linear model.

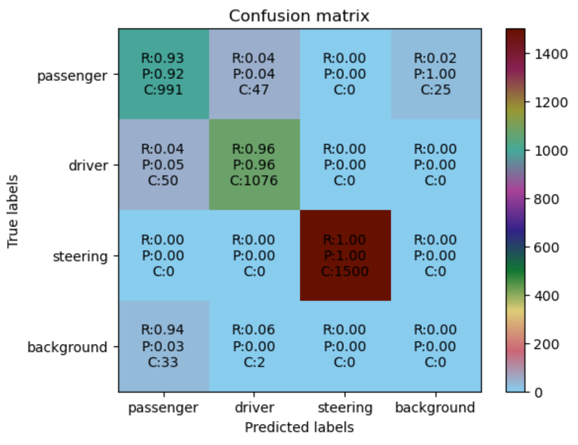


Figure 13. Test results for the KNN model

4.3.5 Meta-model experiments: BP neural network

The meta-model BP neural network in this study had an input layer with 5 nodes, corresponding to the prediction results of the 5 base models. The network consisted of three hidden layers, each with 100 neurons. To prevent overfitting, a dropout rate of 0.65 was employed, along with the ReLU activation function. The output layer featured a single node, utilizing the mean squared error function as the loss function.

The prediction result was a floating-point number that required conversion into a binary classification label (greater than 0.5 for 1, representing the driver class, otherwise 0 for the passenger class). The Adam algorithm served as the optimizer, with an exponential decay strategy for adjusting the learning rate, initially set at 0.01. The learning rate underwent exponential decay, being reduced to 0.8 times its previous value every 200 epochs. A total of 2000 training epochs were performed.

Figure 14 depicts the test results obtained after employing the meta-model, which also represents the final outcome of the entire model. The implementation of the proposed model significantly reduced misclassification cases between drivers and passengers, indicating that this method can effectively perceive the positional relationships among the driver, passenger, and steering wheel.

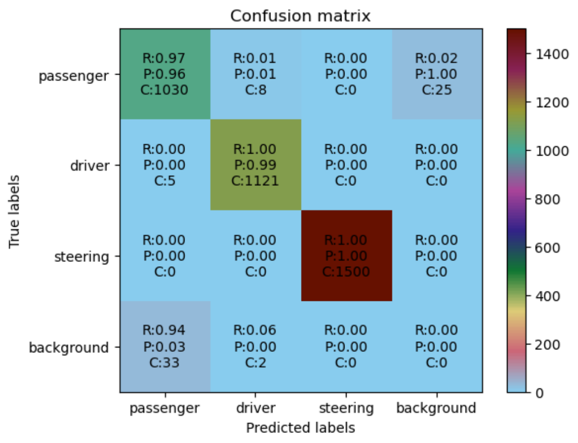


Figure 14. Test results for the YOLOv5+Stacking model

4.4 Benchmark evaluation of models on this study's dataset

Since there is no research on the detection of passenger vehicle drivers, this paper uses several commonly used and different types of detection models to test the data set in this paper. Figure 15 shows the PR curves of several models. As can be seen from Figure 15, the model in this paper is not only superior to the original YOLOv5 model. It is also significantly better than the Transformer-based DETR model. Table 3 shows the comparison of evaluation indicators among these models. The model in this paper is the best among all evaluation indicators.

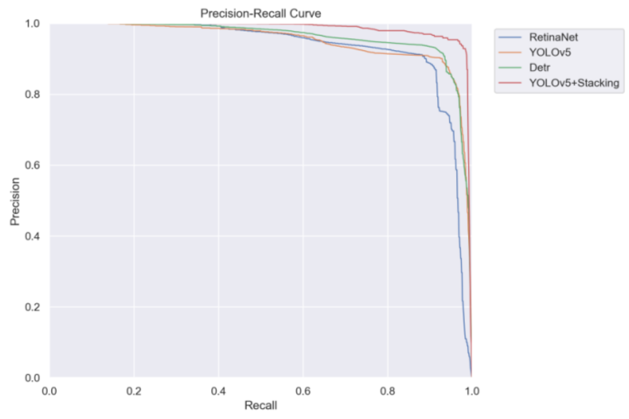


Figure 15. Test results for the SVM-linear model

Table 3. Benchmark evaluation of models on this study's dataset

Model Name	Recall Rate	Precision Rate	F1 Score	mAP@.50
YOLOv5	97.04%	96.54%	96.79%	94.95%
RetinaNet	95.7%	95.3%	95.5%	92.34%
DETR	97.8%	97.3%	97.56%	95.91%
YOLOv5+ Stacking	98.82%	98.52%	98.67%	98.53%

5. CONCLUSION

Building upon the YOLO v5 model, this study first employed a CBAM module to enhance the model's perception of positional information. Subsequently, a non-maximum suppression approach was adopted to reduce false detections of the steering wheel class. Finally, the extraction of detected object bounding box locations and the utilization of the stacking ensemble algorithm further minimized misclassifications between drivers and passengers. The proposed model improves detection accuracy and can precisely locate the primary subject of bus safety monitoring—the driver. This research lays the foundation for subsequent monitoring of driver distraction and face mask detection. In this study, the stacking ensemble algorithm utilized the detection results from the YOLOv5 model as positional information inputs. The overall training and testing process was relatively complex. Future work will investigate how to incorporate positional relationship information between objects into the model to enhance detection accuracy, ultimately enabling end-to-end training, testing, and inference.

REFERENCES

[1] Olson, R.L., Hanowski, R.J., Hickman, J.S., Bocanegra, J. (2009). Driver distraction in commercial vehicle operations (No. FMCSA-RRT-09-042). United States. Department of Transportation. Federal Motor Carrier Safety Administration. https://doi.org/10.21949/1502647

[2] Young, K., Regan, M., Hammer, M. (2007). Driver distraction: A review of the literature. Distracted Driving, Sydney, NSW: Australasian College of Road Safety, 379-405.

[3] Lee, J.D., Young, K.L., Regan, M.A. (2008). Defining driver distraction. Driver Distraction: Theory, Effects, and Mitigation, 13(4): 31-40.

[4] Chiu, N.C., Chi, H., Tai, Y.L., Peng, C.C., Tseng, C.Y., Chen, C.C., Lin, C.Y. (2020). Impact of wearing masks, hand hygiene, and social distancing on influenza, enterovirus, and all-cause pneumonia during the coronavirus pandemic: Retrospective national epidemiological surveillance study. Journal of Medical Internet Research, 22(8): e21257. https://doi.org/10.2196/21257

[5] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 580-587.

[6] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, pp.

1440-1448.

[7] Ravishankar, K., Devaraj, P., Yeliyur Hanumathaiah, S.K. (2023). Floor segmentation approach using FCM and CNN. Acadlore Transactions on AI and Machine Learning, 2(1): 33-45. https://doi.org/10.56578/ataiml020104

[8] Redmon, J. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779-788.

[9] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, pp. 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[10] Ross, T.Y., Dollár, G.K.H.P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2980-2988.

[11] Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y. (2018). Relation networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3588-3597.

[12] Chi, C., Wei, F., Hu, H. (2020). Relationnet++: Bridging visual representations for object detection via transformer decoder. Advances in Neural Information Processing Systems, 33: 13564-13574.

[13] Fu, K., Li, J., Ma, L., Mu, K., Tian, Y. (2020). Intrinsic relationship reasoning for small object detection. arXiv preprint arXiv:2009.00833. https://doi.org/10.48550/arXiv.2009.00833

[14] Xu, H., Jiang, C., Liang, X., Li, Z. (2019). Spatial-aware graph relation network for large-scale object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9298-9307.

[15] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020). End-to-end object detection with transformers. In European Conference on Computer Vision, pp. 213-229. https://doi.org/10.1007/978-3-030-58452-8_13

[16] Jiang, Y., Liang, J., Cheng, T., Lin, X., Zhang, Y., Dong, J. (2022). MTPA_Unet: Multi-scale transformer-position attention retinal vessel segmentation network joint transformer and CNN. Sensors, 22(12): 4592. https://doi.org/10.3390/s22124592

[17] Lu, K., Xu, Y., Yang, Y. (2021). Comparison of the potential between transformer and CNN in image classification. In ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application, Shenyang, China, pp. 1-6.

[18] Zhao, Y., Wang, G., Tang, C., Luo, C., Zeng, W., Zha, Z.J. (2021). A battle of network structures: An empirical study of CNN, transformer, and MLP. arXiv preprint arXiv:2108.13002. https://doi.org/10.48550/arXiv.2108.13002

[19] Wolpert, D.H. (1992). Stacked generalization. Neural Networks, 5(2): 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1

[20] Ting, K.M., Witten, I.H. (1997). Stacking bagged and dagged models. In International Conference on Machine Learning. https://api.semanticscholar.org/CorpusID:5254683.

[21] Džeroski, S., Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? Machine Learning, 54: 255-273. https://doi.org/10.1023/B:MACH.0000015881.36452.6e

[22] Pavlyshenko, B. (2018). Using stacking approaches for machine learning models. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, pp. 255-258. https://doi.org/10.1109/DSMP.2018.8478522

[23] Naimi, A.I., Balzer, L.B. (2018). Stacked generalization: an introduction to super learning. European Journal of Epidemiology, 33: 459-464. https://doi.org/10.1007/s10654-018-0390-z

[24] Kunapuli, G. (2023). Ensemble Methods for Machine Learning. Simon and Schuster.

[25] Liang, D., Tsai, C.F., Lu, H.Y.R., Chang, L.S. (2020). Combining corporate governance indicators with stacking ensembles for financial distress prediction. Journal of Business Research, 120: 137-146. https://doi.org/10.1016/j.jbusres.2020.07.052

[26] Aboneh, T., Rorissa, A., Srinivasagan, R. (2022). Stacking-based ensemble learning method for multi-spectral image classification. Technologies, 10(1): 17. https://doi.org/10.3390/technologies10010017

[27] Woo, S., Park, J., Lee, J.Y., Kweon, I.S. (2018). Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 3-19.

[28] Kecman, V. (2005). Support vector machines–an introduction. In Support Vector Machines: Theory and Applications, pp. 1-47. https://doi.org/10.1007/10984697_1

[29] Li, J., Cheng, J.H., Shi, J.Y., Huang, F. (2012). Brief introduction of back propagation (BP) neural network algorithm and its improvement. In Advances in Computer Science and Information Engineering, pp. 553-558. https://doi.org/10.1007/978-3-642-30223-7_87

[30] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238. https://doi.org/10.48550/arXiv.1309.0238