World Electric Vehicle Journal

MDPI

Article

Research on Cone Bucket Detection Algorithm Based on Improved YOLOv5s

Jiyue Zhuo, Gang Li * and Yang He

School of Automobile and Traffic Engineering, Liaoning University of Technology, Jinzhou 121001, China; 219802057@stu.lnut.edu.cn (J.Z.); heyang121000@lnut.edu.cn (Y.H.)
* Correspondence: qcxyligang@lnut.edu.cn

Abstract: In order to address the problems associated with low detection accuracy, weak detection ability of small targets, insufficiently obvious differentiation of colors, and inability to accurately locate the actual position of the target object in the Formula Student Autonomous China, the YOLOv5s algorithm is improved by adding coordinate attention, modifying the color space transformation module, and adding a normalized Gaussian Wasserstein distance module and a monocular camera distance measurement module. Finally, it is experimentally verified that by adding and modifying the above modules, the YOLOv5s algorithm's precision is improved by 6.9%, recall by 4.4%, and mean average precision by 4.9%; although the detection frame rate decreases, it still meets the requirement. Monocular camera distance measurement has a maximum error of 5.64% within 20 m in the Z-direction and 5.33% in the X-direction.

Keywords: deep learning; target detection; YOLOv5; attention mechanism; monocular camera distance measurement

Citation: Zhuo, J.; Li, G.; He, Y. Research on Cone Bucket Detection Algorithm Based on Improved YOLOv5s. *World Electr. Veh. J.* **2023**, *14*, 269. https://doi.org/10.3390/wevj14100269

Academic Editors: Joeri Van Mierlo and Grzegorz Sierpiński

Received: 28 August 2023
Revised: 18 September 2023
Accepted: 25 September 2023
Published: 27 September 2023

1. Introduction

As part of the continuous progress in artificial intelligence technology, automobiles are gradually developing towards electrification, intelligentization, and network connectivity. The research work on driverless cars in terms of intelligence is a challenging and promising field, which involves a number of disciplines, including artificial intelligence, control systems, sensor technology, and so on. Although there has been great progress in artificial intelligence technology in recent years, there are still certain challenges to the landing of driverless cars, such as that current sensor technology has not yet been able to fully solve the challenges in situations such as inclement weather and night time, and there is still no perfect solution regarding the morality, ethics, policies, and regulations related to driverless cars, so the complete realization of driverless cars is still a certain degree of difficulty. Based on the above problems, automobile companies put forward the solution of advanced driving assistance systems to provide intelligent car customers with adaptive cruise control, lane-keeping assist, automatic parking, and other auxiliary driving functions, and the installed ratio of advanced driving assistance systems for new energy vehicles sold in China has already reached 4.9% and shows an increasing trend every year. So, in the future, there will be more organizations that will be involved in research on driverless cars.

The Society of Automotive Engineering has organized the driverless formula racing competition for college students to cultivate their exploration in the direction of intelligent vehicles. The main technologies used in driverless formula racing cars are environment perception, simultaneous localization and map building, path planning, and trajectory tracking. As the first step in driverless driving, the advantages and disadvantages of the perception results play a decisive role in the stable operation of the driverless racing car.

The YOLOv5s algorithm has been favored by many teams in driverless formula car racing competitions due to its fast detection speed and good detection accuracy, but there are still some problems in its practical application. The YOLOv5s algorithm, as a

single-stage target detection algorithm, shows somewhat reduced detection accuracy when compared to two-stage detection algorithms [1–3], and the objects that need to be detected by driverless formula racing cars are mainly cone buckets, whose shapes are relatively small and occupy fewer pixels on the image, while YOLOv5s adopts a PANet module that fuses the feature depth maps, which decreases its ability to detect small targets. Distant cone buckets are more difficult to detect accurately, and formula racing cars being able to detect cone buckets farther away is beneficial for the cars to make decisions and plan in advance. The YOLOv5s model employs color space transformation in the training data as a preprocessing step before the model training. This transformation alters the color information of the objects in the dataset, resulting in a decrease in the model's accuracy when it comes to color recognition. At the same time, it should be noted that the YOLOv5s method, functioning as a 2D target detection algorithm, lacks the capability to detect the depth information of the target. Consequently, the detection outcomes it produces are unsuitable for the purpose of path planning in the context of racing cars.

Aiming at the above problems, this paper will start the research by improving the accuracy of the YOLOv5s algorithm in detecting cone buckets, its ability in color differentiation, its ability to detect small targets, and its ability to detect the actual position of the target object, as well as referencing a large amount of related literature. Yu Zhang et al. [4] improved the detection of small targets by adding the Flip-Mosaic algorithm to the YOLOv5 algorithm. Daniel Padilla Carrasco et al. [5] incorporated a multiscale mechanism into the YOLOv5 framework. The objective of this addition was to mitigate the issue of excessive training parameters and enhance the model's efficacy in detecting small targets. Ying Zhang et al. [6] integrated a self-supervised learning network into the YOLOv4 network, which reduces the amount of manual labeling required and increases the model's prediction ability and detection accuracy. Wentong Wu et al. [7] improved the YOLOv5 algorithm by using a localized Fully Convolutional Neural Network (FCN), which improves its detection capability for small targets. To enhance the detection capabilities of the YOLOv5 model for tiny and dense objects, Xingkui Zhu et al. [8] replaced the original detection head with the Transformer's detection head and added an extra detection head, as well as a convolutional attention mechanism module. Bojan Strbac et al. [9] suggested a technique for determining the distance of an object based on the YOLO algorithm and the stereo vision concept. K. Karthika et al. [10] offered a method for estimating the distance to the vehicle ahead using an artificial neural network and a monocular camera. Zhiguo Liu et al. [11] deleted the feature layer and prediction head with poor feature extraction ability in YOLOv5, and also integrated a new type of feature extractor with stronger feature extraction ability into the network; at the same time, they borrowed the idea of a residual network to integrate coordinate attention into the network. Then, the hybrid dilation convolution was combined with the redesigned residual structure to enhance the feature and position information extraction ability of the shallow layer of the model and optimize the feature extraction ability of the model for different scale targets. Weizhen Song et al. [12] proposed a Chinese traffic sign detection algorithm based on YOLOv4-tiny. They added an improved lightweight BECA attention mechanism module to the backbone feature extraction network of YOLOv4-tiny, an improved dense SPP network to the augmented feature extraction network, a yolo detection layer, and k-means++ clustering to obtain a priori frames more suitable for traffic sign detection. Alessandro Betti et al. [13] proposed a simple, fast, and efficient YOLO-S network that utilizes a small feature extractor, as well as skip connections via bypass and tandem, and a remodeling-pass-through layer to facilitate feature reuse throughout the network and combine low-level location information with more meaningful high-level information.

Many improvement methods have been proposed in the existing research for the problems of the YOLO algorithm in application, but the majority focus on replacing the backbone network, modifying the loss function, adding the attention mechanism, increasing the detection head, etc. Although these methods can improve the detection accuracy of YOLOv5 and the ability to detect small targets, they do not improve the ability of YOLOv5

to differentiate colors, and the results of using IoU to calculate the similarity between the bounding boxes are sensitive to small-size targets, and the IoU value varies greatly on small-size targets, so the existing methods are not a good solution for the YOLOv5s algorithm in the application of driverless formula racing. In this paper, we propose to add coordinate attention, modify color space transformation, and add a normalized Gaussian Wasserstein distance module and monocular camera distance measurement module to the YOLOv5s algorithm to address the problem of YOLOv5s in driverless formula racing car applications.

2. YOLOv5s Network Structure

YOLOv5 contains four network models: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, which increase in depth and width in turn, while the detection accuracy also increases in turn [14,15]. The study object in this article is YOLOv5s, and the structural diagram of the YOLOv5s model is presented in Figure 1. There are three networks in the model: the backbone network, the neck network, and the head network.

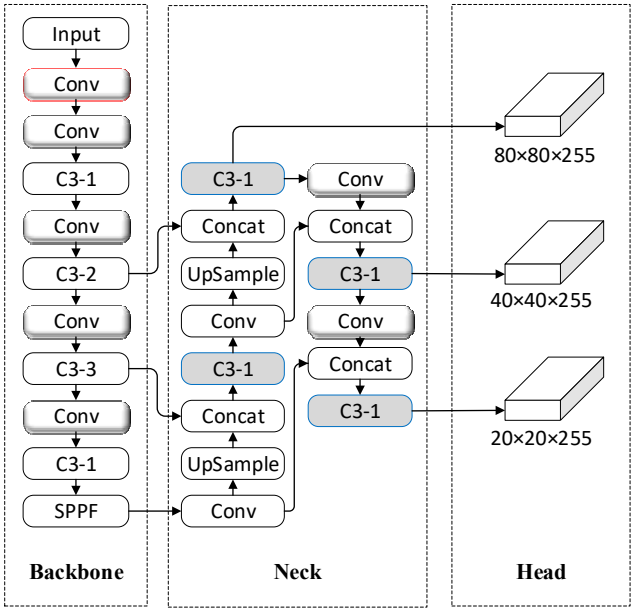


Figure 1. YOLOv5s network architecture.

The backbone network is utilized to extract the target's multi-scale characteristics by iteratively stacking the three modules: Conv, C3, and SPPF. The neck network is used to blend and combine the features and fuse the extracted features. The head network is used to predict the target classification as well as the confidence level, etc., by using three different prediction layers for predicting the features at different scales.

3. Coordinate Attention

The attention mechanism is used to increase the network's detection capabilities by giving weights to distinct objects to highlight some critical aspects while disregarding other observable information.

The concept of coordinate attention involves integrating spatial information into channel attention in order to establish their interdependence, hence improving the network's ability to accurately localize and recognize objects [16]. The architecture of the coordinate attention network is depicted in Figure 2. This network employs two 1D global average pooling operations to combine input features in the vertical and horizontal directions, resulting in two distinct directional feature maps. These feature maps are subsequently encoded as vertical and horizontal attention maps. Each attention map captures the long-range dependencies of the input feature maps along its respective direction.

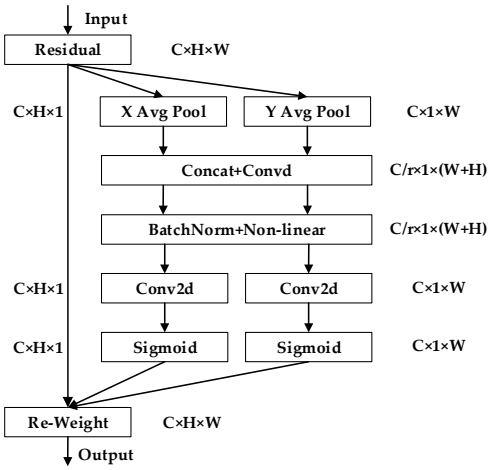


Figure 2. Network architecture of coordinate attention.

3.1. Coordinate Information Embedding

The process of encoding coordinate information is achieved by using two separate 1D averaged convolutions with pooling kernels of dimensions (H,1) and (1,W) in both the horizontal and vertical directions for each channel. Here, the output of the vertical direction in the c-th channel after 1D pooling can be expressed as:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{1}$$

Similarly, the output of the horizontal direction in the c-th channel after 1D pooling can be expressed as:

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le i < H} x_c(j, w) \tag{2}$$

The two 1D pooling operations in question perform feature aggregation in both the vertical and horizontal directions, resulting in two direction-aware feature maps. Additionally, these pooling operations enable the attention mechanism to capture distant dependencies along the respective direction, while preserving accurate positional information in the other spatial direction.

3.2. Coordinate Attention Generation

Coordinate attention generation works to connect the aggregated feature maps, starting with a 1×1 convolutional transform function, F_1, and a nonlinear activation function, δ, to encode the spatial information both horizontally and vertically. So:

$$f = \delta\left(F_1\left(\left[z^h, z^w\right]\right)\right) \tag{3}$$

Here, [,] denotes the splicing of the 1D convolution results in the vertical and horizontal directions, and z^h and z^w are the outputs of the coordinate information embedding process in the vertical and horizontal directions, respectively. After splicing, f is split along the vertical and horizontal directions, respectively, to determine f^h and f^w. Then, f^h and f^w are transformed into feature maps with the same number of channels as the input, X, using two 1×1 convolution operations, F_h and F_w, respectively, and, finally, their nonlinearity is increased using the sigmoid function, which is δ:

$$g^h = \sigma\left(F_h\left(f^h\right)\right) \tag{4}$$

$$g^w = \sigma(F_w(f^w)) \tag{5}$$

Here, g^h and g^w are the weights generated by coordinate attention, which acts on the input, X, and can be written as:

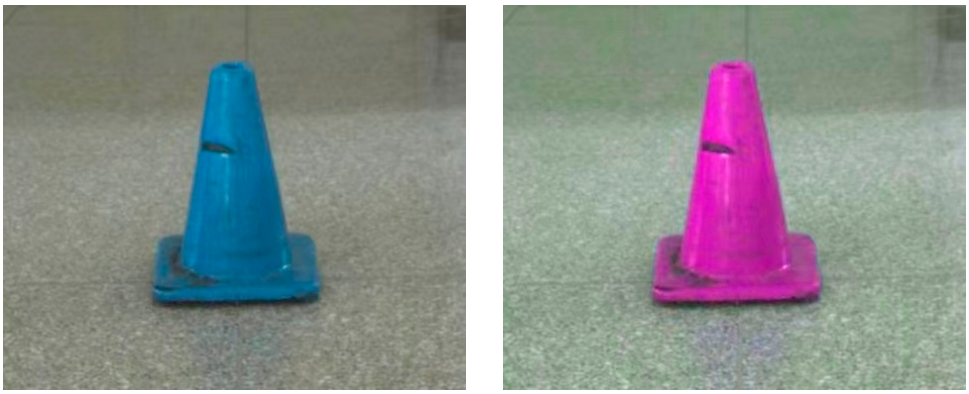
$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \quad (6)$$

The act of coordination attention serves the purpose of not only discerning the significance of various channels but also using both horizontal and vertical attention on the input feature maps. By including spatial information in the encoding process, this methodology enables precise localization of the specific item of interest, thereby enhancing the accuracy of the model's identification capabilities.

4. Modifying the Color Space Transformation Module

In the process of driverless formula car racing, the cone buckets can be mainly divided into red, blue, and yellow. In practical applications, the racing car needs to judge the color of the cone bucket to determine whether the cone bucket is on the inner side of the race road or the outer side of the race road, as well as whether the cone bucket is a change of direction, deceleration, or termination cone bucket, and so on. The racing car judges the color of the recognized cone bucket, and then controls the racing car to perform the corresponding action. Therefore, it is necessary to accurately distinguish the color of the cone bucket in the process of application.

The YOLOv5s method incorporates data augmentation techniques to enhance the performance of the model, hence mitigating the risk of training the model with a limited number of training images. In data augmentation, color space transformation is used to change the hue, saturation, and value of the training images. The hue transformation is the color transformation. Here, the procedure of the color space transformation part is extracted for demonstration and obtained, as shown in Figure 3.



(**a**) (**b**)

Figure 3. Color space transformation results. (**a**) Original images; (**b**) color space transformed image.

In Figure 3, (a) is the blue cone bucket without color space transformation, and (b) is the result of the blue cone bucket after color space transformation, from which it can be seen that the hue transformation changes the color information of the cone bucket, and the blue cone bucket is transformed into a purple cone bucket. If this part is fed into the YOLOv5s model for training, it will lead to the model's classification of the cone bucket color not being accurate enough, and the driverless formula racing car mainly relies on the color of the cone bucket to judge and perform decision-making, so this hue transformation will affect the recognition of the cone bucket, thus affecting the judgment and decision-making of the racing car. In this paper, we improve the accuracy of the YOLOv5s model in recognizing the color of cone buckets by removing the hue transformation.

5. Normalized Gaussian Wasserstein Distance

Since the objects that need to be detected by the driverless formula racing car are mainly cone buckets, and the size of the cone buckets is 20 × 20 × 30 (cm), when the cone buckets are far away from the camera, the pixel size occupied by the imaging of the cone buckets on the camera is relatively small. If we want the camera to be able to detect cone buckets at longer distances, then we need to improve the YOLOv5s algorithm for the detection of small targets.

The limited amount of information included in the tiny targets poses a challenge for the model in extracting distinguishing characteristics that identify them from other categories. Consequently, this difficulty results in a larger likelihood of mistakes occurring during the detection phase.

The YOLOv5s algorithm uses IoU to represent the similarity between the bounding boxes, but IoU is more sensitive to the size of the object, as shown in Figure 4 [17]. For a small target object, a very small change in position will cause the IoU to become very small, but for a normal-sized object, the same change will not cause a large change in the IoU.

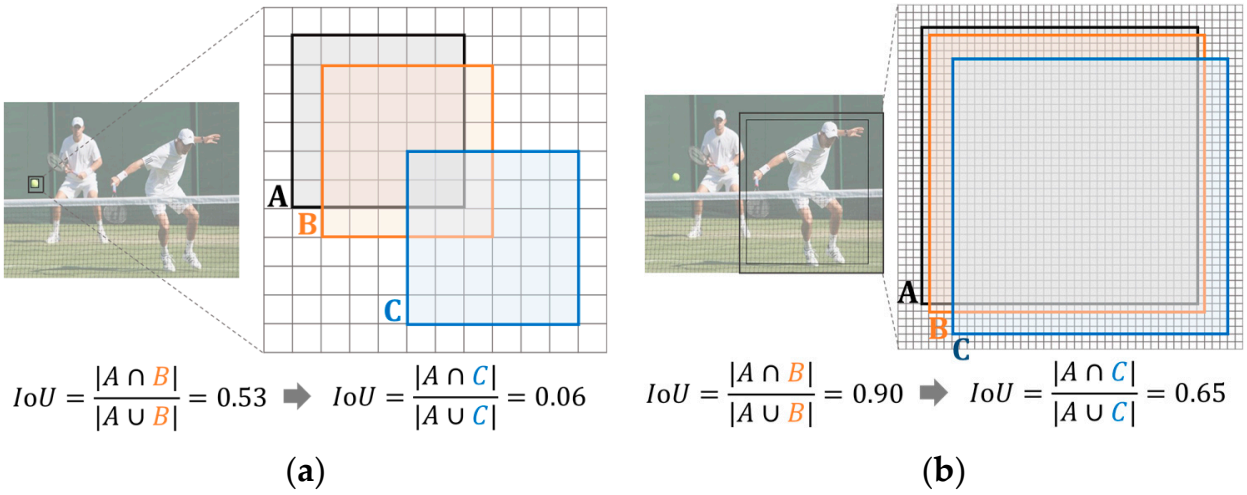


Figure 4. Sensitivity analysis of small- and normal-sized targets. (**a**) Small-sized target; (**b**) normal-sized target.

Thus, it is known that IoU is not suitable for measuring small targets. In [17], the authors measure the degree of similarity between two bounding boxes by modeling the bounding box as a Gaussian distribution and then using the Wasserstein distance instead of IoU. The advantage of using the Wasserstein distance is that the similarity between the two bounding boxes can be measured, even if they do not have overlapping parts, and the method is insensitive to small-sized targets.

In the context of target detection, it is commonly observed that the target object tends to occupy the central region of the bounding box, while the surrounding background pixels are typically concentrated along the edges of the bounding box. In order to effectively represent the relative importance of different pixels within the bounding box, a 2D Gaussian distribution is employed as a model. This distribution assigns the highest weight to the pixels located at the center of the bounding box, with the weights gradually decreasing in a descending manner as one moves toward the boundary. The bounding box, R, defined by its center coordinates (cx, cy) and its width and height (w, h), may be represented using an inner ellipse.

$$\frac{(x - \mu_x)^2}{\sigma_x^2} + \frac{\left(y - \mu_y\right)^2}{\sigma_y^2} = 1 \tag{7}$$

The center coordinates of the ellipse are denoted as μ_x, and μ_y, where μ_x represents the length of the x-axis, and μ_y represents the length of the y-axis, as in:

$$\begin{cases} \mu_x = cx \\ \mu_y = cy \\ \sigma_x = \frac{w}{2} \\ \sigma_y = \frac{h}{2} \end{cases} \tag{8}$$

The probability density function of a 2D Gaussian distribution can be calculated by the following formula:

$$f(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{2\pi|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \tag{9}$$

where $\mathbf{x}$ and $\boldsymbol{\mu}$ denote the vectors of coordinates (x, y) and means, respectively, and $\sum$ denotes the covariance matrix of the Gaussian distribution. When $\mathbf{x}$ and $\mathbf{u}$ satisfy Equation (10), this ellipse is the isodensity contour line of a 2D Gaussian distribution.

$$(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = 1 \tag{10}$$

Hence, the bounding box, R, denoted as (cx, cy, w, h), may be represented as a two-dimensional Gaussian distribution $N(\boldsymbol{\mu}, \sum)$, where:

$$\boldsymbol{\mu} = \begin{bmatrix} cx \\ cy \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix} \tag{11}$$

Thus, the correspondence between the bounding boxes, A and B, may be transformed into a distance between two Gaussian distributions.

For the two 2D Gaussian distributions, $\boldsymbol{\mu_1} = N(\mathbf{m_1}, \sum_1)$ and $\boldsymbol{\mu_2} = N(\mathbf{m_2}, \sum_2)$, the second-order Wasserstein distance between them is defined as:

$$W_2^2(\boldsymbol{\mu_1}, \boldsymbol{\mu_2}) = \|\mathbf{m_1} - \mathbf{m_2}\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma_1} + \boldsymbol{\Sigma_2} - 2\left(\boldsymbol{\Sigma_2}^{\frac{1}{2}} \boldsymbol{\Sigma_1} \boldsymbol{\Sigma_2}^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) \tag{12}$$

This can be simplified:

$$W_2^2(\boldsymbol{\mu_1}, \boldsymbol{\mu_2}) = \|\mathbf{m_1} - \mathbf{m_2}\|_2^2 + \left\|\boldsymbol{\Sigma}_1^{\frac{1}{2}} - \boldsymbol{\Sigma}_2^{\frac{1}{2}}\right\|_F^2 \tag{13}$$

where: $\|\cdot\|_F$ is the Frobenius norm.

For the Gaussian distributions, N_a and N_b, modeled by the bounding box, $A = (cx_a, cy_a, w_a, h_a)$ and $B = (cx_b, cy_b, w_b, h_b)$, the above equation can be simplified as:

$$W_2^2(N_a, N_b) = \left\| \left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2}\right]^T, \left[cx_b, cy_b, \frac{w_b}{2}, \frac{h_b}{2}\right]^T \right) \right\|_2^2 \tag{14}$$

However, $W_2^2(N_a, N_b)$ refers to the distance and cannot be used directly as a similarity measure; therefore, it needs to be transformed into an exponential form of normalization:

$$NWD(N_a, N_b) = \exp\left(-\frac{\sqrt{W_2^2(N_a, N_b)}}{C}\right) \tag{15}$$

The constant, C, is strongly correlated with the dataset.

Since the IoU-based loss function does not provide a gradient to optimize the network when there is no overlapping portion of the two bounding boxes, a loss function based on the normalized Gaussian Wasserstein distance is used in this paper:

$$L_{NWD} = 1 - NWD(N_p, N_g) \tag{16}$$

The variable, N_p, represents the Gaussian distribution of the anticipated boxes, whereas N_g represents the Gaussian distribution of the real boxes.

6. Monocular Camera Distance Measurement

6.1. Internal Reference Calibration

The camera internal reference calibration is a crucial step in determining the location of the cone bucket inside the camera coordinate system. The internal reference calibration of the camera is derived from the camera's imaging principle, which is used to compute the internal reference matrix. The camera operates based on the concept of small-hole imaging [18]. Consequently, the imaging process of the camera may be elucidated by the use of the small-hole camera model, as seen in Figure 5.

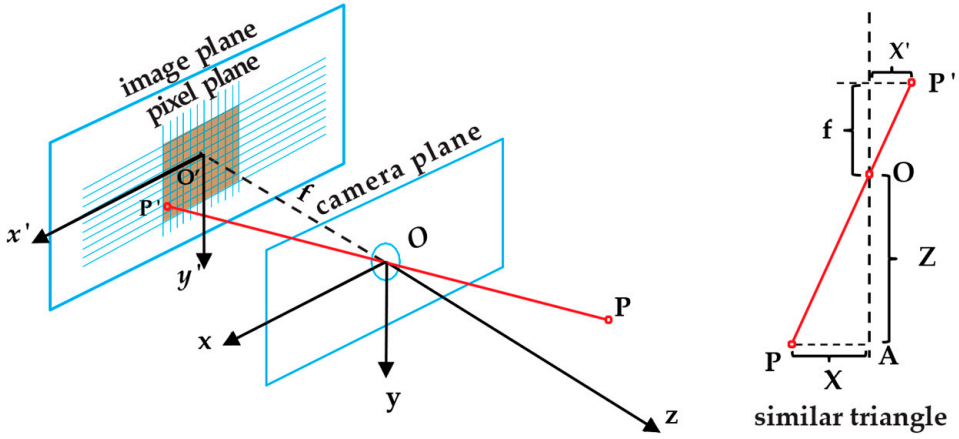


Figure 5. Small-hole camera model.

Figure 5 depicts the camera coordinate system as O-x-y-z, the image plane coordinate system as O′-x′-y′, and the projection of a point P(X,Y,Z) under the camera plane coordinate system onto the pixel plane coordinate system as P′(X′,Y′). The camera's focal length, denoted as f, is determined through the application of the geometric relationship of similar triangles.

$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'} \tag{17}$$

To make the equation more straightforward, translate the image plane in front of the camera plane, and remove the negative sign to get the following equation:

$$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'} \tag{18}$$

organized as:

$$\begin{cases} X' = f\frac{X}{Z} \\ Y' = f\frac{Y}{Z} \end{cases} \tag{19}$$

In the pixel plane coordinate system, the origin is positioned in the upper left corner of the image. The u-axis runs parallel to the *x*-axis, extending in the positive direction to the right. Similarly, the v-axis is parallel to the *y*-axis, pointing downwards in the positive direction. To acquire the coordinates (u,v) in the pixel plane coordinate system, it

is imperative to apply scaling and translation operations to the point, P′(X′,Y′), within the image plane coordinate system.

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \tag{20}$$

This is obtained by taking Equation (18) into Equation (19) and combining αf and βf into f_x and f_y, respectively:

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases} \tag{21}$$

Writing the above equation in the form of a matrix yields:

$$Z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = KP \tag{22}$$

The equation presented involves the use of the internal reference matrix, denoted as K, which may be acquired via the process of camera internal reference calibration.

6.2. Monocular Camera Distance Measurement

The YOLOv5 algorithm, which is primarily used for 2D target detection, is limited to identifying the location of the target object within an image. However, in the context of driverless formula racing, it is necessary to determine the precise position of the cone buckets within the vehicle's coordinate system. To achieve this, the algorithm must estimate the object's position in the camera coordinate system based on the image data. The specific algorithm used for this purpose is illustrated in Figure 6.

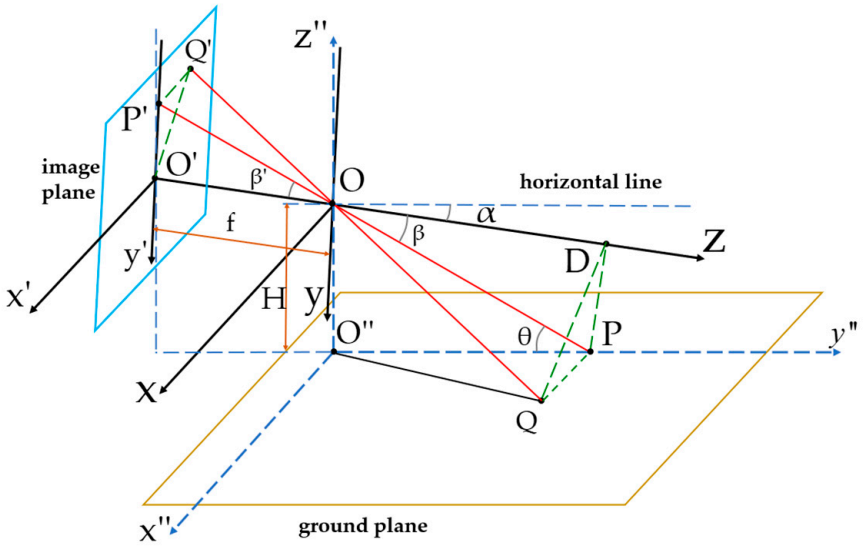


Figure 6. Principle of monocular distance measurement.

In Figure 6, O-x-y-z represents the camera coordinate system, O′-x′-y′ represents the image plane coordinate system, and O″-x″-y″-z″ represents the coordinate system of the camera coordinate system's projection on the ground. Let Q be the cone bucket's point of contact with the ground, Q′ be the equivalent point of Q in the image plane, Q′P′ be the distance of Q′ from the y′-axis, and P be the point of projection of Q on the yOz plane. OO′ is the focal length f of the camera, as in the following equation:

$$\beta' = \arctan\left(\frac{O'P'}{f}\right) \tag{23}$$

where O′P′ is the distance in the y′-axis direction of Q′ in the image plane coordinate system.

Since β′ and β are opposite angles, β′ = β, and since θ = α + β, so:

$$OP = \frac{H}{\sin(\theta)} \tag{24}$$

where α is the downward inclination of the camera relative to the horizontal line, and H is the camera's height above the ground plane, these two parameters can be obtained by measurement when the camera is installed.

The intersection point, D, is obtained by making a vertical line to the *z*-axis through point P. Here, OD is the distance of point Q from the *z*-axis in the camera coordinate system:

$$OD = OP \times \cos(\beta) \tag{25}$$

Similarly, in the camera coordinate system, DP is the distance of Q from the *y*-axis:

$$DP = OP \times \sin(\beta) \tag{26}$$

Connecting QD and Q′O′, we can see that triangle QPD and triangle Q′P′O′ are similar, so:

$$QP = \frac{DP}{O'P'} Q'P' \tag{27}$$

Here, Q′P′ represents the distance of Q′ along the x′-axis in the image plane coordinate system, and QP represents the distance of Q along the *x*-axis in the camera coordinate system.

The above calculations are then used to compute the exact 3D position, XYZ, of the cone buckets in the camera coordinate system.

7. Experimental Verification

7.1. Dataset and Experimental Environments

The dataset is collected mainly based on the actual application scenarios. The detection target of the driverless formula racing car is the cone buckets, whose main difference is the different colors, namely red, blue, and yellow. The different colors of the cone buckets are placed in different positions; the red cone buckets are in the outer circle of the racing road, the blue cone buckets are in the inner circle of the racing road, and the yellow cone buckets are in the starting and ending positions. The racing road is divided into three types of racing road: straight-line acceleration, 8-word surround, and high-speed racing. We create the racing track environment according to the above requirements, capture it with real cars, and then label the position of the cone buckets on the image; the different color cone buckets are distinguished by different labels. The size of the images in this dataset is 1920 × 1200, and there are 5652 images in total; the training, validation, and test sets are divided into three ratios of 5:2.5:2.5, 6:2:2, and 8:10:10, and the experimental results of the three ways of dividing the dataset are averaged. The models in this experiment were trained and tested using a GeForce RTX3060 graphics card with CUDA version 11.3 and PyTorch version 1.12.1.

7.2. Experimental Methods

The methodology of the experiment is to verify the enhancement of the method proposed in this paper with respect to YOLOv5s through comparative experiments. First, a comparison is made in terms of performance parameters, and the YOLOv5s network model—after adding and modifying each module—is compared with the original YOLOv5s network in an ablation test to observe the enhancement of performance parameters. Then, the effectiveness of adding and modifying each module is verified through real vehicle experiments, and the results of the experiments are shown through visualization.

7.3. Experimental Evaluation Indicators

The accuracy and real-time performance of the algorithm's detection are the most important indicators for the application of this algorithm to driverless formula racing cars; therefore, in this paper, we have selected evaluation indicators related to the accuracy and real-time performance of the detection. These include Precision, Recall, mean Average Precision (mAP), Parameter Number (Pa) of the model, and Frames Per Second (FPS).

Precision is defined as the rate of True Positives detected among all Positives detected.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{28}$$

Recall is the rate of detected Positives among all ground true positives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{29}$$

Mean Average Precision (mAP) is the ratio of Average Precision (AP) to the number of categories, k, whereas Average Precision is the region below the Precision–Recall curve.

$$\text{mAP} = \frac{\sum_{i=1}^{k} \text{AP}_i}{k} \tag{30}$$

The Parameter Number (Pa) is used to determine the scale of the model and is computed for convolutional layers as follows:

$$\text{Pa} = \text{C}_{\text{in}} \times \text{C}_{\text{out}} \times \text{K}_{\text{h}} \times \text{K}_{\text{w}} \tag{31}$$

For the fully connected layer, the formula is as follows:

$$\text{Pa} = \text{C}_{\text{in}} \times \text{C}_{\text{out}} \tag{32}$$

where C_{in}, C_{out}, K_h, and K_w stand for the number of input channels, the number of output channels, the height, h, of the convolution kernel, and its width, w, respectively.

The Frames Per Second (FPS) of the output is used to determine the speed of model detection; the higher the FPS, the quicker the model is detected.

7.4. Ablation Experiment

To assess the efficacy of incorporating coordinate attention (CA), modifying color space transformation (HSV), and integrating the Normalized Gaussian Wasserstein Distance module (NWD) into the YOLOv5s algorithm as proposed in this study, ablation experiments were conducted. These experiments aimed to evaluate the influence of modifying various modules on enhancing the detection performance of YOLOv5s within a consistent experimental setting. The YOLOv5s model was used as the baseline model for this experiment. The experiment utilized a self-created cone bucket dataset. Each algorithm is run 30 times, and the average result of these 30 runs is calculated and displayed in Table 1.

Table 1. Comparison of ablation experiments.

Model	Precision/%	Recall/%	mAP/%	Pa/10^6	FPS
YOLOv5s	86.2	88.6	91.6	7.018	55.6
YOLOv5s-CA	89.3	90.9	94	7.043	51.5
YOLOv5s-HSV	90.6	91.3	94.2	7.018	55.6
YOLOv5s-NWD	92	87.7	94.5	7.018	55.6
YOLOv5s-CA-HSV	90.5	92.5	94.5	7.043	51.5
YOLOv5s-CA-HSV-NWD	93.1	93	96.5	7.043	51.5

Upon analyzing the data presented in Table 1, it can be seen that increasing any part alone will improve the detection performance of YOLOv5s, while adding three modules will enhance the detection accuracy even more. Although increasing the coordinate attention will lead to an increase in the parameter number and a decrease in the detection rate, the decrease in the detection rate is very small, and it will not have any effect on the detection rate required by the driverless formula racing car.

To evaluate the effectiveness of the method proposed in this paper, our method is compared with the YOLOv3, YOLOv4 [3], EfficientDet [19], SSD [20], and Faster R-CNN [21] methods in Table 2, each of which is also run 30 times and averaged.

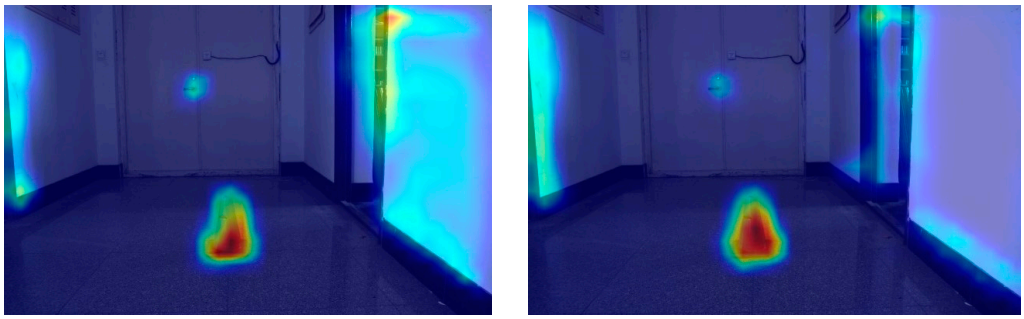
Table 2. Comparison of different models.

Model	Precision/%	Recall/%	mAP/%	Pa/10^6	FPS
YOLOv3	87.8	77.2	86.5	63.0	35.6
YOLOv4	88.1	88.9	91.7	64.4	30.4
YOLOv5s	86.2	88.6	91.6	7.018	55.6
EfficientDet	87.3	87.8	92.1	21	15.4
SSD	88.7	90.4	93.4	26.2	15.3
Faster R-CNN	86.3	88.2	91.2	137	3
Ours	93.1	93	96.5	7.043	51.5

From the above table, it can be seen that the model proposed in this paper improves the attention to the detection target due to the addition of coordinate attention and the optimization of the IOU loss function for small target objects, so the model proposed in this paper shows a great improvement in the detection accuracy compared to other target detection models. Since the number of parameters of YOLOv5s itself is very small compared to other algorithms, it has a great advantage in detection speed, while the method in this paper improves the detection accuracy of YOLOv5s. Although it leads to a very small decrease in detection speed, its detection speed is still much higher than that of other detection algorithms. In conclusion, the method proposed in this paper is superior to other detection algorithms in terms of detection speed and detection accuracy.

7.5. Coordinate Attention Comparison

In order to verify the effect of adding coordinate attention to the target detection effect, the images were collected and detected using the YOLOv5s model with added coordinate attention, the heat map was output using Grad-CAM, and the results obtained are shown in Figure 7.



(**a**) (**b**)

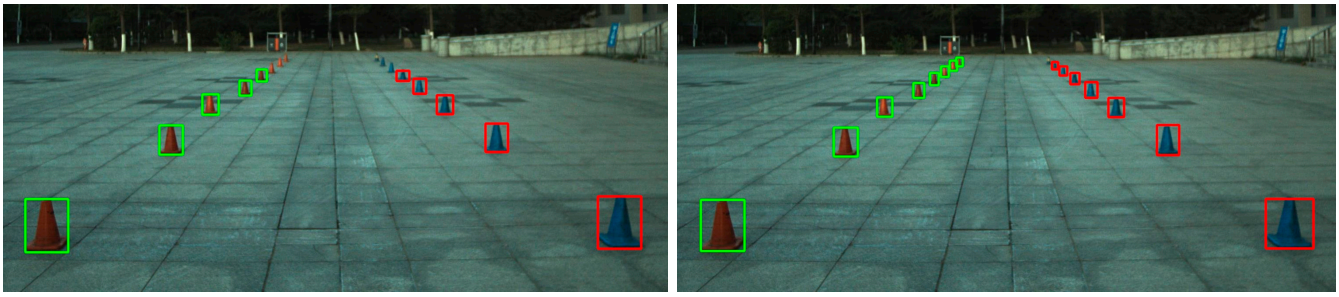
Figure 7. Comparison of thermograms of YOLOv5s and YOLOv5s–coordinate attention. (**a**) Heat map of YOLOv5s; (**b**) heat map of YOLOv5s–coordinate attention.

In Figure 7, (a) is the heat map generated using YOLOv5s detection, and (b) is the heat map with added coordinate attention; the warmer the hue of the heat map, the higher

the model's attention to the area. From the heat map, it can be seen that the heat map corresponding to the cone bucket after adding coordinate attention has a warmer hue and covers a wider area of the cone bucket compared to the heat map without adding coordinate attention. This shows that the more attention the model pays to the pixels corresponding to the cone bucket, the higher the possibility that these pixels are the objects that need to be detected, and it can be concluded that the addition of coordinate attention improves the accuracy of the model in detecting the cone bucket.

7.6. Small Target Detection Experiment

The purpose of adding normalized Gaussian Wasserstein distance is to improve the accuracy of YOLOv5's detection of small targets; therefore, the effect of YOLOv5s in detecting small targets after adding the normalized Gaussian Wasserstein distance module is verified by comparing the normal YOLOv5s model and YOLOv5s model with normalized Gaussian Wasserstein distance added, and the results are shown in Figure 8.



(**a**) (**b**)

Figure 8. Comparison of the detection effect of YOLOv5s with YOLOv5s with the addition of normalized Gaussian Wasserstein distance. (**a**) Detection effect of YOLOv5s; (**b**) detection effect of YOLOv5s with added normalized Gaussian Wasserstein distance.

In Figure 8, (a) is the effect of YOLOv5s model detection without the addition of normalized Gaussian Wasserstein distance; five red cone buckets and five blue cone buckets are detected; (b) is the YOLOv5s model with the addition of normalized Gaussian Wasserstein distance; all the eight red cone buckets are detected, and seven blue cone buckets are detected. The comparison shows that the YOLOv5s model with the addition of the normalized Gaussian Wasserstein distance is more effective in the detection of small targets.

7.7. Monocular Camera Distance Measuring Experiment

In this paper, the monocular distance measurement module is added to the YOLOv5s model in order to realize the use of a monocular camera to obtain the accurate positioning of the cone buckets in the camera coordinate system. The specific experiments are conducted by placing five cone buckets on the left and right sides of the race car, with the blue cone buckets on the left side placed at 2.4 m intervals, the red cone buckets on the right side aligned with the blue cone buckets, and the left and right side of the cone buckets spaced at 3 m intervals. Then, YOLOv5s is used with monocular distance measurement added to detect the position of the cone buckets, and the result is shown in Figure 9b. The calculated cone bucket position is printed into the terminal, and the cone bucket position is shown in the contents of the red box in Figure 9.

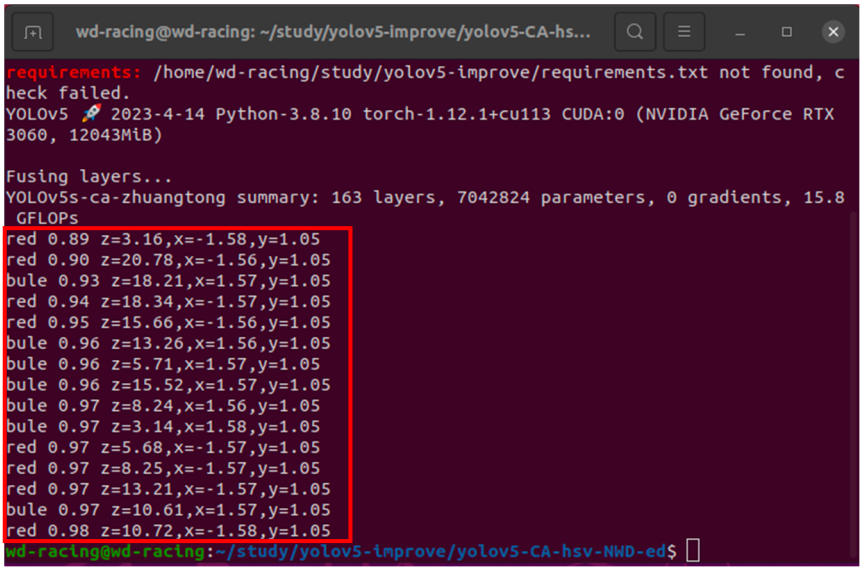


Figure 9. Monocular camera distance measurement results.

The results are analyzed by comparing the above-detected position with the actual position and then calculating the relative error [22] by the following equation:

$$\delta = \frac{|\mathrm{pd} - \mathrm{gt}|}{\mathrm{gt}} \times 100\% \tag{33}$$

where δ denotes the relative error, pd denotes the distance of the cone bucket detected by the algorithm, gt denotes the actual distance, and $|\,\mathrm{pd} - \mathrm{gt}\,|$ denotes the absolute error.

The calculated maximum error of the cone bucket in the camera coordinate system within 20m in the Z direction is 5.64%, and the average error is 4.96%; the maximum error in the X direction is 5.33%, and the average error is 4.62%. Due to the internal parameter calibration, there will be a certain error, and the picture taken by the camera will produce an aberration. Although the aberration will be corrected to a certain extent, there will still be an error, so while the camera detects the position of the object itself with an error, according to the effect of the test of the real car, the error of the method is within the permissible range.

8. Discussion

In this paper, for the problems of the YOLV5s algorithm applied in driverless formula racing, we propose to add a coordinate attention mechanism, modify the color space transformation module, add a normalized Gaussian Wasserstein distance module, and a monocular ranging module into the YOLOv5s algorithm to improve the YOLOv5s algorithm. Adding the coordinate attention mechanism can improve the YOLOv5s algorithm's attention to the cone buckets and reduce their attention to other information to improve the accuracy of the model's detection of the cone buckets; modifying the color space transformation module so that the YOLOv5s algorithm does not transform the color when data enhancement is carried out improves the algorithm's ability to distinguish the color; adding the normalized Gaussian Wasserstein distance module adopts the Wasserstein distance instead of the IoU to measure the similarity between two bounding boxes, which reduces the sensitivity of the IoU to the size of the object, and improves the detection capability of YOLOv5s for small targets; adding the monocular ranging module enables ables it to locate the position of the cone buckets under the camera coordinate system. The experiments show that our proposed method can effectively solve the problems in practical applications.

Although many researchers have carried out corresponding research for the above problems, it has been for structured road vehicles, in which the detection targets are pedestrians and vehicles. For the detection of small target cone buckets by driverless

formula racing cars, there is very little research, so this paper provides a certain reference significance for designers and enthusiasts of driverless formula racing cars.

9. Conclusions

This paper proposes a monocular camera-based cone bucket detection algorithm based on the YOLOv5s algorithm for driverless formula racing cars. The detection accuracy of the algorithm is improved by using coordinate attention in YOLOv5s. In order to better distinguish the color of the cone buckets, this work turned off the hue transformation during data enhancement. To improve the detection of distant cone buckets in YOLOv5s, normalized Gaussian Wasserstein distance was used as a loss function. Finally, to obtain the real position of the cone bucket, the principle of similar triangles was used to calculate the position of the cone bucket under the camera coordinate system. By training and testing on the homemade cone bucket dataset, it can be seen that the proposed algorithm improves the precision rate by 6.9%, the recall rate by 4.4%, and the mean average precision by 4.9% relative to YOLOv5s, and it is able to satisfy practical uses, although the detection frame rate has decreased. Monocular camera distance measurement has a maximum error of 5.64% within 20m in the Z direction and 5.33% in the X direction in the camera coordinate system.

The improvement in the above YOLOv5s algorithm can be applied in the intelligent driving car to perceive the environment around the car, and this improvement method can make the intelligent car "see more accurately", "see farther", and also locate the position of obstacles, so that the intelligent car can execute the corresponding decision in advance, which can significantly reduce the occurrence of traffic accidents.

Although the algorithm has significantly improved the detection accuracy and can meet the real-time requirements, it still has the problem of a large number of parameters, which occupy a large computational memory, so simplifying the network of YOLOv5s through techniques such as network pruning and knowledge distillation will be the focus of our next research.

Author Contributions: Conceptualization, G.L., Y.H. and J.Z.; methodology, J.Z.; software, J.Z.; validation, J.Z., G.L. and Y.H.; formal analysis, J.Z.; investigation, J.Z.; resources, G.L.; data curation, J.Z., Y.H.; writing—original draft preparation, J.Z.; writing—review and editing, J.Z., G.L. and Y.H.; visualization, J.Z.; supervision, G.L. and Y.H.; project administration, G.L.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation General Program of China (51675257); and Liaoning Provincial Natural Science Foundation General Program (2022-MS-376).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
2. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
3. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors* **2022**, *22*, 464. [CrossRef] [PubMed]
4. Zhang, Y.; Guo, Z.; Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-Time Vehicle Detection Based on Improved YOLO v5. *Sustainability* **2022**, *14*, 12274. [CrossRef]
5. Carrasco, D.P.; Rashwan, H.A.; García, M.Á.; Puig, D. T-YOLO: Tiny vehicle detection based on YOLO and multi-scale convolutional neural networks. *IEEE Access* **2023**, *11*, 22430–22440. [CrossRef]
6. Zhang, Y.; Hou, X.; Hou, X. Combining Self-Supervised Learning and Yolo v4 Network for Construction Vehicle Detection. *Mob. Inf. Syst.* **2022**, *2022*, 1–10. [CrossRef]
7. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PLoS ONE* **2021**, *16*, e0259283. [CrossRef] [PubMed]

8. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–25 June 2021; pp. 2778–2788.
9. Strbac, B.; Gostovic, M.; Lukač, Ž.; Samardzija, D. YOLO Multi-Camera Object Detection and Distance Estimation. In Proceedings of the 2020 Zooming Innovation in Consumer Technologies Conference, Novi Sad, Serbia, 26–27 May 2020; pp. 26–30.
10. Karthika, K.; Adarsh, S.; Ramachandran, K.I. Distance Estimation of Preceding Vehicle Based on Mono Vision Camera and Artificial Neural Networks. In Proceedings of the International Conference on Computing, Communication and Networking Technologies, Kharagpur, India, 1–3 July 2020; pp. 1–5.
11. Liu, Z.; Gao, Y.; Du, Q.; Chen, M.; Lv, W. YOLO-Extract: Improved YOLOv5 for Aircraft Object Detection in Remote Sensing Images. *IEEE Access* **2023**, *11*, 1742–1751. [CrossRef]
12. Song, W.; Suandi, S.A. TSR-YOLO: A Chinese Traffic Sign Recognition Algorithm for Intelligent Vehicles in Complex Scenes. *Sensors* **2023**, *23*, 749. [CrossRef] [PubMed]
13. Betti, A.; Tucci, M. YOLO-S: A Lightweight and Accurate YOLO-like Network for Small Target Selection in Aerial Imagery. *Sensors* **2023**, *23*, 1865. [CrossRef] [PubMed]
14. Dong, X.D.; Yan, S.; Duan, C.Q. A lightweight vehicles detection network model based on YOLOv5. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104914. [CrossRef]
15. Kasper-Eulaers, M.; Hahn, N.; Berger, S.; Sebulonsen, T.; Myrland, Ø.; Kummervold, P. Short Communication: Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5. *Algorithms* **2021**, *14*, 114. [CrossRef]
16. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
17. Wang, J.; Xu, C.; Yang, W.; Yu, L. A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. *arXiv* **2021**, arXiv:2110.13389.
18. Yan, G.; Zhuochun, L.; Wang, C.; Shi, C.; Wei, P.; Cai, X.; Ma, T.; Liu, Z.; Zhong, Z.; Liu, Y.; et al. OpenCalib: A Multi-sensor Calibration Toolbox for Autonomous Driving. *arXiv* **2022**, arXiv:2205.14087. [CrossRef]
19. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
22. Mao, J.; Wei, H.; Sheng, W. Target distance measurement method using monocular vision. *IET Image Process.* **2020**, *14*, 3181–3187.