CoordLight-YOLOv5: A Lightweight Object Detection Algorithm for Autonomous Motorsport

Virgil (Zhihao) Xing, Liang Zhao, Graeme Best
University of Technology Sydney, Australia
zhihao.xing-1@student.uts.edu.au, {liang.zhao, graeme.best}@uts.edu.au

Abstract

Autonomous motorsport is a rapidly developing field, among which autonomous racing based on cone tracks is a challenging testing and racing environment. Object detection is one of its indispensable technologies. We propose an improved lightweight model, CoordLight-YOLOv5, motivated by autonomous racing projects that require a high frame rate and detection accuracy by onboard embedded computing. In the proposed method, we deleted large object detection feature maps from the backbone and neck network in order to reduce model training and detection time and reduce model computational costs. Furthermore, we introduce CoordConv layer in the neck network to enhance object localisation ability in Cartesian coordinates without using spatial transformations. Our experiments with five autonomous racing datasets show that compared to YOLOv5s, the average number of model parameters is reduced by 91.5%, the average floating-point operation is reduced by 78%, the detection speed has been increased from the original 94 frames per second to 117 frames per second, and the average detection accuracy decreased by only 2.5%.

1 Introduction

The progress in autonomous motorsport vehicles heavily relies on the combination of object recognition and precise depth estimation, especially when navigating tracks defined by traffic cones. Discerning the exact confines of the track and optimising the path of the car is crucial for success. To achieve this, it is necessary to accurately estimate the precise location and distance of the road markers. The integration of advanced computer vision methodologies with state-of-the-art sensor technologies offers high detection and depth precision, which is essential for maintaining safety and achieving high racing performance [Katare *et al.*, 2023].

YOLOv5 [Jocher, 2020] has demonstrated high real-time detection speed and accuracy in autonomous racing, enabling the car to make decisions instantly and accurately avoid obstacles [Wu *et al.*, 2021]. Its lightweight design makes it easy to deploy in embedded systems with limited resources, and it is adaptable enough to be fine-tuned specifically for racing environments. The combination of these features ensures the safety and efficiency of the car on the track, making YOLOv5 an ideal choice for autonomous racing.

In autonomous driving motorsports, balancing accuracy and efficiency in object detection models is a critical challenge. High performance is essential for safety and competitiveness, as decisions made in fractions of a second are crucial. However, the models must also be lightweight due to the limitations of the onboard hardware systems in race vehicles. These limitations include factors such as weight, power consumption, and heat dissipation. Overly complex models can cause processing delays or system failures, which are detrimental in high-speed races. Therefore, developing an efficient model without sacrificing accuracy is a key task in the evolving field of motorsport automation.

For these challenges, we propose the concept of a new network, CoordLight-YOLOv5, which extends the commonly used YOLOv5 method. There are two main changes we made to standard YOLOv5. Firstly, to ensure the model is lightweight enough to meet the requirements for autonomous motorsport target detection, we cut out the largest feature map so that the model more efficiently detects smaller objects in the image. We also introduce CoordConv [Liu *et al.*, 2018] into the model's neck network to further enhance the model's ability to localise objects in the Cartesian coordinate system, eliminating the need for spatial transformation and improving accuracy. We are primarily interested in developing this model for our UTS autonomous motorsport team; one of our competition cars is pictured in Figure 1.

Figure 1: UTSME22: Ramona, UTSME's Seventh electric vehicle, built from the ground up to participate in the Formula SAE Australasia competition [UTSME, 2022]. The proposed computer vision algorithms are designed to enable autonomous operation of our motorsport vehicles.

We perform experiments with five datasets of image sequences of traffic cones recorded onboard motorsport vehicles, pictured in Figures 2 and 3. In comparison to YOLOv5s, our optimised model, CoordLight-YOLOv5, significantly reduces the number of parameters by 91.5%. It also shows a 78% reduction in floating-point operations, which results in improving detection rates from an original 94 fps to a faster 117 fps. Our model has a decreased accuracy of only 2.5%, despite these significant efficiency improvements that are necessary for the context of autonomous motorsport.

2 Related Work

Object detection is crucial for autonomous cars, with significant recent advancements promising to transform both motorsport and everyday driving. Particularly in autonomous motorsports, this technology can precisely navigate traffic cone-based circuits for high-speed races [Ess *et al.*, 2010]. The two-stage object detection such as Faster R-CNN and Mask R-CNN first uses region generation algorithms such as Selective Search to generate a series of candidate regions based on the regions in the image that may contain targets. These candidate regions are image regions of different sizes and shapes, which may contain target objects of interest. Then, it classifies and regresses the candidate regions. However, it has the drawbacks of slow detection speed and high memory consumption, which makes it unsuitable for real-time detection for autonomous driving [Bharati and Pramanik, 2020]. In contrast, single-stage detectors, such as YOLO and SSD, offer both accuracy and speed, meeting real-time autonomous driving demands.

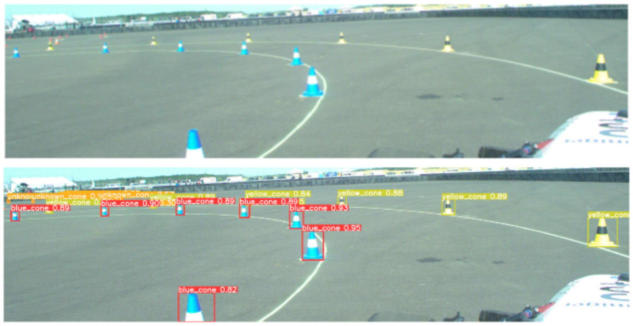


Figure 2: A labelled image from one of the autonomous motorsport datasets that we use to evaluate our method [sma, 2022].

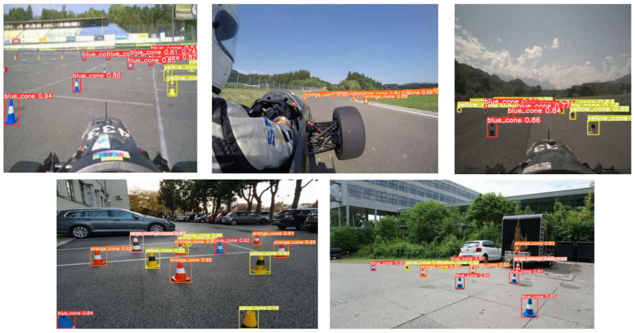


Figure 3: Sample images with object detection result by using our proposed CoordLight-YOLOv5.

Previously, YOLOv3 was widely used in autonomous driving related tasks [Choi *et al.*, 2019; Adarsh *et al.*, 2020; Zhao and Li, 2020]. YOLOv3 uses a new convolutional network structure called Darknet-53. This network has a deeper level than the previously used Darknet-19, allowing it to better detect small objects while maintaining fast computation [Redmon and Farhadi, 2018]. YOLOv4 has made further improvement by using a new convolutional neural network structure called CSPMarknet53, which has significantly improved its speed and performance compared with its predecessor [Bochkovskiy *et al.*, 2020; Wu *et al.*, 2020; Wang *et al.*, 2021]. Although YOLOv5 has not made improvements to the backbone network, it integrates many advanced methods in the field of computer vision [Zhu *et al.*, 2021; Jocher *et al.*, 2022]. The implementation of YOLOv5 is based on pytorch, and its training and reasoning process is designed to be very simple, making it easier to use. And YOLOv5 provides four pre-trained models (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x) to adapt to different computing power and detection needs. Although there have been new YOLO versions released in recent years, such as YOLOv6, YOLOv7, and this year's new YOLOv8, YOLOv5 has gained popularity in the field of autonomous driving due to its mature architecture, ease of use, flexibility, and excellent speed and accuracy [Wu *et al.*, 2021; Jocher *et al.*, 2022]. Motivated by all of the above work, we also work with YOLOv5, but make several improvements to the model to make it more efficient in the context of detecting cones in autonomous motorsport.

According to the size and complexity of the model, YOLOv5 can be divided into YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. YOLOv5x is the largest and most accurate variant, but it is also the slowest. Typically, YOLOv5x is used in scenarios where precision is paramount, such as scientific research or applications that require the highest level of accuracy. The YOLOv5s variant is the smallest and fastest variant, with the goal of achieving efficient operation on devices with limited resources. Although its accuracy may be lower than other variants, it is very suitable for mobile devices and edge computing in terms of speed and accuracy. These characteristics motivate us to focus on the YOLOv5s variant for the context of autonomous motorsport.

3 Methods

Our goal is to perform object detection of traffic cones that define the edge of the race track. An example of this object detection is illustrated in Figure 2. Typically, an image contains many cones that all need to be detected simultaneously. The size of each cone in the image is relatively small, which motivates us to focus on having a high accuracy on detecting smaller objects. Also, we are interested in performing this cone detection at a high frame rate so that the path planner can react quickly and enable the car to drive at high speeds.

To optimise object detection for autonomous motorsport, we delve into the complexity and potential of the YOLOv5s architecture, aiming to shape it into a more efficient yet equally powerful variant. This section begins by summarising the baseline YOLOv5s method's network structure that contributes to its strong performance. We also summarise the CoordConv structure proposed by [Liu *et al.*, 2018] which solves translation invariance. We then propose our new method, named CoordLight-YOLOv5, which makes several modifications to YOLOv5s, including adding the CoordConv technique to enhance position-based feature learning, and improving computational efficiency by cutting out feature maps that are unnecessary for the context of autonomous motorsport.

3.1 Baseline Method: YOLOv5s

The structure of YOLOv5s is divided into input, backbone, neck and detection head. Input includes mosaic data augmentation, image adaptive scaling and adaptive calculation of anchor frame. The backbone network includes CBS, C3 module and spatial pyramid pooling-fast (SPPF) [He *et al.*, 2015]. CBS consists of convolutional layer, batch normalisation layer (BN) and sigmoid linear unit activation function (SiLU). The BN can solve the problems of internal covariate offset, gradient disappearance and gradient explosion, and can improve the rate of convergence of the model and provide weak regularisation effect. SiLU is a nonlinear activation function, which can be used to capture and represent complex patterns in data, and it has smooth derivatives throughout the Domain of a function, and it is conducive to optimisation and training process. The C3 in the backbone consists of CBS and residual structure. The SPPF is first passed into the CBS layer, followed by three MaxPool layers of 5x5 size for serial calculation, followed by concat operation, and finally CBS operation again. The SPPF transforms feature maps with unfixed scales into unified scales and integrates multi-scale features.

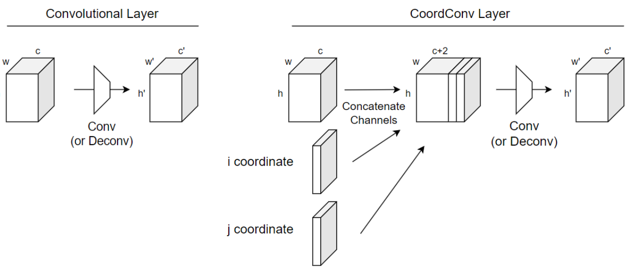


Figure 4: The CoordConv layer structure [Liu *et al.*, 2018], which we use to enable translation variance.

The neck network combines the backbone network structure with the intermediate feature extraction network to enhance the feature expression ability and Receptive field. The neck network is mainly composed of Path Aggregation Network (PANet) [Wang *et al.*, 2019] and CIOU loss. PANet is a feature pyramid structure used to obtain and fuse features at different scales. This enables the model to detect both large and small objects simultaneously. The CIOU loss function is a standard loss function for YOLOv5 [Jocher, 2020] that considers the shape, location and size of the bounding box to provide better training stability and performance.

3.2 Background: CoordConv

Traditional CNNs have translation invariance, which means that no matter where a feature appears in the input, the network will process it in the same way. However, in the context we consider, typically cones will be in particular regions of the image, and performance could be improved by exploiting this knowledge.

One way to include variance of the model when translating over the image is to incorporate the CoordConv

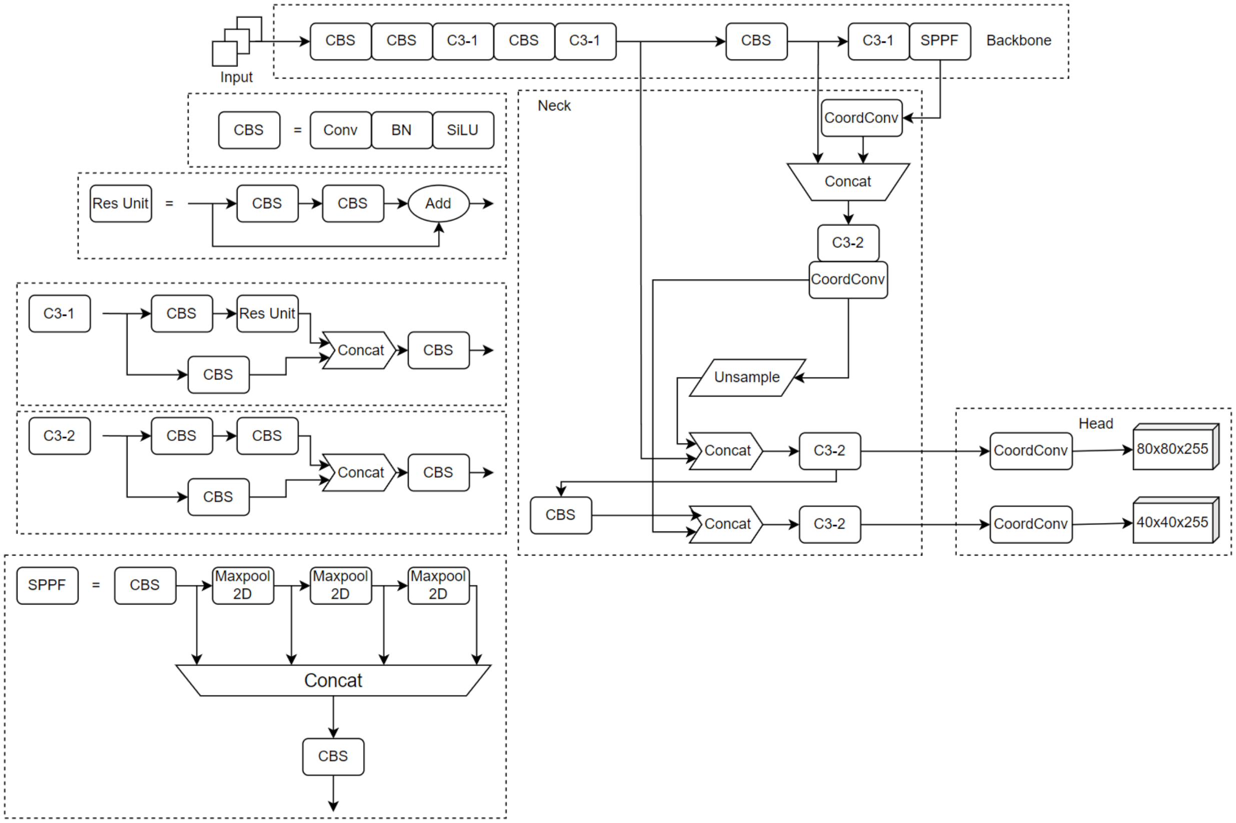


Figure 5: Our proposed Coordlight-YOLOv5 network structure, which extends YOLOv5s v6.0 [Jocher, 2020].

layer structure [Liu *et al.*, 2018]. As shown in Figure 4, CoordConv adds additional channels to the input feature map, which contains the x and y coordinate information of each pixel. In order to make the coordinate information meaningful on feature maps of different sizes, coordinates are usually normalised. Afterwards, feature maps with coordinate channels will be fed into traditional convolutional layers. Since the input now contains spatial coordinate information, the convolutional layer can learn position-based features. In the proposed method, we introduce CoordConv to solve the coordinate transformation problem of input images.

3.3 Proposed Network: CoordLight-YOLOv5

In this research, to make the object detection model applied to the autonomous motorsport as lightweight as possible to cope with performance-limited embedded hardware, and balance the accuracy reduction caused by lightweight, we introduce an innovative network, as visualised in Figure 5, termed CoordLight-YOLOv5. This design's primary objective revolves around achieving significant reductions in model size, ensuring that there's minimal compromise on its accuracy. Drawing structural parallels with YOLOv5, the CoordLight-YOLOv5 architecture can be conceptually segmented into three principal components: the backbone, neck, and head.

Lightweight Network Architecture

Starting with its foundational layer, the backbone, the architecture is designed to harness and channel the fundamental features essential for object detection. However, it's in the successive segments, the neck and the head where the model departs notably from YOLOv5s. Specifically, to shed model heft, the P5 feature maps, typically present in these sections, are deliberately omitted. This strategic exclusion is pivotal in accentuating the model's lightweight character.

More specifically, the output of each layer in YOLOv5 can be considered as a feature map. These feature maps gradually capture more abstract and advanced features as the network depth increases. The five-layer feature maps from p1 to p5 in YOLOv5 have ratios of 1:2, 1:4, 1:8, 1:16, and 1:32 to the original image, respectively. The resolution of each layer is lower than that of the previous layer, but the receptive field is larger. P5 is the highest level and largest scale feature map in YOLOv5, with a scale of 1/32 of the original image. This usually means that P5 feature maps are useful for detecting larger objects, while smaller objects may be better detected on higher resolution feature maps such as P3 or P4. However, by analysing that the dataset is mainly composed of small-sized objects, the P5 feature map does not play a major role. Therefore, we propose to remove the P5 feature map from the YOLOv5s network structure to reduce computational complexity and model size without significantly reducing accuracy [Liu *et al.*, 2022].

Adding CoordConv

To further optimise the model, we borrowed insights from the seminal work on PP-YOLO [Long *et al.*, 2020]. Our experiments highlighted potential areas for enhancements. Consequently, the 1×1 convolution layer in the Feature Pyramid Network (FPN) and the inaugural convolution layer in the detection head have been replaced with CoordConv layers. This substitution not only effectively manages the increase in Floating Point Operations Per Second (FLOPS), but also optimises the model parameters. By integrating CoordConv, the model reaps the benefits of improved spatial location capabilities, thereby enhancing overall performance metrics.

4 Experiments

To comprehensively assess the robustness and versatility of our proposed CoordLight-YOLOv5 approach, we carried out a series of experiments including a comparative study with YOLOv5, an ablation analysis, a generality assessment, and benchmarking against state-of-the-art.

The hardware configuration included a Intel Xeon Gold 6238R 2.2GHz 28cores CPU, 180GB RAM, and an NVIDIA Quadro RTX 6000 Passive GPU. The training was performed with Linux, Python 3.8, PyTorch 1.8, CUDA 10.2, and CUDNN 8.2. Kati 9K Dataset trains for 400 epochs, formulastudent for 400, Cone Detection Dataset for 400, Cone Dataset for 500, and Fsoco Dataset for 500. Stochastic gradient descent is used for training, updating and optimising the weights.

4.1 Evaluation Indicators

The performance of the model will be evaluated based on three evaluation indicators: precision (P), recall rate (R) and mean average precision (mAP), as following the standard definitions.

Model complexity is evaluated by parameters floating-point operations (FLOPs), and model size, defined as

$$\text{Parameters} = C_{\text{in}} \times C_{\text{out}} \times K \times K + C_{\text{out}}, \quad (1)$$

$$\text{FLOPs}_{\text{conv}} = 2 \times H' \times W' \times C_{\text{in}} \times C_{\text{out}} \times K \times K, \quad (2)$$

and

$$\text{FLOPs}_{\text{pool}} = H' \times W' \times C \times K \times K, \quad (3)$$

where H' and W' are the height and width of the output feature map, C_{in} is the number of input channels, C_{in} is the number of output channels, and K is the size of the convolution kernel.

4.2 Results

Comparison to YOLOv5

YOLOv5 can be divided into YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x according to the size and complexity of its network structure. We did a comparative study on them on Kati 9K Dataset as shown in Table 1. YOLOv5s is one of the smallest models and achieves excellent detection speed and accuracy. Although YOLOv5n has a smaller size, its accuracy cannot meet the requirements. These results motivated why we chose to extend the YOLOv5s variant.

Ablation Study

The ablation study aims to evaluate the importance of each part of the model and its impact on performance. We also showed the impact of CoordConv on performance when applied at different locations in the network. We conducted six experiments in total on Kati 9K Dataset as shown in Table 2 and Figure 2, including: YOLOv5s, Coord-YOLOv5 introduces CoordConv in the 1×1 convolution layer of FPN and the first convolution layer of detection head, CoordBB-YOLOv5 introduces CoordConv in backbone network, Light-YOLOv5 removes the largest feature map, and Coordlight-YOLOv5 is our proposed method.

As can be seen from the Table 2, compared with the baseline method, CoordConv introduced in the 1×1 convolution layer of FPN and the first convolution layer of detection head can improve the performance of the model but will increase the amount of model calculation and reduce the number of frames per second of the model. Introducing CoordConv into the model backbone will increase the calculation amount of the model a little but greatly reduce the number of frames per second of the model. Light-YOLOv5 removes the largest feature maps from the baseline model. It is the model with the smallest size and the fastest detection among them. However, this method will also greatly reduce the accuracy and cannot meet the requirements of the project. Our proposed method, Coordlight-YOLOv5, best balances efficiency and accuracy. Precision, Recall, mAP(0.5), and mAP(0.5:0.95) are slightly reduced, respectively 1.7%, 2.4%, 2.2%, 3.7%, but FLOPs, size, parameters have massive reductions, 78.4%, 89.6%, 91.5%, and a frame rate increase of 23, which is very helpful for small embedded hardware.

Generality Study

We used five open-source datasets to evaluate our improved model and the sample images as shown in Fig-

Table 1: Comparison of different sizes of YOLOv5 Models: Evaluation of model complexity, computational cost, and performance metrics on kati dataset.

Model	**Parameters**	**FLOPs(G)**	**mAP(0.5)**	**mAP(0.5:0.95)**	**FPS**	**Weight(M)**
YOLOv5n	1,763,224	4.1	0.755	0.454	101	3.9
YOLOv5s	7,018,216	15.8	0.787	0.491	94	15.8
YOLOv5m	20,861,016	47.9	0.803	0.506	71	42.3
YOLOv5l	46,119,048	107.7	0.804	0.512	48	92.8

Table 2: Ablation study on YOLOv5 Variants: Comparison of model complexities, computational costs, and performance metrics on kati dataset.

Model	**Layer**	**FLOPs (G)**	**Weight (M)**	**Parameters**	**P**	**R**	**mAP(0.5)**	**mAP(0.5:0.95)**	**FPS**
YOLOv5s	157	15.8	14.5	7,018,216	0.928	0.71	0.787	0.491	94
Coord-YOLOv5	173	16.4	15.2	7,365,736	0.926	0.71	0.791	0.494	84
CoordBB-YOLOv5	165	15.9	14.5	7,035,496	0.925	0.71	0.787	0.491	19
Light-YOLOv5	120	3.1	1.4	409,644	0.898	0.666	0.738	0.426	121
CoordLight-YOLOv5	124	3.4	1.5	591,680	0.911	0.686	0.765	0.454	117

ure 3. The dataset contains cones of different types, colours, sizes, and sizes on the track to demonstrate the generality of our improved model. Since the datasets comes from cones on real tracks, it is more in line with the intended application scenarios. The Kati 9K Dataset [kati, 2022] is constructed from 9600 images. The formulastudent [Alpendre, 2023] dataset consists of 8933 images. The Cone Detection Dataset [UMotorsport, 2023] consists of 8507 images. The Cone Dataset [sma, 2022] consists of 5838 images. The Fsoco Dataset [Ma, 2022] consists of 3928 images. All datasets are divided into training, testing, and validation sets according to 8:1:1 ratio. From the size distribution of the target in the dataset as shown in Figure 7, the cone is mainly a small-sized target.

In order to verify the generality of our proposed method, we conducted comparison experiments between the baseline model and our proposed method on five open-source datasets. As can be seen from the Table 3, it has achieved excellent results in all datasets. Compared with the baseline method, although the mAP is slightly reduced, the calculation amount and weight are greatly reduced, with the FPS is greatly improved. And the sample results can be seen in Figure 3.

Comparison to the State-of-the-Art

In order to further validate the effectiveness of our proposed method, we compared the model with six current popular models for autonomous driving.

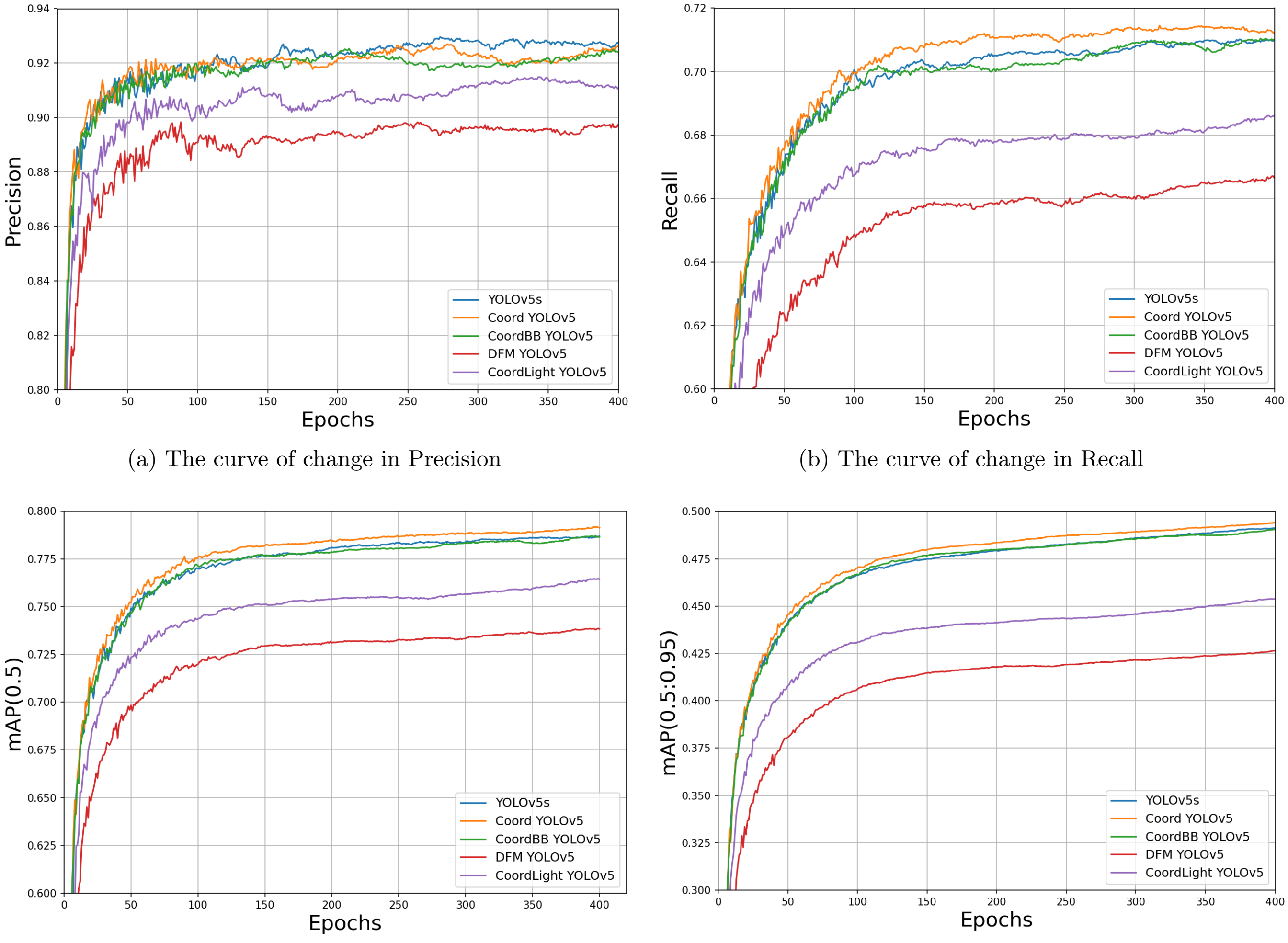
Hierarchical-Split Block [Yuan *et al.*, 2020] enhances ResNet by providing a broader receptive field, boosting model performance. HSCSP [Li *et al.*, 2023] integrates Hierarchical-Split Block with Cross Stage Partial (CSP) to further improve the receptive field and performance. GhostNet [Han *et al.*, 2020] incorporates the Ghost module to produce "Ghost" feature maps, reducing computational requirements. This network prioritises a balance between efficiency and accuracy, suitable for resource-limited devices. Convolutional Block Attention Module (CBAM) [Woo *et al.*, 2018] is an attention mechanism that augments deep convolutional neural networks' feature representation through channel and spatial attentions. It refines feature maps, enabling enhanced context capture. MobileNet [Howard *et al.*, 2017] is designed for mobile and edge devices, leveraging depthwise separable convolution to minimise computations and model size, suitable for various vision tasks with constrained resources. ShuffleNet [Zhang *et al.*, 2018] optimises performance on mobile devices through grouped convolution and channel shuffle, ensuring efficient feature combination and interaction while maintaining accuracy.

As shown in Table 4 and Figure 8, compared with Mobilenet and Ghostnet, the amount of calculation, weight, mAP and FPS all have better performance. Compared with Shufflenet, although the amount of calculation and weight are slightly higher than it, both mAP and FPS have better performance. Although Hierarchical-Split Block and CBAM are comparable to the baseline model in mAP, other indicators are not satisfactory.

4.3 Discussion

Our YOLOv5 comparison demonstrated the selection of YOLOv5s as the baseline model due to its balance of lightweight characteristics and accuracy, making it ideal for real-time auto-driving detection.

In our ablation studies, we integrated CoordConv into the YOLOv5 network's neck and head. Preliminary data indicates that while this increases model complexity slightly, it improves accuracy. Our proposed model,



(a) The curve of change in Precision

(b) The curve of change in Recall

(c) The curve of change in mAP(0.5)

(d) The curve of change in mAP(0.5:0.95)

Figure 6: Comparative analysis of precision, recall, and mAP scores in ablation study.

which removes the largest feature map and incorporates CoordConv, significantly reduces model complexity (78.4% decrease in FLOPs, 89.6% in size, and 91.5% in parameters) while preserving high accuracy.

Generality tests revealed the versatility of our method across five open-source datasets, emphasising its efficacy in real-time detection for self-driving racing cars.

Finally, compared to mainstream methods like HS-YOLOv5s, Ghost-YOLOv5, CBAM-YOLOv5, Mobilenet-YOLOv5, and Shufflenet-YOLOv5, our Coordlight-YOLOv5 consistently exhibited the best performance in terms of balancing between size, complexity, and accuracy

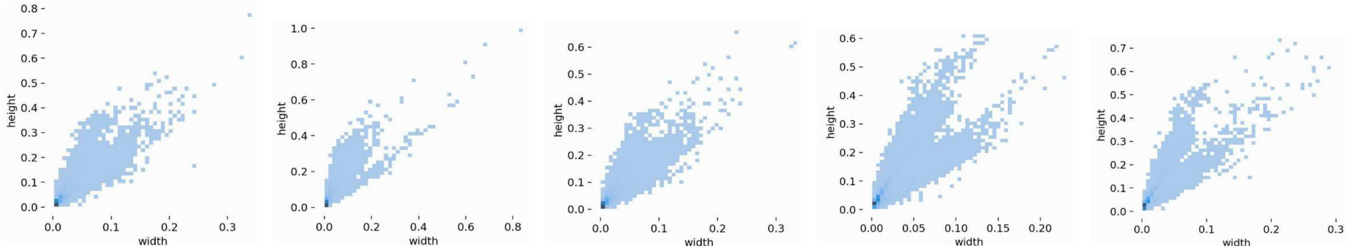
5 Conclusion

We proposed an improvement to YOLOv5 for the context of autonomous motorsport cone detection. The proposed method improves efficiency for performance-constrained embedded hardware, while maintaining high detection accuracy. We introduce CoordConv into the YOLOv5 network structure, in order to solve the translation invariance of traditional CNN by adding spatial awareness. For the motorsport context, the target cone objects are typically a small volume target, so we cut out the largest feature map to reduce the complexity of the model. Compared with related methods such as Mobilenet-YOLOv5 and Shufflenet-YOLOv5, our model has improved accuracy, model complexity and detection speed.

Future directions include testing the model with the embedded hardware onboard our autonomous racing cars, and to further improve the accuracy of the model while maintaining efficiency and frame rate.

References

[Adarsh *et al.*, 2020] Pranav Adarsh, Pratibha Rathi, and Manoj Kumar. YOLOv3-Tiny: Object detection and recognition using one stage improved model.



(a) Kati 9K Dataset [kati, 2022].

(b) formulastudent [Alpendre, 2023].

(c) Cone Detection Dataset [UMotorsport, 2023].

(d) Cone Dataset [sma, 2022].

(e) Fsoco Dataset [Ma, 2022].

Figure 7: Cones object size distribution in five different open source datasets used for generality experiment.

Table 3: Generality experiment: Comparative performance of YOLOv5s and Coordlight-YOLOv5 on five different open source datasets.

Dataset	**Model**	**Parameters**	**FLOPs (G)**	**mAP (0.5)**	**mAP (0.5:0.95)**	**FPS**	**Weight (M)**
Kati	YOLOv5s	7,018,216	15.8	0.787	0.491	94	14.5
Kati	Coordlight-YOLOv5	591,680	3.4	0.765	0.454	117	1.5
Fomulastudent	YOLOv5s	7,020,913	15.8	0.823	0.490	103	14.4
Fomulastudent	Coordlight-YOLOv5	592,262	3.4	0.803	0.470	117	1.5
UMotorsport	YOLOv5s	7,023,610	15.8	0.74	0.492	102	14.5
UMotorsport	Coordlight-YOLOv5	592,844	3.4	0.717	0.451	116	1.5
sma	YOLOv5s	7,023,610	15.8	0.705	0.423	99	14.4
sma	Coordlight-YOLOv5	592,844	3.4	0.676	0.390	115	1.5
Fsoco	YOLOv5s	7,020,913	15.8	0.848	0.595	103	14.4
Fsoco	Coordlight-YOLOv5	592,262	3.4	0.822	0.545	115	1.4

In *Proceedings of the International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 687–694, 2020.

[Alpendre, 2023] Diogo Alpendre. formulastudent Dataset. `https://universe.roboflow.com/diogo-alpendre-jjhgx/formulastudent`, 2023. Visited on 2023-08-10.

[Bharati and Pramanik, 2020] Puja Bharati and Ankita Pramanik. Deep learning techniques—R-CNN to Mask R-CNN: A survey. In Asit Kumar Das, Janmenjoy Nayak, Bighnaraj Naik, Soumen Kumar Pati, and Danilo Pelusi, editors, *Computational Intelligence in Pattern Recognition*, pages 657–668, 2020.

[Bochkovskiy *et al.*, 2020] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[Choi *et al.*, 2019] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[Ess *et al.*, 2010] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 29(14):1707–1725, 2010.

[Han *et al.*, 2020] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. GhostNet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[Howard *et al.*, 2017] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[Jocher *et al.*, 2022] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, Yonghye Kwon, Jiacong Fang, Kalen Michael, Diego Montes, Jebastin

Table 4: Comparative performance of YOLOv5s with various backbone integrations and feature enhancements

Model	**Parameters**	**FLOPs (G)**	**mAP (0.5)**	**mAP (0.5:0.95)**	**FPS**	**Weight (M)**
YOLOv5s	7,018,216	15.8	0.787	0.491	94	14.5
YOLOv5s-Mobilenetv3_small	3,543,926	6.3	0.733	0.445	76	7.5
YOLOv5s-Mobilenetv3_large	5,204,806	10.3	0.764	0.473	71	10.9
YOLOv5s-Shufflenetv2	439,048	1.3	0.677	0.389	78	1.3
YOLOv5s-HSB	5,441,152	14.1	0.781	0.488	71	11.4
YOLOv5s-Ghost	3,681,120	8.0	0.763	0.473	81	7.9
YOLOv5s-CBAM_backbone	7,225,358	16.1	0.781	0.467	86	14.9
YOLOv5s-CBAM_backbone_C3	7,330,582	17.1	0.784	0.492	74	15.1
YOLOv5s-CBAM_neck	7,061,518	15.8	0.786	0.488	87	14.6
Coordlight-YOLOv5	591,680	3.4	0.765	0.454	117	1.5

Nadar, Piotr Skalski, et al. ultralytics/yolov5: v6.1-TensorRT, TensorFlow edge TPU and OpenVINO export and inference. *Zenodo*, 2022.

[Jocher, 2020] Glenn Jocher. YOLOv5 by Ultralytics. https://github.com/ultralytics/yolov5, 2020.

[Katare *et al.*, 2023] Dewant Katare, Diego Perino, Jari Nurmi, Martijn Warnier, Marijn Janssen, and Aaron Yi Ding. A survey on approximate edge AI for energy efficient autonomous driving services. *IEEE Communications Surveys & Tutorials*, 2023.

[kati, 2022] kati. kati 9K Dataset. `https://universe.roboflow.com/kati/kati-9k-q1hox`, 2022. Visited on 2023-08-10.

[Li *et al.*, 2023] Guofa Li, Yingjie Zhang, Delin Ouyang, and Xingda Qu. An improved lightweight network based on YOLOv5s for object detection in autonomous driving. In *Computer Vision – ECCV 2022 Workshops*, pages 585–601, 2023.

[Liu *et al.*, 2018] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the CoordConv solution. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[Liu *et al.*, 2022] Haiying Liu, Fengqian Sun, Jason Gu, and Lixia Deng. Sf-YOLOv5: A lightweight small object detection algorithm based on improved feature fusion mode. *Sensors*, 22(15):5817, 2022.

[Long *et al.*, 2020] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, and Shilei Wen. PP-YOLO: An effective and efficient implementation of object detector. *arXiv preprint arXiv:2007.12099*, 2020.

[Ma, 2022] Qi Ma. Fsoco Dataset. `https://universe.roboflow.com/qi-ma/fsoco-gv7ev`, 2022. Visited on 2023-08-10.

[Redmon and Farhadi, 2018] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[sma, 2022] sma. Cones Dataset. `https://universe.roboflow.com/sma/cones-hp5il`, 2022. Visited on 2023-08-10.

[UMotorsport, 2023] UMotorsport. Cone Detection Dataset. `https://universe.roboflow.com/umotorsport/cone-detection-ssuto`, 2023. Visited on 2023-08-10.

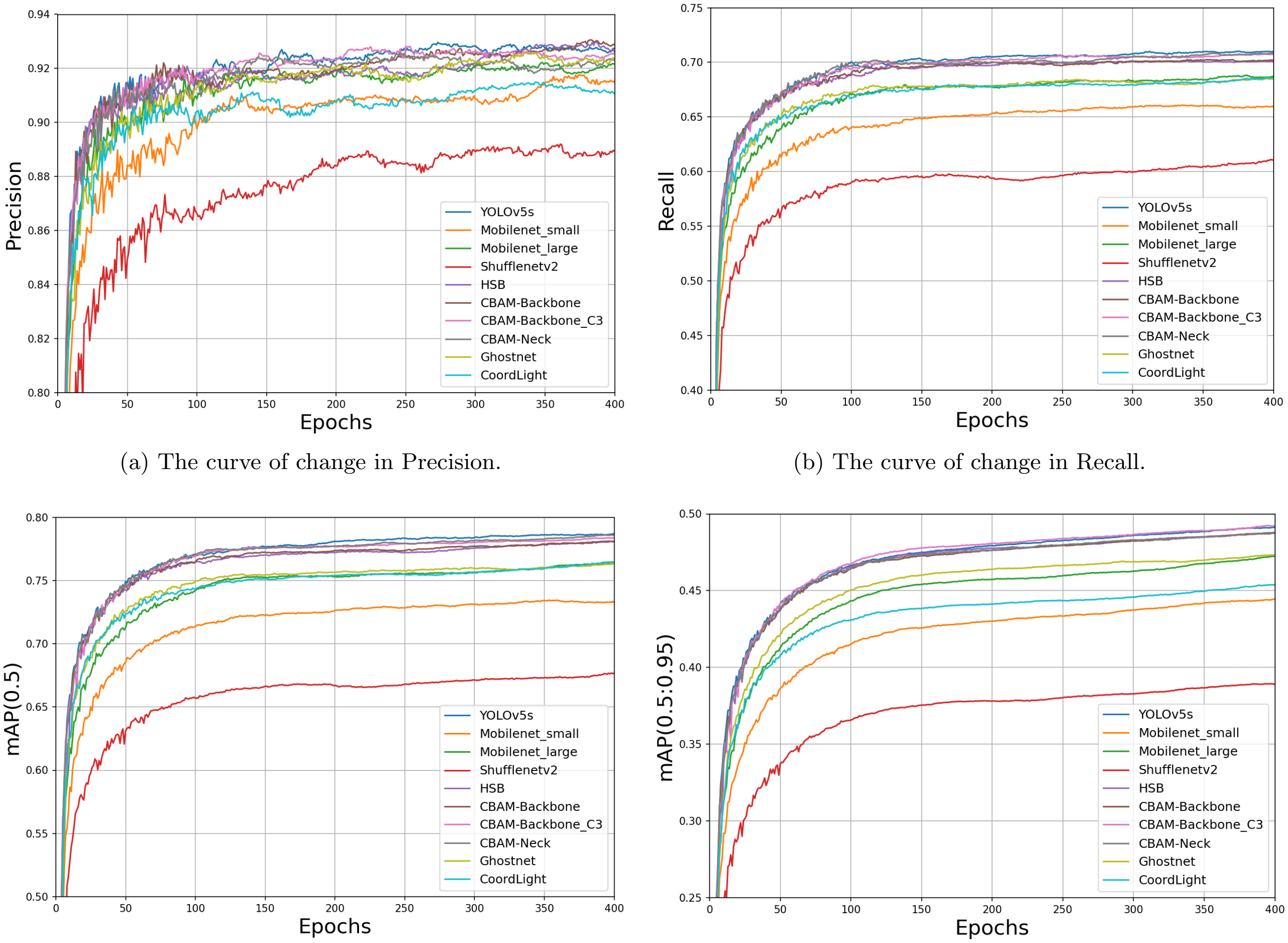
[UTSME, 2022] UTSME. UTSME22: Ramona, our seventh electric vehicle, built from the ground up to participate in the formula sae australasia competition. `https://www.utsmotorsports.com/utsme22-ramona/`, 2022. Visited on 2023-08-10.

[Wang *et al.*, 2019] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019.

[Wang *et al.*, 2021] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13029–13038, 2021.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[Wu *et al.*, 2020] Dihua Wu, Shuaichao Lv, Mei Jiang, and Huaibo Song. Using channel pruning-based YOLOv4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Computers and Electronics in Agriculture*, 178:105742, 2020.



(a) The curve of change in Precision.

(b) The curve of change in Recall.

(c) The curve of change in mAP(0.5).

(d) The curve of change in mAP(0.5:0.95).

Figure 8: Comparative analysis of precision, recall, and mAP scores across related methods.

[Wu *et al.*, 2021] Tian-Hao Wu, Tong-Wen Wang, and Ya-Qi Liu. Real-time vehicle and distance detection based on improved YOLOv5 network. In *Proceedings of the World Symposium on Artificial Intelligence (WSAI)*, pages 24–28, 2021.

[Yuan *et al.*, 2020] Pengcheng Yuan, Shufei Lin, Cheng Cui, Yuning Du, Ruoyu Guo, Dongliang He, Errui Ding, and Shumin Han. HS-ResNet: Hierarchical-split block on convolutional neural network. *arXiv preprint arXiv:2010.07621*, 2020.

[Zhang *et al.*, 2018] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Zhao and Li, 2020] Liquan Zhao and Shuaiyang Li. Object detection algorithm based on improved YOLOv3. *Electronics*, 9(3), 2020.

[Zhu *et al.*, 2021] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. 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 (ICCV) Workshops*, pages 2778–2788, 2021.