ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

Low-power neural network based on YOLOv5s

Xiaokun Qi, Tian He*

College of Mechanical and Electrical Engineering, Qingdao University, Qingdao, 266071, China
qixiaokun1210@foxmail.com
**Corresponding author: he_t@x263.net*

Abstract: *With the rapid development of convolutional neural networks, object detection technology has been revitalized, finding broader applications in robots, cars, cameras, and more. Many object detection algorithms can efficiently annotate anchor boxes for objects within the field of view, determining the specific location of the detection target. However, as the performance of detection algorithms improves, the issue of energy consumption arises, as high performance requires the sacrifice of high energy. Yet, most mobile devices have limited resources and cannot bear such a high load. The goal of this paper is to construct a low-energy consumption object detection algorithm, providing insights for the continuous efficient operation of mobile devices. YOLOv5s is selected as the base model, and its structure is improved. In neural networks, convolution operations are the main part of energy consumption. To reduce convolution operations, the backbone network is changed to the ShuffleNetv2 module, and DWConv is used to replace the C3 structure in the detection head. To improve model accuracy, a channel attention mechanism is introduced, and the loss function is modified. Ultimately, the new model significantly reduces the model's parameter quantity and size, thereby reducing its energy consumption, while maintaining essentially unchanged accuracy.*

Keywords: *Object Detection, Neural Networks, YOLOv5s, Energy Consumption*

1. Introduction

Object detection is an indispensable task in visual systems, capable of identifying specific targets in images or videos, and accurately marking their location and bounding boxes. The development of object detection technology benefits from the advancement of deep learning and neural networks. By training large-scale datasets and utilizing algorithms such as convolutional neural networks, object detection models can learn the features and contextual relationships of different targets in images, thereby achieving high accuracy and robust detection results. In outdoor environments, the types of targets may be diverse, and the quantity is large. Neural network-based object detection algorithms can quickly and accurately identify targets under environmental disturbances such as complex backgrounds and changes in illumination, and extract key information such as the target's location, bounding box, and category to assist the motion system in coordinate determination, thereby planning the most effective action path.

However, with the increasing performance requirements of visual systems, object detection algorithms need to process a large amount of input data and carry out complex calculations and inference processes. The neural network models involved are becoming more and more complex, leading to increasingly prominent energy consumption problems. In mobile devices, energy consumption has always been a significant challenge. Mobile devices usually operate under the support of a battery with limited power, and reasonable power allocation can make it stand by for a long time. By optimizing the structure of the neural network, the calculation and storage requirements can be reduced, thereby reducing energy consumption. For example, lightweight network structures can be used, or network pruning techniques can be adopted to reduce network parameters and computations. An important issue is how to reduce energy consumption while maintaining algorithm accuracy. Usually, there is a certain trade-off between accuracy and energy consumption, that is, excessive pursuit of low energy consumption may lead to a decrease in accuracy. Therefore, we need to find a balance between accuracy and energy consumption to meet the needs of practical applications. This paper constructs a low-energy lightweight neural network. Based on YOLOv5s, improvements are made to reduce its convolution operations, the backbone network is changed to ShuffleNetv2, and DWConv is used to replace the C3 structure in the detection head, channel attention mechanism is introduced, the loss function is changed, and under the premise of basically unchanged accuracy, the model's parameter quantity and size are greatly reduced, thereby reducing its energy consumption.

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

The rest of this paper is arranged as follows. The second section reviews related research, the third section proposes a low-energy object detection model, and the fourth section concludes.

2. Related Research

Currently, there are various lightweight object detection methods, among which YOLO is widely used. Although YOLO has a relatively fast detection speed, occupies less computational resources during inference, and consumes relatively low energy, it still needs to be lightweighted for long-term tasks on resource-limited mobile devices. This can be achieved by improving the algorithm architecture to reduce its energy consumption. Hu[1] proposed a new object detection model based on lightweight convolutional neural networks, Micro-YOLO, which significantly reduces the number of model parameters and computational costs while maintaining detection speed and accuracy. The author improved YOLOv3-tiny by replacing the original convolutional layer with Depthwise Separable Convolution (DSConv) and Mobile Inverted Bottleneck Convolution (MBConv), and designed a progressive channel-level pruning algorithm to minimize the number of parameters and maximize detection performance.

Zhang[2] designed a new lightweight object detection system, CSL-YOLO, to address edge computing tasks. The author proposed a lightweight convolution module called Cross-Stage Lightweight (CSL), and the backbone of the network is composed of multiple convolution modules. The Feature Pyramid Network (FPN) was improved by replacing the 3x3 convolution in FPN with the CSL-Module. In the intermediate unfolding stage, depth convolution is used instead of pointwise convolution to generate candidate features. With only 43% of the FLOPs and 52% of the parameters of TinyYOLOv4, it achieved superior detection performance.

Luo[3] improved YOLOv4 using the original network structure as the basic skeleton, and used the lightweight neural network MobileNet as the main network for feature extraction. He proposed an Adaptive Spatial Feature Fusion (ASFF) method to solve the poor effect of PANet in multi-scale feature fusion. The author also redefined the position loss function to improve the model's accuracy and compensate for the lack of accuracy after the reduction of parameters. Compared with the original algorithm and other mainstream detection algorithms, the improved model is more suitable for deployment on mobile and embedded devices for real-time detection. Similar to Luo's method, Chen et al. (2021, UAV Lightweight Object Detection Based on the Improved YOLO Algorithm) proposed an improvement to the YOLOv5 network structure based on MobileNetv3. It introduced the lightweight neural network MobileNetv3 as the backbone network structure of the model. Through experiments, it was found that the memory usage of the optimized YOLOv5 network model can be reduced by 72.4% compared to the original model. Similar to the method of Luo and others, Chen[4] proposed an improvement to the YOLOv5 network structure based on MobileNetv3. It introduced the lightweight neural network MobileNetv3 as the backbone network structure of the model. Through experiments, it was found that the memory usage of the optimized YOLOv5 network model can be reduced by 72.4% compared to the original model.

Gao[5] also proposed YOLO-TLA based on YOLOv5, which effectively improved its performance in small object detection. YOLO-TLA introduces additional small object detection layers to generate larger scale feature maps for better recognition of small objects. In addition, it uses the C3CrossCovn module to reduce the amount of parameters and computational requirements without losing accuracy, thereby reducing the resources consumed by the device when performing tasks.

Gong[6] made improvements to YOLOv7, using the network structure of YOLOv7 as a foundation. To reduce computational resources, ShuffleNetv2 was used to replace the original backbone structure. Due to the reduction in the number of parameters, its accuracy would inevitably decrease, so the Vision Transformer self-attention mechanism was introduced to achieve efficient feature representation. After integrating the three, the efficiency of the resulting model was improved, and the computational resources were significantly reduced.

The Google team[7] proposed a new method for object detection model search on mobile devices, called MobileDets. This method combines Neural Architecture Search (NAS) and regular convolution, seeking a balance between real-time performance and accuracy. The authors proposed an expanded search space: a fully convolutional sequence that includes Instance Batch Normalization (IBN) and Tensor Decomposition, referred to as Tensor Decomposition Based Search Space (TBD). On the COCO object detection task, the inference time of MobileDets is not much different from that of MobileNetV3+SSDLite, but its overall performance is superior to MobileNetV3+SSDLite.

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

Baidu[8] open-sourced a lightweight real-time object detection model, PP-PicoDet, in 2021, which outperforms YOLOv5. This model uses ShuffleNetV2 as the backbone network and enhances it by proposing a new backbone network, ESNet (Enhanced ShuffleNet). In terms of the detection head, the authors use depthwise separable convolution with a kernel size of 5*5 to enhance the receptive field. Both its Neck and Head have four scale branches, maintaining consistent channel numbers. Compared to YOLOX-Nano, PP-PicoDet only requires 0.99M parameters to achieve a mAP of 30.6%, while reducing inference latency by 55%. As a domestic giant, Tencent[9] is not to be outdone, proposing a lightweight model method based on self-supervised distillation learning called Distilled Contrastive Learning (DisCo). This method abandons the shared queue, making the entire framework independent of MoCo-V2, allowing the model to combine with other self-supervised/unsupervised learning methods that are more effective than MoCo-V2. During the self-supervised learning and distillation stages, the dimension of the MLP hidden layer is increased to further enhance the effect after the lightweight model is distilled.

Alibaba[10] breaks the norm by proposing a new concept for improving models, namely designing a backbone with a small computational load and a neck with a large computational load, enabling the network to interact with spatial information in high-resolution feature maps and semantic information in low-resolution. The authors believe that compared to the backbone structure, the Feature Pyramid Network (FPN) contributes more to the model's detection capabilities. As a result, they propose a new type of FPN structure, the Generalized-FPN (GFPN). The core idea is to introduce a global feature pyramid and extract richer semantic information through multi-scale feature fusion. This structure draws on DenseNet and designs a dense link to increase feature reuse. To maintain the effectiveness of the computational load, GFPN uses a log2n-link, allowing for a deeper network structure. To overcome large-scale changes, GFPN introduces the Queen-Fusion structure to increase feature fusion.

3. Low-Energy-Consumption Object Detection Algorithms

3.1 Main Convolution Module Improvement

As is well known, the number of parameters and FLOPs are key factors affecting the energy consumption of a model. Therefore, this paper adopts a model lightweighting method based on YOLOv5 to reduce the number of model parameters and energy consumption for energy optimization. In this experiment, the backbone network in the original structure is replaced with ShuffleNet_v2. The Shuffle module does not introduce too many branches, there are only two paths, and it can ensure that the number of input and output channels of the feature map are equal. When performing feature fusion, the Concat operation is used instead of Add, thereby reducing the consumption of memory access. The 3x3 depthwise separable convolution block in the ShuffleNet_v2 model greatly reduces the number of parameters compared to ordinary convolutions, with less accuracy loss. At the end of the original backbone network, there is an SPPF module. This experiment removes this module because it requires parallel operations, which affects the detection speed.

3.2 Introduction of Attention Mechanism

After modifying the backbone network, the model's accuracy may decrease due to the reduction in the number of parameters. Therefore, it is considered to introduce an attention mechanism to improve its accuracy.

In ordinary convolution operations, each channel is considered equally important, and no particular channel is treated specially. However, in reality, when we extract features, we do not need to pay attention to all the information in the feature map. Therefore, Monenta proposed SENet, which allows the network to autonomously obtain the weights of each channel through learning, to enhance the required channel features and reduce the unnecessary channel features. As can be seen from the figure 1, the SE module is composed of two parts, namely Squeeze and Excitation. The Squeeze operation compresses features in the spatial dimension through global average pooling, making the features on each channel one-dimensional, enhancing the correlation between channels, and shielding the interference of spatial distribution correlation. To some extent, this can obtain a global receptive field. In the Excitation operation, there are two 1x1 convolution kernels. The first convolution performs dimension reduction, and the second performs dimension increase, keeping the input and output consistent. Then the sigmoid function is connected to restrict the weight between 0 and 1. Finally, the obtained weight is combined with the original feature quantity to complete the SE operation.

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

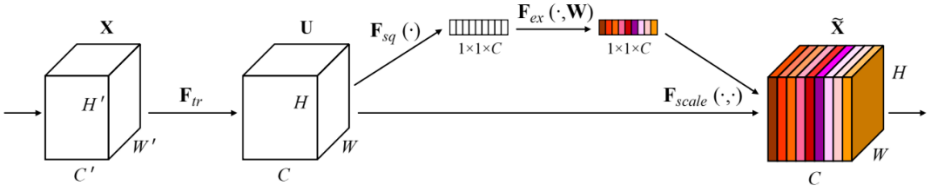

Figure 1: Channel Attention Mechanism

CBAM (Convolutional Block Attention Module) is an attention mechanism that combines spatial and channel dimensions, which is shown in the figure 2. Compared to SENet, which only focuses on channel information, CBAM can obtain more image features. Channel attention mechanism usually focuses on what information is important. In order to calculate the channel attention mechanism, average pooling is commonly used to compress the spatial dimension of the input feature map. However, some people believe that max pooling can also obtain important information from the feature map. Spatial attention mechanism focuses on where the information is important, which is an improvement on channel attention. In order to calculate the spatial attention mechanism, it is necessary to perform average pooling and max pooling operations along the channel, and then connect them to generate more persuasive feature information.

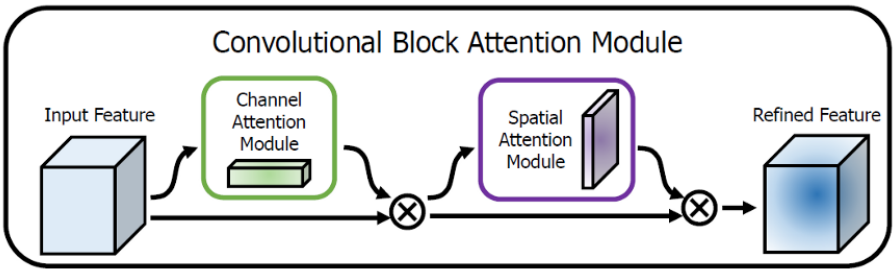

Figure 2: Spatial Attention Mechanism

3.3 Improved Network Structure

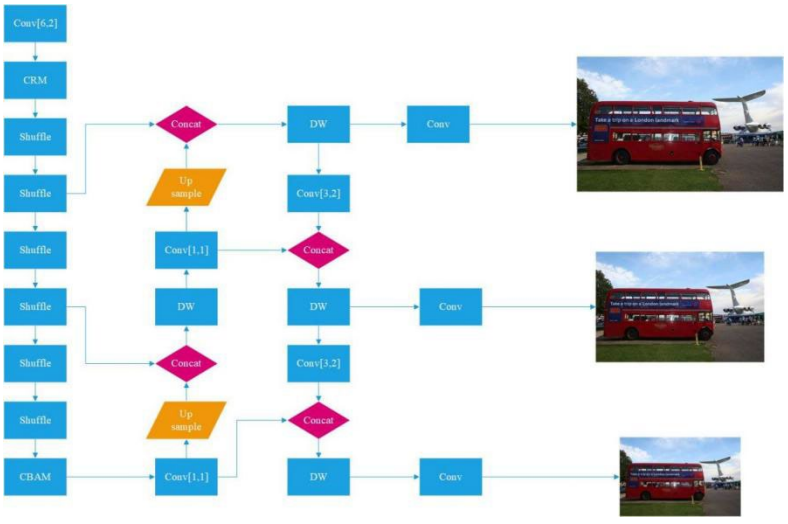

Figure 3: Improved Network Structure

This paper improves the model based on YOLOv5s. In order to reduce the parameter size and FLOPs of the backbone network, the backbone network is changed to a more lightweight ShuffleNet_v2. This module can maintain an effective receptive field with fewer parameters. Of course, as the number of parameters decreases, the features that the backbone network can learn are reduced. To compensate for the loss in accuracy, this experiment introduces the CBAM spatial attention mechanism at the end of the

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

main network, thereby improving the accuracy of the improved model. To reduce parallel operations, the SPPF module at the end of the original model is removed. The C3 module in the connection layer also contains a large number of parameters, so it is replaced with the depth-separable convolution DW module, further reducing the model parameter size. The structure of the improved network is shown in the figure 3.

3.4 Loss Function

Intersection over Union (IoU) is a commonly used metric in object detection tasks, also known as the Jaccard index. It can reflect the degree of overlap between the true detection box and the predicted detection box.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

In YOLOv5, the authors use CIoU. The formula is as follows. The authors suggest that an excellent regression localization loss should consider the following factors: overlap area, center point distance, and aspect ratio. The calculation formula for v is based on the aspect ratio of the prediction box and the target box, and then takes their arctangent (arctan), which can make the change of the aspect ratio smoother. After taking the arctangent, calculate the square of the difference between these two angles to get a non-negative value. Finally, multiply by a scaling factor of 4/π^2, so that the maximum value of v is 1.

α is a balance factor used to adjust the balance between the distance term and the aspect ratio error term v. The formula is designed so that when IoU is close to 1, that is, when the prediction box and the target box highly overlap, α is close to v. Therefore, when the overlap of the prediction box and the target box is high, more emphasis is placed on the aspect ratio error. When the overlap of the prediction box and the target box is relatively low, and α is close to 1, more emphasis is placed on the distance term.

$$CIoU = \frac{\rho^2(b,b^{gt})}{c^2} + \alpha v \tag{2}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{3}$$

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

$$\alpha = \frac{v}{(1-IoU)+v} \tag{5}$$

Although CIoU has improved compared to the traditional IoU, it still cannot effectively locate in some complex scenarios. CIoU mainly focuses on the center point distance and aspect ratio between the prediction box and the real box, but it cannot provide good feedback for the angle information of the rotating target. Moreover, its gradient may be discontinuous or change drastically in some cases, which may affect the convergence of the model. In contrast, WIoU introduces rotation-related geometric features, which can better handle rotating targets. At the same time, its gradient is smoother, and the positioning accuracy has also improved to a certain extent.

WIoU has three versions: v1, v2, and v3. This article will use v3 for analysis.

WIoUv3 introduces a concept called 'outlier degree' to describe the quality of the prediction box. The outlier degree reflects the degree of difference between the prediction box and the real box. Specifically, the first step is to find the prediction box that has the maximum IoU with the real box, which we call the 'optimal prediction box'. Then, the Euclidean distance between the geometric center of the optimal prediction box and the real box is calculated as part of the outlier degree, and the logarithm of the area ratio between the optimal prediction box and the real box is calculated as another part of the outlier degree. Finally, the two parts are added together with certain weights to get the final value of the outlier degree. The smaller the outlier degree value, the closer the geometric center position and size of the prediction box are to the real box, indicating higher quality of the prediction box. Conversely, the larger the outlier degree value, the greater the difference between the prediction box and the real box, indicating lower quality.

$$\beta = \frac{Loss_{IoU}^{*}}{Loss_{IoU}} \in [0, +\infty] \tag{6}$$

$$Loss_{WIoU_{v3}} = \gamma\, Loss_{WIoU_{v1}} \tag{7}$$

$$\gamma = \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{8}$$

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

3.5 Experimental Results and Analysis

3.5.1 Data Preparation

This study focuses on objects in outdoor environments. Nearly 10,000 images were collected from various sources, including objects such as people, cars, buses, bicycles, airplanes, motorcycles, etc. The size of the images is 640×640.

To validate the detection performance of the improved target detection model, ablation experiments were conducted on a self-built dataset, keeping other factors constant. The backbone network was changed to ShuffleNet_v2, the SPPF module was removed, different attention modules such as channel attention SE and spatial attention CBAM were added, and the DW module was added based on the change of the backbone network to ShuffleNet_v2. The results of the ablation experiments are shown in the table 1.

Table 1: Ablation Experiment

Network Structure	SPPF	Attention Module	Number of Parameters	GFLOPs	Inference Time (ms)	mAP0.5	Map0.5:0.95
YOLOv5s	√	×	7035811	16.0	23.8	65.73%	42.32%
Shuffle	×	×	3344392	2.9	21.5	61.00%	32.25%
Shuffle	×	SE	3381416	2.9	22.0	62.10%	37.70%
Shuffle	×	CBAM	3381514	3.0	22.5	64.50%	39.60%
Shuffle_DW	×	CBAM	1126730	2.0	19.3	61.85%	32.28%

From the experiment, it is observed that when the backbone network is replaced from the original C3 module to the ShuffleNet_v2 module, as introduced earlier, both the parameter size and GFLOPs significantly decrease. The parameter size is about 47.5% of the original, and GFLOPs is about 18% of the original. This undoubtedly reduces the computational overhead of the device and theoretically would reduce the energy consumption during training. However, with the significant reduction in parameter size, the features extracted from the input image by the convolutional neural network will also decrease. Although this was considered during the module design, the problem of accuracy decline is still inevitable. To address the issue of accuracy decline, this experiment introduces attention mechanisms in subsequent experiments with ShuffleNet as the backbone network, hoping to improve the accuracy problem. When the channel attention mechanism SE module and the spatial attention mechanism CBAM are introduced into the neural network structure, it can be clearly observed that, compared to the original YOLOv5s algorithm, the accuracy is still insufficient. But compared to the neural network after the improved backbone without the attention mechanism, its accuracy has improved. Moreover, although new modules have been introduced, the model's parameter size and GFLOPs have not changed significantly, so the energy consumption should not change much. In addition to the backbone network, there are many C3 modules in the subsequent connection layer, so on the basis of replacing the C3 module in the backbone network, the connection part uses the depth separable convolution DW module for replacement. As can be seen from the table above, the model becomes smaller again on the basis of the previous replacement, the parameter size is about 16% of the original, and GFLOPs is about 12.5% of the original. Although there is a decline in accuracy, it will be improved through subsequent adjustments. In summary, when the original model YOLOv5 is improved, the model parameter size and GFLOPs are significantly reduced, and the accuracy has declined.

Next, we validate the improvement of the loss function. Based on the aforementioned Shuffle_DW experiment, the loss function CIoU of the original model is replaced with the new WIoU. The experimental results are shown in the Table 2.

Table 2: Improved Loss Function

Network Structure	Loss Function	Weight File Size(M)	mAP0.5	mAP0.5:0.95
YOLOv5s	CIoU	14.4	65.73%	42.32%
Shuffle_DW	CIoU	2.4	61.85%	32.28%
Shuffle_DW	WIoU	2.4	63.50%	39.50%

It is evident from the table that the accuracy of the improved model has increased based on the original, but the model size has not increased, indicating that the introduction of the loss function WIoU is successful. Although there is a slight gap in accuracy between the final model and the original model, the parameter volume and model size are significantly reduced. We cannot excel in all directions, as the saying goes, you can't have your cake and eat it too, a moderate gap is acceptable. The accuracy

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

comparison of all models is shown in the following figure. From the trend of model accuracy, we can find that blindly pursuing model lightweight will inevitably pay a certain accuracy price. The mAP0.5 of this experiment is about 3% lower than the original model, and mAP0.5:0.95 is about 2% lower. If the performance of the model is improved in the future, then this loss is worth it.

Of course, simply comparing the parameter volume and model size cannot explain the specific changes in energy consumption. We ultimately need to return to energy consumption analysis. Therefore, this experiment starts from reality, with the help of hardware detection software HWiNFO, to measure the time, power, and energy consumption of the model during training and inference, in order to verify whether the improved model meets our energy consumption requirements.

Firstly, we compare the original YOLOv5s model with various variant models in terms of training time, power, and energy consumption during training. We observe the differences between different models during training. Based on the original YOLOv5s, we perform normalization. The results are shown in the figure 4.

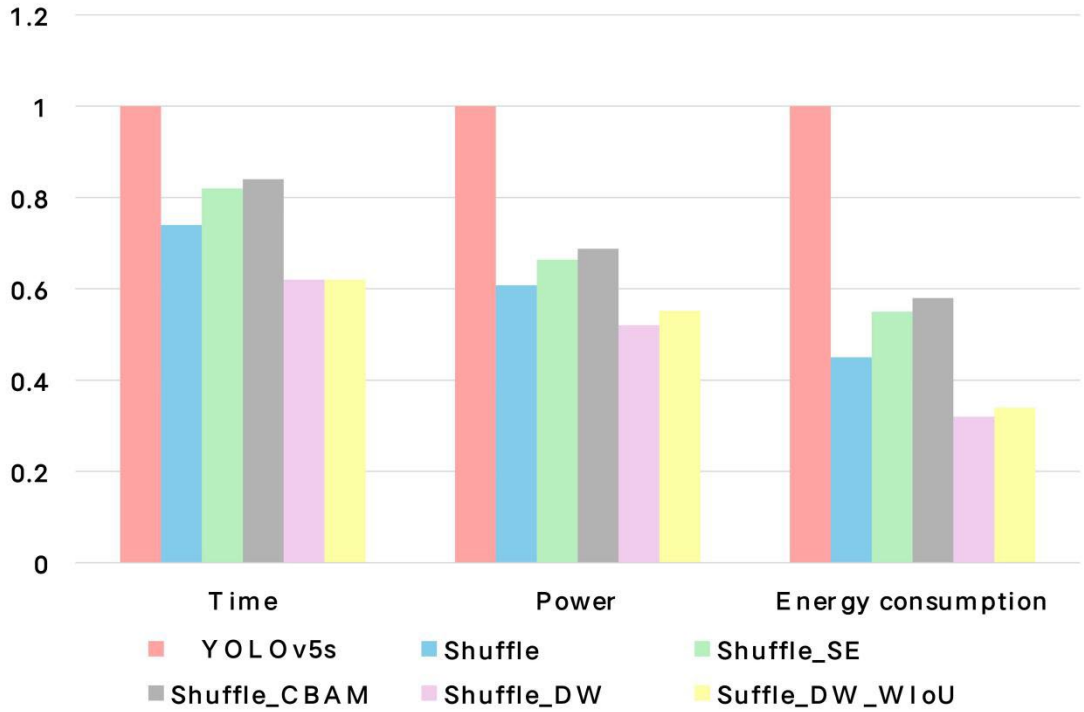

Figure 4: Comparison of Training Time, Power, and Energy Consumption for Different Models

After normalization, the differences between different models can be clearly observed. When the backbone network of the model is improved, the training time, power, and energy consumption of all models are significantly reduced. In the last two models, the loss function IoU was modified. After the modification, the power and energy consumption slightly increased compared to before the modification. However, we are pursuing a balance between accuracy and energy consumption, so this slight change is acceptable. Therefore, after improving the original model, the energy consumption of the current model is 34% of the original, which meets the expectations of the improvement.

Regarding energy consumption, it mainly occurs when the model performs visual tasks, so we also need to validate our model during the inference stage. Similar to the above process, we compare the time, power, and energy consumption required by different models to perform the same inference task. For ease of processing, we also normalize the results. The experimental results are shown in Figure 5.

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

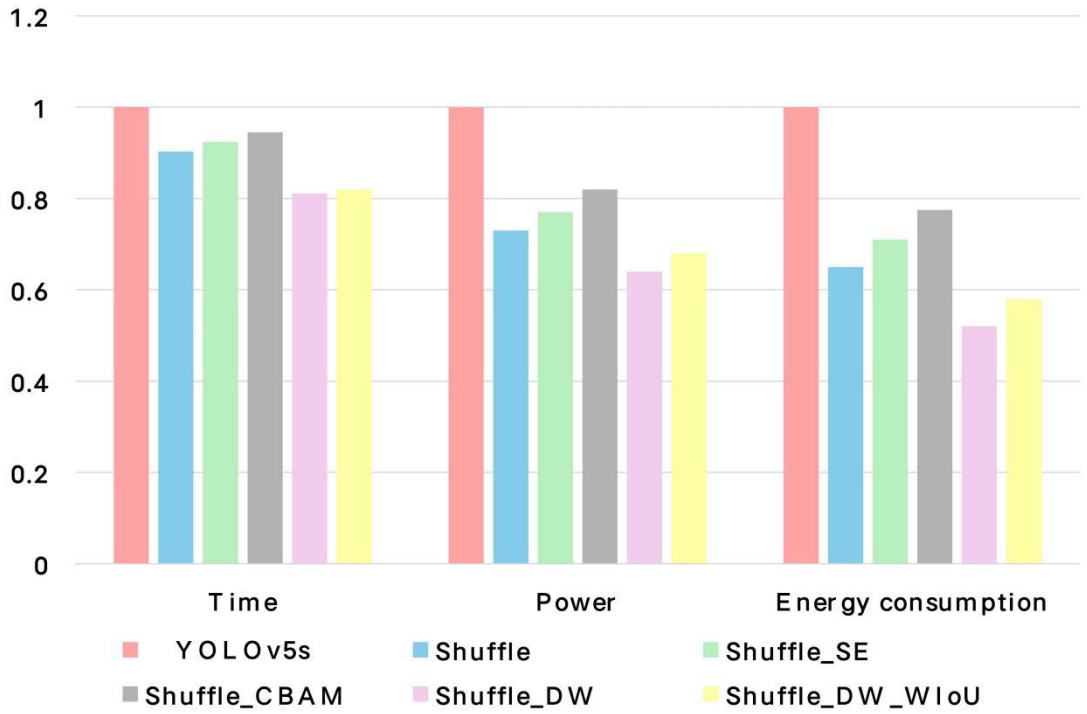

Figure 5: Comparison of inference time, power, and energy consumption across different models

From our experiments, it is evident that after improving the model, due to the reduction in the number of parameters, FLOPs, and overall model size, the inference time also decreases. When different models perform inference, due to the inclusion of various modules, there will be differences in resource utilization, hence the power consumption also varies. In terms of total energy consumption, after comparison, the energy consumption of the improved model when performing the same task is about 60% of the original model. This demonstrates that our model has achieved optimization in terms of energy consumption. While completing the same tasks and sacrificing a small amount of accuracy, the energy consumption of the model can be greatly reduced, achieving long-term endurance. The validation results of our experimental model are shown below. Compared with the original model at the beginning, there are differences in the detection results of a few targets, but overall, it meets our expectations.The training process data and results verification are shown in Figure 6 and 7.

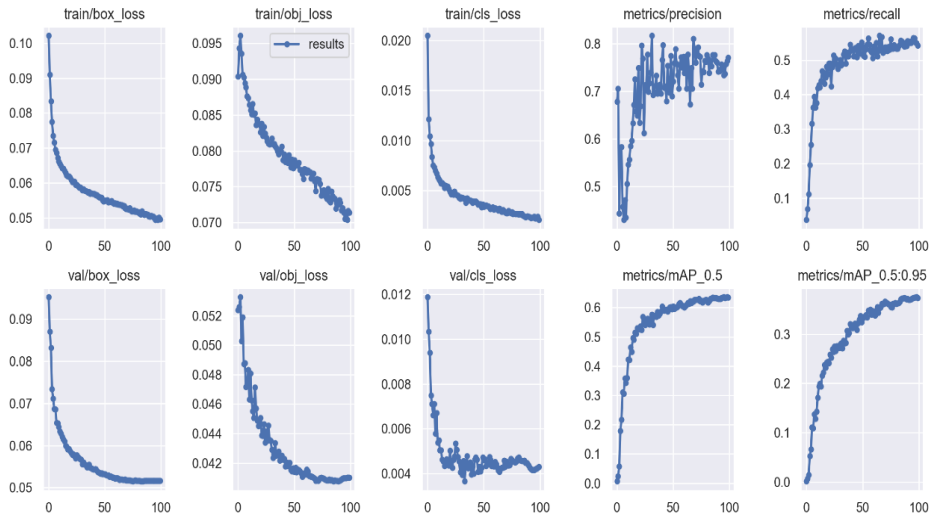

Figure 6: Training Process Data

ISSN 2706-655X Vol.6, Issue 2: 64-72, DOI: 10.25236/IJFET.2024.060209

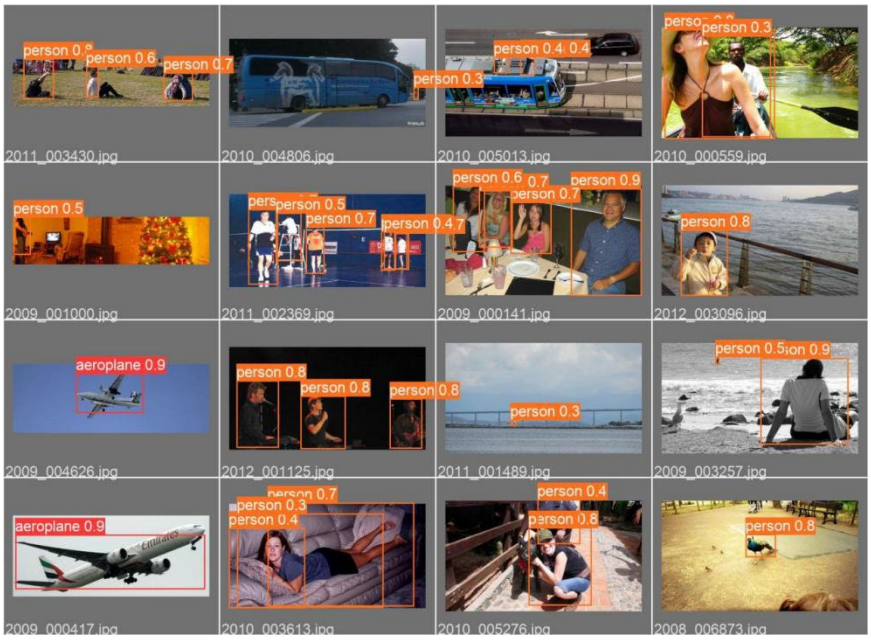

Figure 7: Results Verification

4. Conclusion

This experiment is based on YOLOv5, with modifications made to the model. The backbone network was replaced, an attention mechanism module was added, and the loss function was adjusted to construct our own object detection model. Finally, an ablation study was conducted to compare the parameters of different models and observe the effects of the improved model. The experiment shows that this model reduces the energy consumption to 60% of the original without losing a significant amount of accuracy.

References

[1] Hu L, Li Y. Micro-YOLO: Exploring Efficient Methods to Compress CNN based Object Detection Model; proceedings of the International Conference on Agents and Artificial Intelligence, F, 2021 [C].
[2] Zhang Y, Lee C-C, Hsieh J-W, et al. CSL-YOLO: A New Lightweight Object Detection System for Edge Computing [J]. ArXiv, 2021, abs/2107.04829.
[3] Yujie Luo, Jian Zhang, Liang Chen, et al. Lightweight target detection algorithm based on adaptive spatial feature fusion [J]. Laser & Optoelectronics Progress, 2022, 59(4): 0415004--11.
[4] Chen Y, Chen X, Chen L, et al. UAV Lightweight Object Detection Based on the Improved YOLO Algorithm [J]. Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering, 2021.
[5] Gao P, Ji C-L, Yu T, et al. YOLO-TLA: An Efficient and Lightweight Small Object Detection Model based on YOLOv5 [J]. ArXiv, 2024, abs/2402.14309.
[6] Gong W. Lightweight Object Detection: A Study Based on YOLOv7 Integrated with ShuffleNetv2 and Vision Transformer [J]. ArXiv, 2024, abs/2403.01736.
[7] Xiong Y, Liu H, Gupta S, et al. MobileDets: Searching for Object Detection Architectures for Mobile Accelerators [J]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 3824-33.
[8] Yu G, Chang Q, Lv W, et al. PP-PicoDet: A Better Real-Time Object Detector on Mobile Devices [J]. ArXiv, 2021, abs/2111.00902.
[9] Gao Y, Zhuang J-X, Li K, et al. DisCo: Remedy Self-supervised Learning on Lightweight Models with Distilled Contrastive Learning [J]. ArXiv, 2021, abs/2104.09124.
[10] Jiang Y, Tan Z, Wang J, et al. GiraffeDet: A Heavy-Neck Paradigm for Object Detection [J]. ArXiv, 2022, abs/2202.04256.