Automation and Machine Learning (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2024.050104
ISSN 2516-5003 Vol. 5 Num. 1

YOLOv8 with Multi Strategy Integrated Optimization and Application in Object Detection

Jiafeng Li[1], Chenxi Yan[2]

[1]*College of Software Engineering, Sichuan University, Chengdu, 610207, China*
[2]*School of Computer Science, Northeast Electric Power University, Jilin, Jilin, 132011, China*

Keywords: Yolov8s; ShuffleNet-v2; MHSA; lightweighting

Abstract: YOLOv8s model often used at Object detection which is widely used in many fields, cannot aggregate feature information well for a specific task, and the number of parameters is large and the accuracy is not high. Aiming at the above problems of traditional YOLOv8s, a new lightweight and performance-balanced YOLOv8s network structure is proposed. The C2F module in the backbone network are replaced with ShuffleNet-v2, and in order to further improve the accuracy degradation due to the decrease in the number of parameters, a global multi-attention mechanism is added to obtain global information, learn the correlation between features at different scales, and fuse them. SGD is used as an optimizer to further improve accuracy. The experimental results show that the introduction of ShuffleNet-v2 and MHSA effectively reduces the number of parameters of the model, significantly reduces the training time, and the accuracy is considerable, and compared with other optimizers SGD has the largest performance improvement, and has an excellent performance in terms of the balance between lightweighting and algorithmic performance.

1. Introduction

Object detection, as one of the core problems in the field of computer vision, is widely used in many fields such as face recognition, intelligent transportation, industrial inspection, automatic driving, etc.[1] With the rapid development of deep learning in recent years, object detection has become a hot topic in theory and application research[2-5]. However, existing methods generally have the problem of poor detection accuracy when the object scale distribution is inconsistent[6]. To solve this problem some scholars proposed a model based on deep learning, but this simultaneously increases the complexity of the model and reduces the training efficiency.

In 2017, Joseph Redmon et al. proposed the YOLOv2 model[7] , an article proposing a method to jointly train object detection and classification by simultaneously training the YOLOv2 model on the COCO detection dataset and the ImageNet classification dataset, allowing it to predict object classes without labeled detection data. However, compared to fast R-CNN, YOLOv2 performs poorly in terms of position error and has low recall, making it prone to training instability.In 2018, Joseph Redmon proposed the YOLOv3 model based on this[8] that uses a new network for feature extraction, allows cross-scale prediction, and is trained on the same dataset, resulting in a large improvement in precision. However, the model is not suitable for dealing with objects of larger sizes and has some difficulties in perfectly aligning the bounding boxes of objects. Subsequently, some scholars have

also captured multi-scale contextual information by applying dilation convolution at different scales, and utilized the dilated spatial pyramid module (DSPM) to[9] to realize multi-scale target detection.

With the continuous research on deep learning, Alexey Bochkovskiy et al. proposed a new YOLOv4 model in 2020[10], which used CSPDarknet53 backbone network and Mosaic data enhancement method to modify the parameters of the model through self-adversarial training to improve the stability and accuracy of the model, and was trained on MS COCO dataset, achieving 43.5% AP, but due to the complexity of the model and the large amount of computation, which resulted in a long training time, and when dealing with small-scale dataset is prone to overfitting problem. Afterwards, some teams also further improved YOLOv4 by replacing the CSPBlock module with the ResBlock-D module[11] that aims to increase the training speed, reduce the complexity, and apply to real-time detection, but again it is difficult to balance the contradiction between model size and detection accuracy.

Based on the improvement of the above scholars, this paper aims at the current traditional yolov8s model[12] with lower accuracy and complex model, an improved yolov8s algorithm is proposed. Specifically, the backbone network is replaced with multiple ShuffleNet-v2 serial connections, and a multi-head attention mechanism is introduced at the last layer, while the stochastic gradient descent method and YOLOv8s are combined to improve the accuracy and reduce the model complexity at the same time.

2. Models and methods

2.1 Data pre-processing techniques

YOLOv8 follows the Mosaic data enhancement algorithm proposed in YOLOv4, which not only enriches the background of the image, but also improves the batch size during training, which makes it possible to get good results even when training on smaller datasets, which is conducive to improving the performance of small target detection, and it can reduce the pressure on the GPU and accelerate the training efficiency.

The Mosaic data enhancement of YOLOv4 refers to the CutMix data enhancement approach, by randomly cropping four images and then splicing them, each of which has a corresponding bounding box, and then finally splicing the four images again onto the same image as training data, and at the same time obtaining the corresponding bounding box of this image, which is equivalent to passing in four images for learning at the same time, and greatly enriches the background of the detected objects.

2.2 YOLOv8s model

Since YOLOv8s has higher accuracy compared to v8n and faster inference speed compared to v8m, it can balance the speed and accuracy. Therefore, YOLOv8s is chosen for target detection in this paper.

The backbone network of YOLOv8s, which continues the idea of CSP, adopts the CSPDarkNet structure, divides the feature map into two parts, one for convolution operation and the other for jump connection, and replaces all the C3 modules with C2F modules, and still uses the SPPF modules; in the neck network, the convolutional structure of the sampling stage on the PAN-FPN is deleted; and the head part is replaced by the current mainstream decoupled head structure, which separates the classification and detection heads, and also switches from Anchor-Based to Anchor-Free. The network structure is shown in Figure 1.

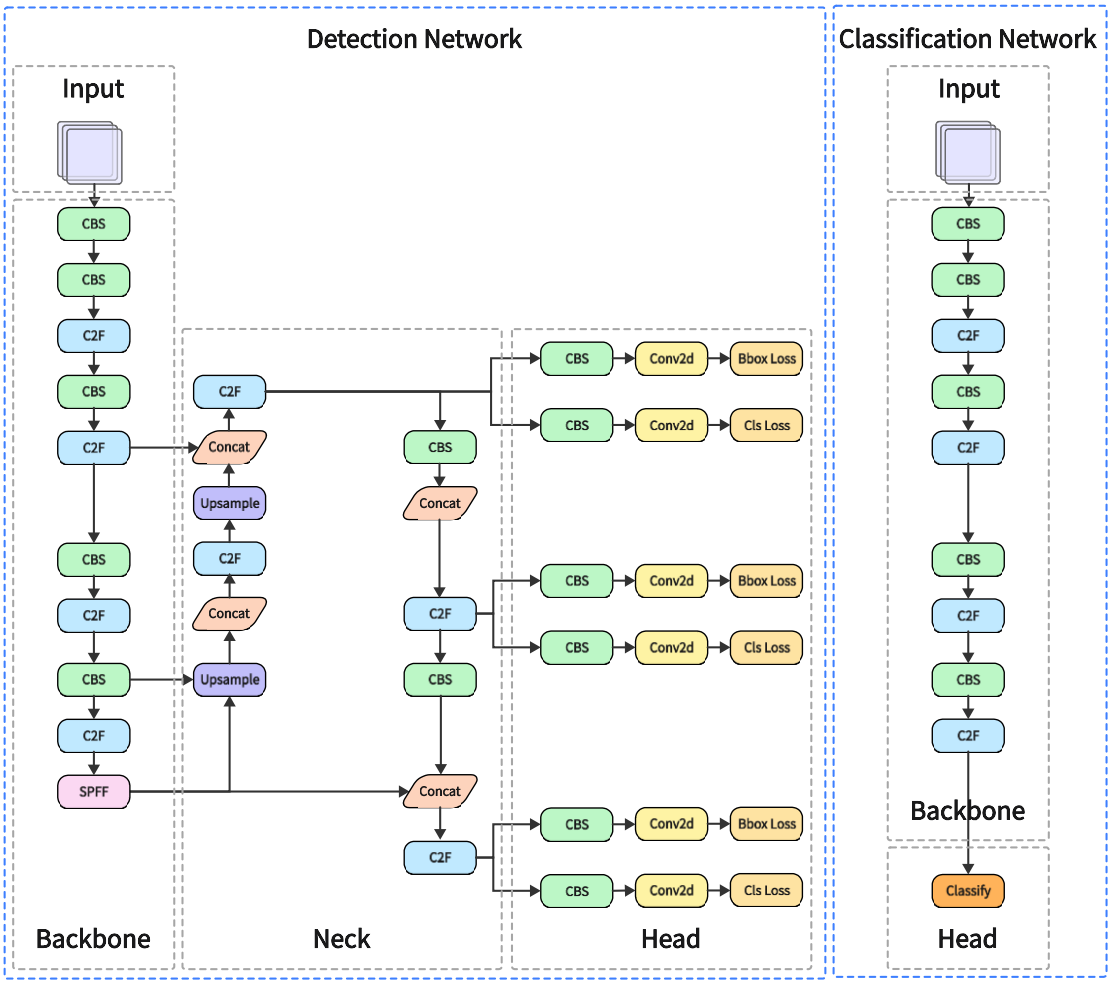


Figure1: YOLOv8s network architecture diagram

Although YOLOv8s inference is faster and more accurate, it still suffers from the following problems:

(1) Faster reasoning may come with a loss of accuracy;

(2) For tasks on small devices, the YOLOv8s model still has a large number of parameters and high model complexity;

(3) YOLOv8s model is ineffective for small target detection

Therefore, YOLOv8s model can be further optimized to reduce the number of model parameters, complexity, and improve the training accuracy.

2.3 Improved YOLOv8s modeling

In this paper, we propose the following improvements to address the problems of large number of YOLOv8s parameters, high model complexity, low training efficiency, and low accuracy: optimize the architecture of the backbone network by replacing the CBS and C2F modules with multiple ShuffleNet-v2 serial connections; add a global multi-head attention module (MHSA) to the last step of the backbone network to capture the global information and improve the model's generalization ability while balancing the loss of accuracy due to lightweighting; Stochastic Gradient Descent (SGD) is used as the optimizer. The improved network structure is shown in Figure 2.

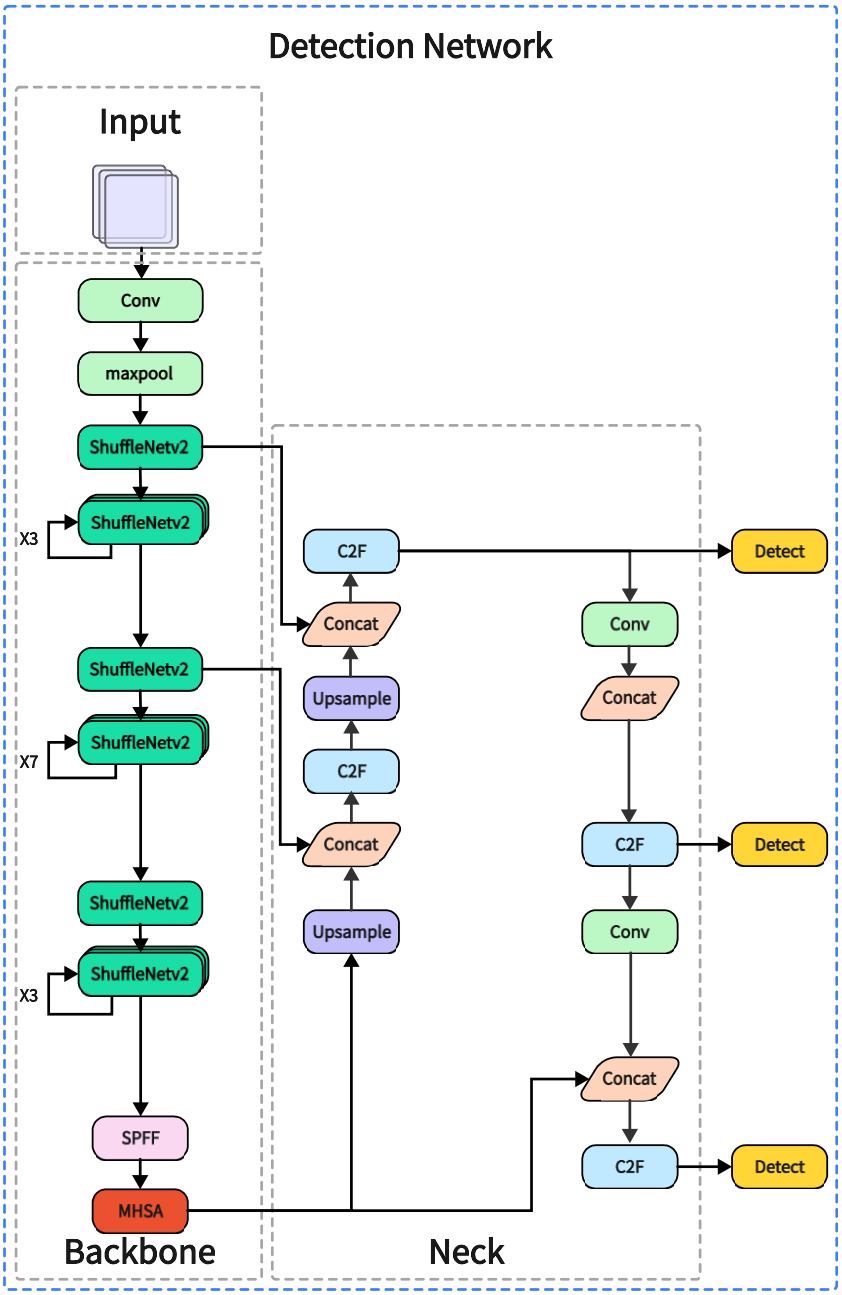
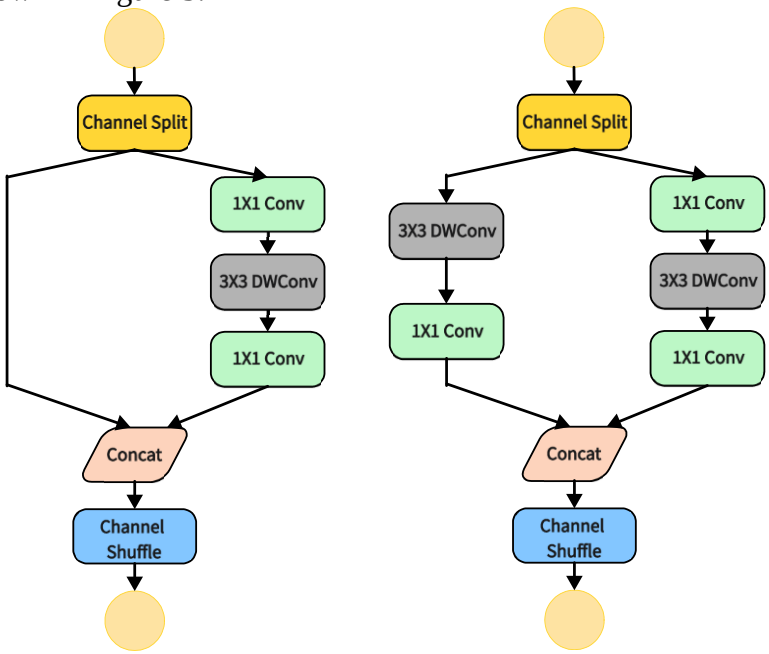


Figure2: Diagram of the improved YOLOv8s network structure

2.3.1 ShuffleNet-v2

Aiming at the problem of huge computational volume of YOLOv8s, this paper proposes to utilize ShuffleNet-v2 to reduce the number of model parameters and achieve the purpose of lightweighting.[13] ShuffleNet-v2 is a lightweight network module, which adopts the technique of group convolution, dividing the input channels into several smaller groups, performing convolution operation in each group separately, and then merging them at the end, which greatly reduces the number of parameters. At the same time, it introduces the operation of channel rearrangement, which improves the representation ability of the network, reduces the redundancy information, and further reduces the complexity of the model, and makes use of the structure of dimensionality reduction, convolution, and then dimensionality enhancement. The v2 version introduces the Channel Split

operation on the basis of the original, the outputs of the two branches are concatenated and then Channel Shuffle to get the final result, which enhances the generality of the model. The structure of ShuffleNet-v2 is shown in Figure 3.



(a) ShuffleNet-v2 basic unit. (b) ShuffleNet-v2 unit for spatial down sampling

Figure3: ShuffleNet-v2 network structure

2.3.2 Multi-pronged self-attention mechanisms

Since ShuffleNet-v2 achieves lightweighting by reducing the number of parameters and computation, it brings certain loss of accuracy and has limitations in dealing with complex scenes, small target detection, etc. In order to balance these losses and to increase the model generalization ability so that it can be used in a variety of scenarios, this paper proposes to utilize the global multi-head self-attention mechanism[14] which acquires global information, learns the correlation between features of different scales and fuses them, enhances the semantic information of the features, combats the influence of occlusion and complex background, and improves the robustness and accuracy of target detection.The structure of MHSA is shown in Figure 4.

Assume that M heads are used, i.e., there are M different QKV, $Q_i = (q_{i1}, q_{i2}, ..., q_{iM})$, $K_i = (k_{i1}, k, ..., k_{iM})$, $V_i = (v_{i1}, v_{i2}, ..., v_{iM})$, for each query there:

$$Q_i = x_i * W_{qi} \tag{1}$$

Where W_{qi} is the hyperparameter, and idem:

$$K_i = x_i * W_{ki} \tag{2}$$

$$V_i = x_i * W_{vi} \tag{3}$$

After getting the query and the corresponding key value, it is then normalized by *softmax* and finally multiplied with V_i to get the output:

$$y_i = Attention(Q_i, K_i, V_i) = softmax(\frac{Q_i \cdot K_i^T}{\sqrt{d_k}})V_i \tag{4}$$

d_k is the scaling factor and y_i is the output of carrying out an attention mechanism, since there are M attention heads, all there will be M outputs of y_i, and finally all the y_i are connected to get the final output:

$$y = concat(y_1, y_2, ..., y_M) \tag{5}$$

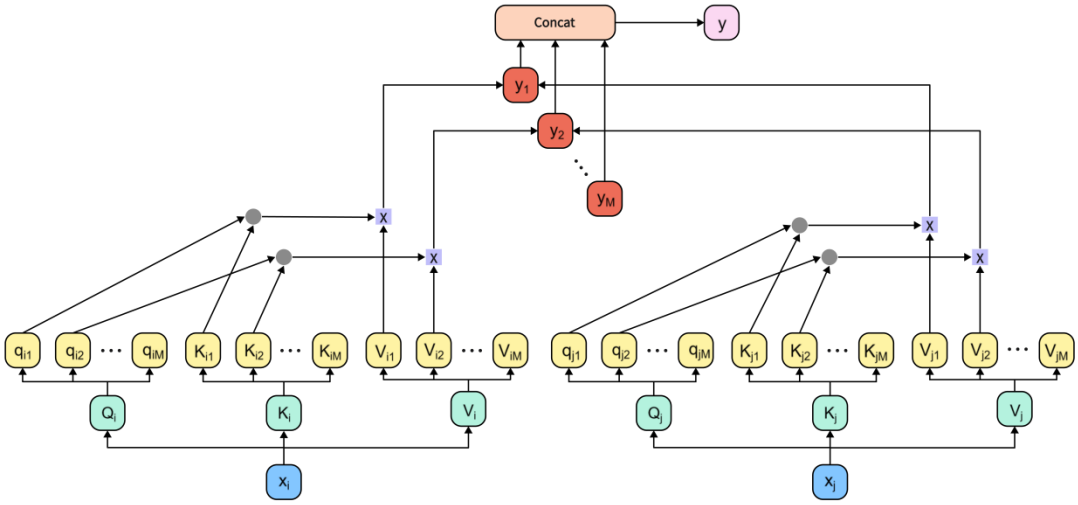


Figure4: Multi-head Attention Mechanism

2.3.3 Stochastic Gradient Descent

In order to improve the training efficiency, this paper uses stochastic gradient descent algorithm to optimize the parameters to improve the accuracy of the[15] .

In deep learning the objective function is usually the loss function averaged across the samples in the training dataset.

$$f(x) = \frac{1}{n}\Sigma_{i=1}^{n} f_i(x) \tag{6}$$

Where $f_i(x)$ is the loss function corresponding to the *ith* sample and $f(x)$ is the objective loss function. For the stochastic gradient descent method, the gradient is calculated by the formula:

$$\nabla f(x) = \frac{1}{m}\sum_{i=1}^{n} f_i(x) \tag{7}$$

Its computational complexity is $O(1)$, which can be drastically reduced since m does not change significantly as n increases.

3. Experiments

3.1 Introduction to the dataset

The images for STL-10 are from ImageNet, with a total of 113,000 RGB images of 96 * 96 resolution, of which 5,000 are in the training set, 8,000 are in the test set, and the remaining 100,000

are unlabeled images, and a total of 10 categories are included (airplanes, birds, cars, cats, dogs, horses, monkeys, boats, and trucks), with 500 training samples for each category and 800 test samples for each category. The unlabeled images contain not only the ten categories mentioned above, but also other unlabeled animal and vehicle images.

In this paper, we use STL-10 public dataset for model training and validation. Using the five categories of airplanes, birds, cars, ships, and trucks as targets, the training, test, and validation sets are divided using 5:4:4, i.e., the training set contains 2,500 images, the test dataset contains 2,500 images, and the validator contains 25,000 images, and the size of the input images is uniformly 96 * 96 pixels.

The operating system for this experiment is Windows 11, CPU is AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz, and 32 GB of RAM. Pycharm is utilized as the experimental platform, and Pytorch-CPU 3.80 is used for the deep learning framework, as shown in Table 1. The experimental parameter settings are shown in Table 2.

Table1: Experimental environment settings

experimental environment	
operating system	Windows 11
CPU	AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz
random access memory (RAM)	32GB
Use of language	Python
Experimental platforms	Pycharm
organizing plan	Pytorch-CPU 3.80

Table2: Configuration of experimental parameters

Experimental parameters	
data set	STL-10
Number of cycles	300
Batch size	64
Input Resolution	96 × 96
optimizer	SGD
loss function	BCE Loss
learning rate	0.01

3.2 Experimental results

In this paper, the following evaluation metrics are used to measure the performance of the network: precision (P), loss curve (Loss Curve), and mean Average Precision (mAP). The formula for each metric is as follows:

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$mAP = \frac{1}{N} \sum_{i=0}^{n} i * \int_0^1 P * Rdr \tag{10}$$

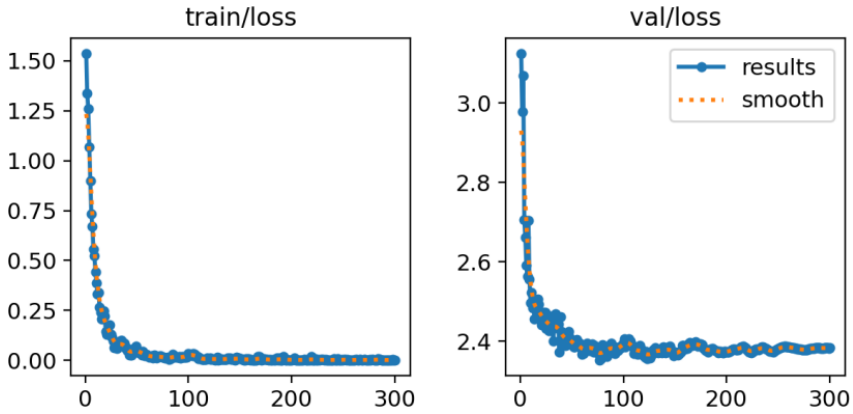


Figure5: Loss variation curves for training and test sets

As can be seen inFigure5: , the loss of the training set and the test set decreases rapidly, and finally converges to 0. The convergence is fast, the model can learn the data features effectively, and the performance on the validation set is stable, and the training effect is good.

Table3: Performance comparison of different optimizers

	airplane	bird	car	ship	truck	mAP
AdamW	0.85	0.94	0.90	0.88	0.84	0.883
Adam	0.71	0.82	0.68	0.77	0.63	0.722
Adamax	0.90	0.94	0.92	0.90	0.85	0.902
NAdam	0.75	0.88	0.86	0.83	0.73	0.812
RAdam	0.93	0.94	0.86	0.82	0.92	0.895
SGD	**0.96**	**0.98**	**0.96**	**0.91**	**0.89**	**0.942**

As can be seen fromTable3, SGD has the highest mAP and outperforms the other optimizers when the ShuffleNet-v2 and MHSA modules are not added, reflecting the superiority of SGD.

In addition to this, the number of network layers (Layers), training time, number of parameters (Params) and the number of floating point operations per second (GFLOPs) at 1 billion operations per second also need to be evaluated as the model proposed in this paper incorporates the ShuffleNet-v2 and the MHSA module, which is needed to quantify the lightweighting effect of the network. The different network architectures in the table use SGD as the optimizer with epoch set to 300 rounds.

Table4: Comparison of the lightweighting effect of different network architectures

	Params	GFLOPs	Training time	mAP	Layers
YOLOv8s	1605509	3.4	2.018	0.745	79
YOLOv8s with ShuffleNetv2	430769	0.4	0.827	0.705	80
YOLOv8s with MHSA	1802885	3.6	2.394	0.726	84
YOLOv8s with ShuffleNetv2 and MHSA	**628145**	**0.6**	**0.827**	**0.748**	**85**

Table4: . shows that ShuffleNet-v2 can greatly reduce the number of parameters and computation of the model, and at the same time, the complexity of the YOLOv8s model with the addition of the MHSA module increases slightly, which is in line with the expected effect, and the YOLOv8s model with the fusion of ShuffleNet-v2 and MHSA can effectively balance the accuracy and the lightweight, and ensure the accuracy while greatly reducing the number of parameters and the computation, and the expected effect is achieved. The expected effect is achieved.

4. Conclusions

Aiming at the current traditional YOLOv8s model with low accuracy and complex model, this paper proposed a new lightweight and performance-balanced YOLOv8s network structure by replacing the original C2F module with ShuffleNet-v2. Meanwhile, in order to further improve the accuracy degradation due to the decrease in the number of parameters, this paper added a global multi-attention mechanism to obtain the global information, which is used to learn the correlation between the features at different scales and fuse them to enhance the semantic information of the features, and adopted the SGD as an optimizer to further improve the accuracy. Experiments on the STL-10 dataset showed that the introduction of ShuffleNet-v2 and MHSA effectively reduced the number of parameters of the model, significantly reduced the training time, and the accuracy was impressive, and compared with other optimizers SGD improves the performance the most, which is excellent in the balance of lightweight and algorithmic performance.

References

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788.

[2] Zaidi S.S.A., Ansari M.S., Aslam A., et al. A Survey of Modern Deep Learning based Object Detection Models [J]. Digital Signal Processing, 2022: 103514.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. cvpr 2001, Kauai, HI, USA, 2001, pp. 1-1.

[4] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580-587.

[5] Carion N., Massa F., Synnaeve G., et al. End-to-End Object Detection with Transformers [M]//Computer Vision - ECCV 2020, Lecture Notes in Computer Science. 2020: 213-229.

[6] Qin, X., Zhang, Z., Huang, C., et al. U2-Net: Going deeper with nested U-structure for salient object detection [J]. Pattern Recognition, 2020: 107404.

[7] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 6517-6525.

[8] Redmon J., Farhadi A., YOLOv3: An Incremental Improvement. [J]. arXiv: Computer Vision and Pattern Recognition, 2018.

[9] Zhang X., Gao Y., Wang H., Wang Q. Improve YOLOv3 using dilated spatial pyramid module for multi-scale object detection [J]. International Journal of Advanced Robotic Systems. 2020; 17(4).

[10] Bochkovskiy A., Wang C.Y., Liao H.Y.M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. Cornell University - arXiv, 2020.

[11] Jiao S, Wang C, Gao R, et al. Harris Hawks Optimization with Multi-Strategy Search and Application[J].Symmetry, 2021.DOI:10.3390/sym13122364.

[12] Li C Y , Li J , Chen H L ,et al.Enhanced Harris hawks optimization with multi-strategy for global optimization tasks[J].Expert Systems with Application, 2021(Dec.):185.DOI:10.1016/j.eswa.2021.115499.

[13] Sun S., Han L., Wei J., et al. ShuffleNetv2-YOLOv3: a real-time recognition method of static sign language based on a lightweight network [J]. SIViP 17, 2721-2729 (2023).

[14] Vaswani A., Shazeer N.M., Parmar N., et al. Attention is All you Need [J]. Neural Information Processing Systems, 2017.

[15] Goyal P., Dollár P., Girshick R., et al. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour[J]. arXiv: Computer Vision and Pattern Recognition, 2017.