Article

GC-YOLOv3: You Only Look Once with Global Context Block

Yang Yang and Hongmin Deng *

School of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China; argus@stu.scu.edu.cn

* Correspondence: hm_deng@scu.edu.cn

Received: 27 June 2020; Accepted: 29 July 2020; Published: 31 July 2020

Abstract: In order to make the classification and regression of single-stage detectors more accurate, an object detection algorithm named Global Context You-Only-Look-Once v3 (GC-YOLOv3) is proposed based on the You-Only-Look-Once (YOLO) in this paper. Firstly, a better cascading model with learnable semantic fusion between a feature extraction network and a feature pyramid network is designed to improve detection accuracy using a global context block. Secondly, the information to be retained is screened by combining three different scaling feature maps together. Finally, a global self-attention mechanism is used to highlight the useful information of feature maps while suppressing irrelevant information. Experiments show that our GC-YOLOv3 reaches a maximum of 55.5 object detection mean Average Precision (mAP)@0.5 on Common Objects in Context (COCO) 2017 test-dev and that the mAP is 5.1% higher than that of the YOLOv3 algorithm on Pascal Visual Object Classes (PASCAL VOC) 2007 test set. Therefore, experiments indicate that the proposed GC-YOLOv3 model exhibits optimal performance on the PASCAL VOC and COCO datasets.

Keywords: object detection algorithm; single-stage detector; GC-YOLOv3; learnable semantic fusion; self-attention mechanism; global context block

1. Introduction

In recent years, deep learning [1] has been popularly accepted all over the world, and exhibits better robustness and higher accuracy than traditional methods. With the rapid development of computer vision technology, researches on object detection algorithms have aroused extensive interest and become a research hotspot [2]. The traditional object detection method can be divided into three steps: (1) region selection—the region of interest can be selected using the sliding window technique with different scales. Despite the redundancy, this method could be used to mark all possible locations with arbitrary scanning. (2) Feature extraction—features can be extracted from labeled regions. Manual feature extraction methods, such as Scale Invariant Feature Transform (SIFT) [3], Histogram of Oriented Gradient (HOG) [4] and Haar-like [5], are commonly used for extracting features from anchor boxes. These traditional methods cannot extract all of the features because of poor robustness and inadaptability to changing shape and light conditions. (3) Classification and regression—all extracted features can be classified [6] and the detection box can be predicted. One channel of outputs is used to determine the category of objects and four channels are used to predict the coordinates of the diagonal points of object boxes.

Generally, the CNN-based object detectors could be divided into two types: single-stage object detectors and double-stage object detectors. Double-stage object detection models, such as Faster Region proposal with Convolutional Neural Network (Faster-RCNN) [7], have achieved good detection results in industrial fields and practical application scenarios. Though Region Proposal Network (RPN) [8] involved in double-stage object detection could roughly determine the regions where

the object to be detected might be contained in the generation phase of the region of interest, thus significantly improving its accuracy with some time loss, single-stage object detection models, such as You-Only-Look-Once (YOLO) [9] and Single Shot multibox Detector (SSD) [10], could greatly reduce detection time, where the object detection problem was treated as a regression process.

Inspired by these studies, a better YOLO algorithm, namely Global Context You-Only-Look-Once v3 (GC-YOLOv3), is proposed in this paper based on global context block and learnable semantic fusion. The originality of this paper lies in the following two aspects: (1) a global context block is added between the feature extraction network and the feature pyramid network to enable feature maps of attention to be transferred to deeper networks. A self-attention mechanism is employed, which is different from the usual mechanisms, to determine the pixel dependencies of object features. On the other hand, different from non-local networks [11], the global context block does not use multiple convolutional layers to extract space and channel features. Instead, only a convolutional layer and a Rectified Linear Unit (ReLU) activation function are used to make attention more non-linear. In this way, the global context block reduces a lot of computational burden compared with previous self-attention mechanisms. (2) Learnable semantic fusion is used to fuse different feature maps produced in the feature extraction network to make better use of the feature extraction network's outputs. The previous fusion method [12] was to splice two tensors directly according to the two channels and then reduce the dimension with 1×1 convolution, where the validity between channels could not be utilized to learn useful channel branches. By contrast, outputs of different convolutional layers are assigned different weights to highlight useful parts in this paper.

The experiment results indicate that GC-YOLOv3 outperforms the original YOLOv3 algorithm without introducing excessive parameters.

2. Related Works

2.1. Object Detection

Nowadays, object detection algorithms are mainly improved by using deep neural networks that can extract more abundant features, combining multi-scale features together for detection. Fu et al. [13] proposed to use a deeper ResNet-101 [14] network for extracting features and a deconvolutional layer for replacing the convolutional layer to improve the accuracy of the model at the cost of adding too many parameters and computations. Shen et al. [15] also proposed a method to improve detection accuracy by taking DenseNet [16] as an example, where the pre-training weights were not required in the training object detection model due to the stem block structure. Although this approach does not make the whole network vertically deeper, the stem block structure would increase the width of the network laterally. Hence, the above-mentioned two approaches would cause too many computations. Bodla et al. [17] proposed to use a soft Non-Maximum Suppression (NMS) algorithm for reducing the confidence of scoring boxes whose overlaps exceeded the threshold, instead of directly suppressing a non-maximum algorithm, with an 1% mean Average Precision (mAP) improvement without increasing the training cost and the number of parameters. This is an example of optimizing the selection of candidate boxes. Too much invalid information extracted from the candidate boxes will inevitably affect the generated object box in this approach. In the aforementioned works, the performances of detection algorithms were improved to different degrees. However, it is more important to find a way to make full use of effective information to enhance the detection ability of the model without adding too many parameters.

2.2. Self-Attention Algorithm

In order to make good use of the limited visual information processing resources, a human needs to select a specific part of the visual region and focus on it. The attention mechanism mimics the internal process of biological observation and combines internal experiences with external sensation to improve the accuracy of observing certain areas. In a Convolutional Neural Network (CNN), a close

relationship is established in the convolutional layer among local pixels using convolution operations, and a remote dependency relationship is modeled by deepening convolutional layers. However, networks with additional convolutional layers entail a large computational cost for forward reasoning and are difficult to optimize. It becomes impossible to build models through remote dependencies, in part because it is more difficult to transmit information among remote pixels. To address this issue, a non-local network [11] was proposed to model long-range dependencies via a self-attention mechanism [18]. In this way, global dependencies were indeed established in the network and the detection effect of the network was improved. However, different query positions in the same picture got almost the same global context information from non-local structures. The query-specific global context information obtained by complex computations and parameters makes little sense. Therefore, a simplified self-attention network structure is needed.

2.3. Semantic Fusion

In the deep convolutional network, feature maps at different levels represent the objects from different perspectives: the higher feature map encodes more semantic information of the objects and acts as a category detector; the lower level of the feature map carries more identification information and can separate objects from similar distractors [19]. The traditional feature fusion method [12] was to directly splice the multi-layer outputs together according to channels, and then use the 1×1 convolutional layer to reduce the channel dimension. This approach was aimed at utilizing useful information from outputs of similar semantics without adding too much computation. However, this method could not properly extract the internal information of feature maps. In view of this, the outputs of the same semantic were assigned different weights to judge the importance of different layers so that the correlation information among convolutional layers could be better utilized to show the difference among the outputs of the feature extraction network.

In Table 1, we list the performances and limitations of the algorithms mentioned in the related work.

Table 1. Performances and limitations of different methods.

Methodology	Performance	Limitation
Non-local network [11]	High accuracy by using self-attention mechanism [18]	Complex computations and parameters
Direct fusion [12]	High accuracy by concatenating different feature maps at different levels [19]	Some internal information lost
Deconvolutional Single Shot Detector (DSSD) [13]	High accuracy by deepening the ResNet-101 [14] network	Too many parameters and computations
Deeply Supervised Object Detector (DSOD) [15]	High accuracy by taking the DenseNet [16] network as an example	Wide network
Soft Non-Maximum Suppression (NMS) [17]	High accuracy by reducing the confidence of scoring boxes	Valid information lost

3. The Proposed Method

3.1. GC-YOLOv3

The YOLO algorithm was proposed by Redmon et al. [9] in 2015 to significantly accelerate object detection. Unlike YOLO and YOLOv2 [20], YOLOv3 [21] maintained the detection speed of the former two while greatly improving the detection accuracy.

The multi-scale network structure introduced in YOLO exhibits excellent performance in object detection. Undoubtedly, the network structure is deeper in YOLO while achieving better results. There are 52 convolutional layers in the feature extraction network called Darknet-53. Generally, the deep network lacks detailed information and thus it is difficult to detect small objects. One of the solutions to this issue is to build a Feature Pyramid Network (FPN), where the feature maps contain more detailed information. Since single-stage detectors achieve a trade-off in speed and accuracy, their performance will be negatively affected if the network is deepened. Inspired by the above-mentioned works on YOLO, we propose a GC-YOLOv3 network to keep the depth of network

layers while the detector could have a better performance in object detection. GC-YOLOv3 keeps the network structure of YOLOv3 and introduces global context block and learnable fusion between the feature extraction network and the feature pyramid network. The structure of the proposed GC-YOLOv3 is shown as in Figure 1, where Figure 1a is the network structure of GC-YOLOv3 and Figure 1b is the specific structure of the convolutional set in the FPN of GC-YOLOv3. The red part in Figure 1 indicates the innovation in this paper, whereas the dashed line represents the branch of the feature extraction network output.

Figure 1. The structure of GC-YOLOv3. (a) The main framework of GC-YOLOv3. (b) Convolutional set.

3.2. Global Context Block

Although the self-attention mechanism performs well on the object detection task, it also has the following disadvantages: (1) too many parameters and too much computation; (2) the network is too deep and there are too many difficulties to optimize; and (3) “global” has a maximum distance limit. Therefore, a more comprehensive structure (global context block) is proposed to achieve the same effect as the self-attention mechanism and simplify the calculation and network depth. Based on the self-attention mechanism, we add a module called global context block to the GC-YOLOv3 structure and build a long-range dependency relationship among all feature pixels in the feature map itself so that GC-YOLOv3 can focus on different regions and the objects can be detected more effectively. The global context block could be regarded as a simplified version of a non-local network [11]. On the one hand, the global context block only uses a mask-convolution layer to obtain the mask in the context modeling compared with the multi-convolutional layer structure of non-local network. In the transform, two convolutional layers are adopted to compress the dimension of channels and reduce the number of parameters. On the other hand, although the global context block is simpler than the non-local network, it assigns different weights to input features from spatial and channel dimensions to highlight useful information. Generally, the global context block is lighter than the non-local network without accuracy loss. Figure 2 shows the structure of the global context block, where $C \times H \times W$ denotes channel C , height H and width W of the input feature, respectively. $\odot$ denotes the matrix multiplication and $\oplus$ denotes the broadcast element-wise addition.

Figure 2. Architecture of the global context block.

The global context block is used to enhance our model. In sequence transformation, a set of key features for mapping to the output are connected to the input and output sequence using global dependence in a self-attention mechanism. When the attention mechanism is used for image recognition, the information processed by the attention mechanism module is included in an explicit sequence of items as shown in Equation (1):

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n), \alpha_i \in R^D \quad (1)$$

where n represents the number of feature vectors and D represents the dimensional space. The formula of the attention is expressed as in Equations (2) and (3):

$$F(\cdot) = \sum_{j=1}^{N_p} \frac{e^{W_{k \times j}}}{\sum_{m=1}^{N_p} e^{W_{k \times m}}} \times \alpha \quad (2)$$

$$\delta(\cdot) = \alpha + W_{v2}(\text{ReLU}(\text{LN}(W_{v1}(F(\cdot))))), W_{v1} \in \mathbb{R}^{rC \times H \times W}, W_{v2} \in \mathbb{R}^{C \times H \times W} \quad (3)$$

where $F(\cdot)$ is the weight of attention generated from the Softmax function, $\delta(\cdot)$ denotes the bottleneck transform and N_p represents the dimension of input feature maps. A method of compressing the channel dimension in the global attention pooling is expected to further highlight the spatial information without any other channel information. Therefore, the 1×1 convolution W_k and Softmax function are adopted to generate the weights of attention and obtain the global context features. In other words, the usefulness of each point in space is regarded as a probability, and each weight of the corresponding global attention maps is thus the probability weighted in space. To make sure that the sum of the probabilities of each prediction is equal to one, the Softmax function is used to ensure that useful information is not out of range.

In the bottleneck transform, a double convolution module is added to improve the performance of the channel attention. After context modeling, the feature maps are respectively processed by the head convolutional layer, layer normalization, ReLU layer and the last convolutional layer to get the output. W_{v1} and W_{v2} are two 1×1 convolutional layers whose input feature map size is $H \times W$ in the feature space and number of channels are $C \times r$ and C , respectively. C represents a channel of input feature maps, and r represents a reduction ratio of the dimension, where r is equal to $1/16$. Furthermore, the ReLU layer is inserted between the two convolutional layers to indicate its nonlinearity. Since the two-layer bottleneck transform makes optimization more difficult, Layer Normalization (LN) is added into the bottleneck transform (in front of the ReLU) for simplifying the network to facilitate generalization.

The global context block also has the following deficiency: when the input feature map of the global context block is very large, multiplying the tensors (*Batchsize*, C , $H \times W$) after the Softmax function with the original input features will consume a large amount of memory and computation. That means it could not efficiently deal with large input features. There are other solutions, such as scaling, but they would lead to losing some useful information. Therefore, the global context block is placed between the feature extraction network and the feature pyramid network rather than embedded in the residual block so as to avoid computation redundancy caused by excessive large input feature maps.

3.3. Learnable Semantic Fusion

The essence of attention is a process that reduces one dimension to obtain the weights of other dimensions so that the network would pay “attention” to what it should learn and abandon irrelevant information, in which “distribution” is the most important part. By assigning larger weights to useful information, the network uses some layers more efficiently to obtain more accurate detection results and reduce training costs. In this sense, fusion is another attention mechanism. Based on the idea of DenseNet [16], all the same output feature maps between the feature extraction network and the feature pyramid network are used to maximize information transmission. Therefore, this mechanism makes full use of the output of each residual block of the same channel in our feature extraction network and optimizes the network by assigning different importance coefficients to different feature maps of each residual so as to obtain different degrees of attention at each layer. This approach is more efficient and beneficial by putting the outputs of the same channels and sizes together without reducing dimension and sampling. Figure 3 shows the structure of the learnable fusion, where (a) represents the main framework and (b) is the specific description of convolutional blocks in (a).

Figure 3. Illustration of learnable fusion architecture. (a) The main framework of learnable fusion. (b) Convolutional block.

A new method of feature fusion, as showed in Figure 3, is to define a process that is different from the previous methods in this work. Our objective is to build a block that could learn features, where useful feature maps could be close to the ground-truth. Therefore, different weights are assigned to different feature maps in order to highlight some useful features. For simplicity, we denote the collection of fusion layer parameters as an array W . Suppose that in the feature extraction network we have M output layers. Each output layer is associated with the fusion layer, in which the corresponding weights are denoted as Equation (4):

$$W = (W_1, W_2, \dots, W_i, \dots, W_M) \quad (4)$$

The function of the fusion is in Equation (5):

$$f_{out} = \sum_i^M W_i^T \times f_{in}^i \quad (5)$$

where W_i^T represents the transpose of W_i and f_{in}^i denotes the i -th feature-level input for learning fusion. Each input component in the fusion layer is multiplied by a weight and f_{out} is the weighted sum of all the feature-level inputs.

4. Implementation

4.1. Dataset

Pascal Visual Object Classes (PASCAL VOC) [22] and Common Objects in Context (COCO) [23] datasets were used for training and testing in our experiment. We conducted our experiments with the PASCAL VOC 2007, PASCAL VOC 2012 and COCO 2017 datasets. The average number of objects on a COCO dataset, especially the small ones in a single picture, was about twice bigger than that of PASCAL VOC. In terms of size and difficulty of detection, the training and testing on the COCO dataset were much more difficult than those of the PASCAL VOC datasets. In Table 2, we list the differences

between the PASCAL VOC and COCO datasets in various aspects. “07 + 12” in Table 2 means the PASCAL VOC dataset, which includes PASCAL VOC 2007 trainval and PASCAL VOC 2012 trainval.

Table 2. Comparison of the differences between the PASCAL VOC and COCO datasets.

Dataset	COCO 2017	PASCAL VOC (07 + 12)
Number of categories	80	20
Number of training pictures	117,264	16,551
Number of testing pictures	5000	4952
Total sample boxes	902,435	52,090
Total sample boxes / total number of images	7.4	2.4

4.2. Data Augmentation

We often encountered the situation that we could not find sufficient data to use in our research. For example, a task with only a few hundred pieces of data will sometimes appear in our daily training. However, the most advanced neural networks currently need thousands of images to facilitate generalization. Only when there is a large amount of data can the algorithm show strong performance. However, it is often difficult to cover all scenarios during data collection. For example, it is difficult to control the proportion of light during image data collection. Therefore, in the pre-processing stage of training, data augmentation techniques shown in Table 3 are usually used to improve the overall capability of the model. RGB pictures with three channels are transformed into an Hue Saturation Value (HSV) color space and the brightness and saturation are each augmented with a probability of 50%. Image clipping, affine transform, and flipping are all operated on the input image with a probability of 0.5, which means that there is a 50% probability of flipping the original image 180 degrees on the horizontal axis, cropping the image with random lengths and widths and mapping the image on the horizontal axis, and a 50% probability the input image is not changed.

Table 3. Specific settings of data augmentation.

Type	Parameter
HSV Saturation	50% probability
HSV Intensity	50% probability
Random Crop	50% probability
Random Affine	50% probability
Random Horizontal Flip	50% probability

4.3. Network Setting

Pytorch was used to train our model on the Nvidia GTX 1080Ti GPU. The Stochastic Gradient Descent (SGD) algorithm was used for optimizing weights, with a momentum of 0.9, a decay of 0.0005 and an initial learning rate of 0.0001. We set the input image size to 416×416 in the training stage with reference to the original YOLOv3 model.

5. Experiments

Several comparative experiments (on PASCAL VOC and COCO) were conducted in this study. Firstly, ablation experiments were performed on the PASCAL VOC 2007 dataset with our proposed model to examine the impact of the module we added to the model. Secondly, two large public datasets (PASCAL VOC and COCO) were used to test the proposed model with other state-of-art methods, and the results verified the theoretical and practical feasibility of GC-YOLOv3. The mean Average Precision (mAP) was used to evaluate the “quality” of an object detection algorithm in our paper. For the PASCAL VOC dataset, we used the PASCAL VOC 2012 evaluation benchmark to guide our model. Additionally, we used mAP@0.5 (the mAP when the Intersection Over Union (IOU) is 0.5) as a performance index to evaluate our model and other ones on the COCO dataset.

5.1. Ablation Study on PASCAL VOC 2007

An ablation study on the VOC 2007 dataset was conducted to investigate the effects of learnable feature fusion and global context block on the units of accuracy and speed. Four different combinations, namely YOLOv3, YOLOv3 with fusion, YOLOv3 with attention, and YOLOv3 with fusion and attention, were studied. These models were trained on the PASCAL VOC (07 + 12: 2007 trainval and 2012 trainval) dataset. Time was evaluated on a single GTX 1080TI GPU and the test image size was 544×544 in the ablation experiments. As can be seen from Table 4, the attention module with learnable fusion that we proposed in this paper was superior to the other common methods in learning useful information and improving the performance of common datasets.

Table 4. Ablation study on the PASCAL VOC2007 test dataset.

Method	Backbone	Global Context Block	Learnable Fusion	Time (ms)	mAP (%)
YOLOv3	Darknet53			25.14	78.6
YOLOv3	Darknet53	√		27.18	80.1
YOLOv3	Darknet53		√	30.47	81.2
YOLOv3	Darknet53	√	√	32.25	83.7

5.2. Performance Improvement on PASCAL VOC 2007

Our GC-YOLOv3 was trained on the PASCAL VOC 2007 dataset for 50 epochs by cosine-decay-learning-rate, which means the learning rate could be modified according to a cosine function. Figure 4 shows the Precision and Recall (P–R) maps of our GC-YOLOv3 for 20 categories of the PASCAL VOC 2007 test dataset. Figure 5 shows the detection results of GC-YOLOv3. It can be seen from Figures 4 and 5 that GC-YOLOv3 detected most medium and large targets very well but the detection effect on overlapping objects (such as chair and bottle) was not as good as that on single targets (such as bus, car, train and person), because GC-YOLOv3 is trained and detected based on an image grid cell. Specifically, if two targets are close to each other in the same cell during the training, the front object will enter the training program and the rear will be ignored. In other words, if two targets are too close to the center of the feature map, one of them will be rewritten and unable to be trained.

Figure 4. P–R maps for the PASCAL VOC 2007 test dataset.

Figure 5. Detection results of GC-YOLOv3.

The effects of each model on the PASCAL VOC 2007 test dataset were also compared, as shown in Table 5. “07 + 12” in Table 5 means the training set is PASCAL VOC 2007 trainval and PASCAL VOC 2012 trainval, and the input image size is 416×416. In order to compare with other detectors, we set the input image size to 320×320 and 544×544 respectively in the test. It can be seen in Table 5 that the detection accuracy of the GC-YOLOv3 algorithm was the best among similar detection algorithms.

Table 5. Comparison of different detection frameworks (including single-stage detector and double-stage detector) on the PASCAL VOC 2007 test.

Method	Backbone	Train Data	mAP	Size	FPS	GPU
Faster R-CNN [7]	VGG16	07 + 12	73.2	1000 × 600	7	Titan X
Faster R-CNN [7]	ResNet101	07 + 12	76.4	1000 × 600	2.4	K40
R-FCN [24]	ResNet101	07 + 12	79.5	1000 × 600	9	Titan X
RetinaNet300 [25]	ResNet101	07 + 12	62.9	300 × 300	11.4	K80
RefineDet320 [25]	ResNet101	07 + 12	79.5	320 × 320	12.9	K80
SSD300 [10]	VGG16	07 + 12	77.1	300 × 300	46	Titan X
SSD321 [10]	VGG16	07 + 12	77.5	320 × 320	11.2	Titan X
YOLOv3 [21]	Darknet53	07 + 12	74.5	320 × 320	45.5	Titan X
GC-YOLOv3	Darknet53	07 + 12	81.3	320 × 320	39	1080Ti
RetinaNet500 [25]	ResNet101	07 + 12	72.2	500 × 500	7.1	K80
RefineDet512 [25]	VGG16	07 + 12	81.2	512 × 512	5.6	K80
SSD512 [10]	VGG16	07 + 12	79.5	512 × 512	19	Titan X
SSD513 [10]	ResNet101	07 + 12	80.6	513 × 513	6.8	Titan X
YOLOv3 [21]	Darknet53	07 + 12	78.6	544 × 544	40	Titan X
GC-YOLOv3	Darknet53	07 + 12	83.7	544 × 544	31	1080Ti

5.3. Performance Improvement on COCO Dataset

The COCO dataset is a large dataset for object detection and segmentation, which is aimed at understanding scenarios that are mainly extracted from complex daily scenes [23]. The locations of objects in the image are demarcated through accurate segmentation. The COCO 2017 dataset, whose trainval set consists of 80 object categories and 117,264 images, was used in our work. Our GC-YOLOv3 model evolved for 100 epochs with a batch size of 8. The cosine-decay learning rate was also used to train our model on the COCO 2017 dataset with an initial learning rate of 0.0001 and a minimum learning rate of 0.000001. The learning rate fluctuated along with the cosine function during the training. Furthermore, our model was validated on the COCO 2017 test-dev dataset and compared with current popular methods. In order to better measure the detection accuracy of the GC-YOLOv3, mAP@0.5 was adopted to comprehensively evaluate the detection performance [23]. As illustrated in Table 6, GC-YOLOv3 showed a more significant improvement than YOLOv3 and

achieved comparable results to similar algorithms, such as Region-based Fully Convolutional Network (R-FCN) [24], YOLOv2 [20], SSD300 [10], SSD500, SSD513, DSSD321 [13] and DSSD513. The COCO 2017 trainval dataset was used to train our model and the COCO 2017 test-dev dataset was used to test different models. The number in brackets in the model represents the size of input images. For instance, (608) indicates that the size of the input was 608×608 . Although GC-YOLOv3's detection accuracy was not as good as that of Retinanet [25], GC-YOLOv3 was better in speed. The reason is that GC-YOLOv3 only adds a few modules to the original basis, while Retinanet uses a deeper feature extraction network (ResNet-101 [14]) to make the detector perform better, which greatly increases the time cost of detection. YOLOv4 [26] outperformed GC-YOLOv3 in speed and accuracy because YOLOv4 integrates many advanced modules of object detection (such as Cross Stage Partial Network (CSPNet) [27], Path Aggregation Network (PANet) [28] and Spatial Pyramid Pooling (SPP) [29]) and uses various useful training methods (such as CutMix [30], hard-Swish [31] and Mish [32]). YOLOv4 is a milestone, but GC-YOLOv3 achieves good results without changing the YOLOv3 network architecture at a large cost. We can conclude that GC-YOLOv3 outperforms most existing methods in accuracy and speed except YOLOv4, but that it is a trade-off method regarding accuracy and cost compared with YOLOv4.

Table 6. COCO2017 test-dev detection results.

Model	Train Data	Test Data	mAP@0.5	FPS
R-FCN (416) [24]	COCO2017 trainval	COCO2017 test-dev	51.9	12
SSD (300) [10]	COCO2017 trainval	COCO2017 test-dev	41.2	46
SSD (321) [10]	COCO2017 trainval	COCO2017 test-dev	45.4	16
SSD (500) [10]	COCO2017 trainval	COCO2017 test-dev	46.5	19
SSD (513) [10]	COCO2017 trainval	COCO2017 test-dev	50.4	8
DSSD (321) [13]	COCO2017 trainval	COCO2017 test-dev	46.1	12
DSSD (513) [13]	COCO2017 trainval	COCO2017 test-dev	53.3	6
Retinanet-50(500) [25]	COCO2017 trainval	COCO2017 test-dev	50.9	14
Retinanet-101(500) [25]	COCO2017 trainval	COCO2017 test-dev	53.1	11
Retinanet-101(800) [25]	COCO2017 trainval	COCO2017 test-dev	57.5	5
YOLOv2(608) [20]	COCO2017 trainval	COCO2017 test-dev	48.1	40
YOLOv3(416) [21]	COCO2017 trainval	COCO2017 test-dev	55.3	35
YOLOv4(416) [26]	COCO2017 trainval	COCO2017 test-dev	62.8	38
GC-YOLOv3(416)	COCO2017 trainval	COCO2017 test-dev	55.5	28

5.4. Attention Mechanism Visualization

To better understand how the attention mechanism improves overall performance, we visualized the feature maps at different scales. In this work, the gradient-weighted class activation mapping (Grad-CAM) technique [33] was used to add the feature map processed by attention unit to the original image. The PASCAL VOC 2007 dataset was visualized as test dataset.

In Figure 6, the heatmap of YOLOv3, the heatmap of GC-YOLOv3 and the detection effect diagram of GC-YOLOv3 were stitched together, where each subfigure in the first column showed the heatmap of an original model, that in the second column was the heatmap of the improved model, and that in the third column was the detection diagram of GC-YOLOv3. The heatmaps showed that the regional attention of the model was positively correlated with the color depth. In some of the graphs (such as row 3 and column 1), high attention was only paid to one kind of object with less attention to the other kind of object, indicating that YOLOv3 could not focus on the places that we should pay attention to. Thus, more weight being given to a single object resulted in the low recognition rate. GC-YOLOv3 relies on more balance to establish a global dependency so that the model could not only focus on one object, but also on all useful information through pixel-level associations.

Figure 6. Visualization of attention maps on the PASCAL VOC 2007 test set: (a) Heatmaps of YOLOv3; (b) Heatmaps of GC-YOLOv3; (c) Detection effect diagrams of GC-YOLOv3.

We can see from Figure 6 that the attention map highlights the area where the actual object is, helping us intuitively feel that the attention mechanism can focus “the eye of the model” on the object we want to detect. The attention unit in the feature extraction network guides the model to concentrate on the units that we are interested in. For a given input image, the attention map is calculated according to the feature maps of different scales that highlight useful areas of different sizes. The attention map weights the spatial features of each location onto the original image, so attention can suppress the features of unrelated regions. In this way, the attention tries to help the model focus on the real objects, thus improving the detection accuracy of the model.

6. Conclusions

A novel YOLOv3 detector called GC-YOLOv3 with global context block and learnable feature fusion was proposed in this paper. Feature dependencies could be discovered and concerns could be focused on useful and relevant areas by using a fast attention unit in our GC-YOLOv3 model. The learnable fusion of weighted multi-layer branches makes great use of the output of the feature extraction network, enabling the head network to recognize objects in the feature maps better. GC-YOLOv3 greatly improves the accuracy with a small amount of additional computational cost,

showing better performance of feature extraction and recognition on two large datasets, and is easier to train without a large amount of parameter tuning.

The global context block achieves good performance, but it is also limited by inflexibility when applying to certain layers. In future work, we will study how to change the global context block structure to achieve adaptability, which will create favorable conditions to apply this block to different levels so as to improve the ability of extracting features and detecting objects.

Author Contributions: Conceptualization, methodology and writing—original draft preparation, Y.Y.; writing—review, editing and supervision, H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a Sichuan University research grant.

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

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
2. Szegedy, C.; Toshev, A.; Erhan, D. Deep neural networks for object detection. *Adv. Neural Inf. Process. Syst.* **2013**, *2*, 2553–2561.
3. Mortensen, E.N.; Den, H.; Shapiro, L.G. A SIFT descriptor with global context. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005.
4. Wang, X.; Han, T.X.; Yan, S. An HOG-LBP human detector with partial occlusion handling. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009.
5. Viola, P.A.; Jones, M.J. Rapid object detection using a boosted cascade of simple features. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; pp. 511–518.
6. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Conference and Workshop on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1137–1149.
8. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26–21 July 2017; pp. 2117–2125.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. SSD: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
11. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
12. Lee, H.; Kwon, H.; Robinson, R.M.; Nothwang, W.D.; Marathe, A.M. Dynamic belief fusion for object detection. In Proceedings of the WACV 2016: IEEE Winter Conference on Application of Computer Vision, Lake Placid, NY, USA, 7–9 March 2016; pp. 1–9.
13. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional single shot detector. *arXiv* **2016**, arXiv:1701.06659.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
15. Shen, Z.; Liu, Z.; Li, J. DSOD: Learning deeply supervised object detectors from scratch. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1919–1927.
16. Huang, G.; Liu, Z.; Maaten, L.V.D. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.

17. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS: Improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
19. Wang, L.J.; Ouyang, W.L.; Wang, X.G. Visual tracking with fully convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3119–3127.
20. Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26–21 July 2017; pp. 6517–6525.
21. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
22. Everingham, M.; Gool, L.V.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]
23. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
24. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
25. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.M.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Ana. Mach. Intell.* **2020**, *42*, 318–327. [[CrossRef](#)] [[PubMed](#)]
26. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
27. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of cnn. In Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition Workshop, Seattle, WA, USA, 14–19 June 2020.
28. Liu, S.; Qi, L.; Qin, H.-F.; Shi, J.P.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
29. He, K.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Ana. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
30. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. CutMix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6023–6032.
31. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.X.; Wang, W.J.; Zhu, Y.K.; Pang, R.M.; Vasudevan, V. Searching for MobileNetV3. In Proceedings of the the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
32. Mishra, D. Mish: A self-regularized nonmonotonic neural activation function. *arXiv* **2019**, arXiv:1908.08681.
33. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 336–359.