applied sciences

MDPI

Article

FFESSD: An Accurate and Efficient Single-Shot Detector for Target Detection

Wenxu Shi [1,2], Shengli Bao [1,2,*] and Dailun Tan [3,*]

[1] Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu 610081, China; shiwenxu17@mails.ucas.ac.cn
[2] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 10049, China
[3] School of Mathematics and Information, China West Normal University, Nanchong 637009, China
* Correspondence: baohigh@casit.com.cn (S.B.); dailun_tan@cwnu.edu.cn (D.T.)

Received: 17 September 2019; Accepted: 8 October 2019; Published: 12 October 2019

Featured Application: This work can be widely used in classification and recognition of image targets, especially for the tasks that need to detect the smaller targets and meet the requirement of real-time detection.

Abstract: The Single Shot MultiBox Detector (SSD) is one of the fastest algorithms in the current target detection field. It has achieved good results in target detection but there are problems such as poor extraction of features in shallow layers and loss of features in deep layers. In this paper, we propose an accurate and efficient target detection method, named Single Shot Object Detection with Feature Enhancement and Fusion (FFESSD), which is to enhance and exploit the shallow and deep features in the feature pyramid structure of the SSD algorithm. To achieve it we introduced the Feature Fusion Module and two Feature Enhancement Modules, and integrated them into the conventional structure of the SSD. Experimental results on the PASCAL VOC 2007 dataset demonstrated that FFESSD achieved 79.1% mean average precision (mAP) at the speed of 54.3 frame per second (FPS) with the input size 300 × 300, while FFESSD with a 512 × 512 sized input achieved 81.8% mAP at 30.2 FPS. The proposed network shows state-of-the-art mAP, which is better than the conventional SSD, Deconvolutional Single Shot Detector (DSSD), Feature-Fusion SSD (FSSD), and other advanced detectors. On extended experiment, the performance of FFESSD in fuzzy target detection was better than the conventional SSD.

Keywords: target detection; feature enhancement; feature fusion; real-time object detection; deep convolutional neural network

1. Introduction

Target detection is one of the main tasks of computer vision, and it is extensively used in areas such as driverless cars, face recognition, road detection, medical image processing, and human–computer interaction. The traditional target detection methods such as Local Binary Patterns (LBP) [1], Scale Invariant Feature Transforms (SIFT) [2], Histograms of Oriented Gradient (HOG) [3], and Haar-like (Haar) [4], are based on hand-crafted features. This feature extracted by the traditional target detection methods has obvious limitations. Firstly, the feature extraction is complex and the calculation speed is slow. Secondly, the artificial features largely limit the application scenarios of the algorithm. It is difficult to satisfy the needs of real-time detection on a complex and large dataset.

In recent years, a lot of target detection algorithms based on the convolutional neural network (CNN) have been proposed to solve the problem of poor accuracy and real-time performance of commonly used traditional target detection algorithms. Target detection algorithms based on convolutional neural networks have been divided into two categories according to the number of feature layers extracted from different scales. The first is the single scale characteristic detector type, such as region with CNN feature (R-CNN) [5], Fast Region-based Convolutional Network method (Fast R-CNN) [6], Faster R-CNN [7], Spatial Pyramid Pooling Networks (SPP-NET) [8], and You Only Look Once (YOLO) [9], and the other is the multi-scale characteristic detector type such as Single Shot Multibox Detector (SSD) [10], Deconvolutional Single Shot Detector (DSSD) [11], Feature Pyramid Networks (FPN) [12], and Feature-Fusion SSD (FSSD) [13]. The former type detects targets of different sizes under a single scale feature, which is a limitation to detection of targets that are too large or too small; the latter type extracts features from different scale feature layers for target classification and location, which improves the detection effect.

Among various target detection methods, SSD is relatively fast and accurate because it uses multiple convolution layers of different scales for target detection. SSD takes the Visual Geometry Group(VGG16) [14] as the basic network, and adopts a pyramid structure feature layer group (multi-scale feature layer) for classification and positioning. It uses features extracted from shallow networks to detect smaller targets, and larger targets are detected by deeper networks features. However, SSD does not consider the relationships between the different layers so that semantic information in different layers is not taken full advantage of. It might cause the problem named "Box-in-Box" [15], which means that a single target is detected by two overlapping boxes. In addition, the feature semantic information extraction by shallow networks is less and might not have enough capability to detect small targets.

To solve these problems, we propose a Single Shot Object Detection with Feature Enhancement and Fusion (FFESSD) by adding the Feature Fusion Module and Feature Enhancement Module to the conventional SSD. Our network can achieve 81.8% mAP on the PASCAL VOC 2007 test [16].

The contributions of our work can be summarized as follows:

(1) To fuse feature information of each layer, we propose a new fusion mechanism referred to as the Feature Fusion Module.
(2) A pair of novel Feature Enhancement Modules are proposed to enhance the extraction of semantic information.
(3) We show that FFESSD achieves state-of-the-art results on the Pascal VOC at a real time processing speed, and the performance of FFESSD in fuzzy target detection is better than the conventional SSD.

2. Related Work

In recent years, various methods using CNN have been widely used in target detection, and many successful cases have been achieved in computer vision tasks such as image classification [14,17–19], target detection [5–13], semantic segmentation [20,21], and instance segmentation [22–25]. Among them, target detection is a basic research which has been widely used in various fields, and the academics have proposed various strategies to improve the performance of target detection. In the earlier works based on CNN, such as R-CNN [5], Fast R-CNN [6], Faster R-CNN [7], and SPP-NET [8], there was a tremendous improvement in performance compared with the traditional target detection techniques. Specifically, these approaches usually use the separate algorithms such as Selective Searches [26] and Edge Boxes [27], to generate a set of region proposals as samples, and then classify the samples based on a CNN, so these methods are also called the two-stage detectors. Thus, although these methods improve the detection accuracy, they are too slow to meet the requirements of real-time detection.

In order to solve the problem of slow detection speed, the first single-stage detector is proposed. Named YOLO (You Only Look Once), it divides input images into multiple grids and performs localization and classification for multiple targets on each grid [9]. However, YOLO only uses the top layer of the feature maps to detect targets of different sizes and there is a lack of low-level high-resolution information, which results in somewhat inaccurate detection of small targets.

To maintain real-time speeds with a higher precision accuracy, Liu et al. [10] proposed the SSD which not only uses the top layer of the feature maps, but also uses low-level feature maps with high-resolution information to detect small targets. Recently, various methods have attempted to improve the accuracy of SSD, especially for small targets. Deconvolutional Single Shot Detector(DSSD) [11] could obtain higher accuracy by applying deconvolution layers to the feature pyramid and using ResNet-101 [17] instead of VGG16 [14]. At the expense of speed, Rainbow-SSD (R-SSD) [15] proposed a method to make explicit the relationship between different layers by using pooling and deconvolution and it achieved higher accuracy than SSD.

According to the above analysis, we propose a new single-shot network model to achieve the fusion of contextual information of each layer. In addition, we designed the shallow feature enhancement module (SFE) and deep feature enhancement module (DFE) to enhance the extraction of semantic information of network features. The proposed FFESSD can maintain a high computational efficiency without sacrificing precision.

3. Proposed Method

In this section, we first review the structure of SSD that uses a single deep neural network to detect targets in images, which is the basis of the proposed method FFESSD, and then describe the architecture of the FFESSD, introduce our Feature Fusion Module and feature enhancement module and explain why our approach has such a good performance.

3.1. Architecture of SSD

This section firstly gives a brief review about the most widely used single-stage detector Single Shot MultiBox Detector (SSD) [10], which is the basis of the proposed method FFESSD.

As shown in Figure 1a, the Single Shot MultiBox Detector (SSD) is based on the reduce VGG16 [14] and an auxiliary structure is added to the end of the base network. It is noteworthy that the SSD takes advantage of multiple convolution layers for target detection, specifically, the Conv4_3 layer is used to detect smaller targets and the deeper layers are adopted for detecting targets of bigger size. However, the shallower layers lack the semantic information and each layer in the feature pyramid is used independently as an input to the classifier network, which results in poor detection accuracy of small targets and causes the problem that a single target is detected by two overlapping boxes. Hence, in this work, we attempt to enhance the relationship between each layer of the feature pyramid and enhance the feature for detecting small targets.

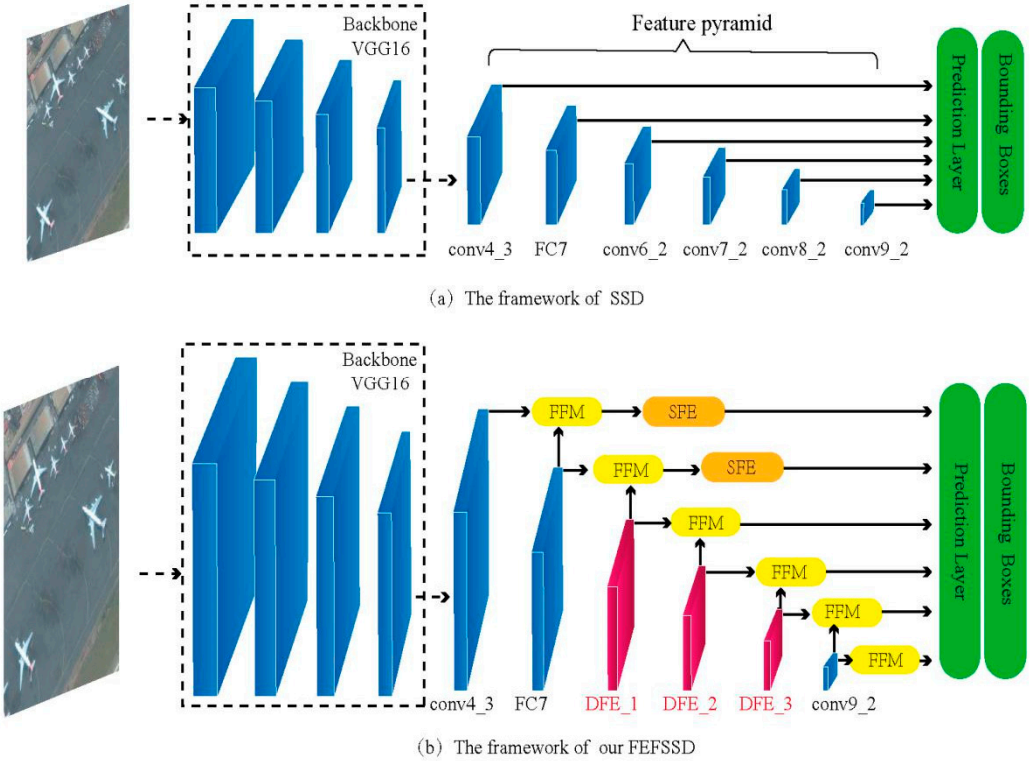


Figure 1. Networks of Single Shot MultiBox Detectors (SSD) and Single Shot Object Detection with Feature Enhancement and Fusion (FFESSD). (**a**) The framework of SSD and (**b**) the framework of FFESSD. The FFM is the Feature Fusion Module, the SFE is the Shallow Enhancement Module, and the DFE is the Deep Enhancement Module.

3.2. FFESSD

As illustrated in Figure 1b, the FFESSD is proposed based on the detection framework of the SSD. We used the Feature Fusion Module (FFM) to combine the feature map of each layer in the feature pyramid, which has different semantic information. Then we added the Shallow Enhancement Module (SFE) to enhance the semantic information of the shallower layers, in addition, we added Deep Enhancement Modules (DFE) to make the deep feature map have more detail about the input image. In the following sections, we will explain these core components in detail. These additional modules are simple, and can be easily combined with the conventional detection networks.

3.2.1. Feature Fusion Module

Instead of directly fusing the feature maps of each layer in the feature pyramid to the prediction module, we designed the Feature Fusion Module and use them in each level of the feature pyramid structure. The inner structure of FFM is shown in Figure 2. The 3-Way FFM is inserted in the first five layers of the feature pyramid structure and the 2-Way FFM is inserted in the Conv9_2.

To meet the contradictory requirements of maintaining low-level semantic information while having the flexibility to learn high-level abstraction, we use 3×3 group convolution followed by 1×1 convolution for learning more non-linear relations and widening the receptive field. In addition, branch 3 in the right side of Figure 2a contains a deconvolution layer whose input is the next layer's feature maps. Through the deconvolution layer, large context information is propagated to a feature

map of small scale, and small targets can be detected by using the information of their surroundings. Finally, the proposed Feature Fusion Module in Figure 2 can be expressed as Equation (1):

$$\begin{aligned} output &= \left\{\hat{x}_{1,2}, \hat{x}_{2,3}, \cdots, \hat{x}_{k-1,k}, \hat{x}_k\right\} \\ \hat{x}_{k-1,k} &= concat(B_1(x_{k-1}), B_2(x_{k-1}), B_3(x_k)) \\ \hat{x}_k &= concat(B_1(x_k), B_2(x_k)) \end{aligned} \tag{1}$$

where x_k is the kth-level feature map, B_1, B_2, and B_3 indicate branch 1, branch 2, and branch 3, respectively.

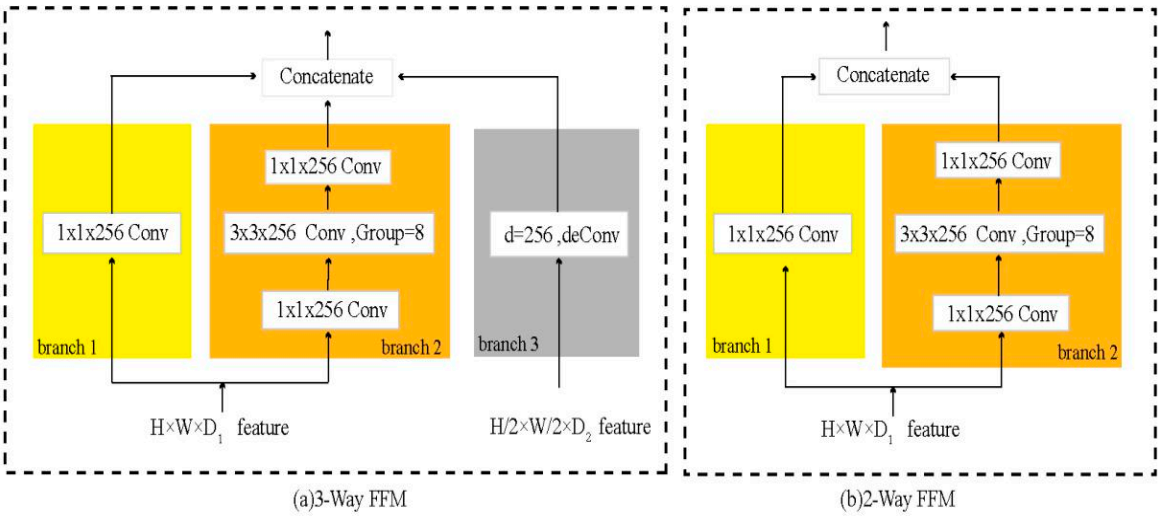


Figure 2. Feature Fusion Module, in which each conv box denotes a Conv + BN + ReLu processing. (**a**) 3-Way FFM is inserted in the Conv4_3, FC7, Conv6_2, Conv7_2, and Conv8_2 of the feature pyramid structure and (**b**) 2-Way FFM is inserted in the Conv9_2.

3.2.2. Feature Enhancement Module

In the original SSD, the depths of the earlier layers, such as Conv4_3 and FC7, used for small target objection are very shallow. Therefore, the reason for the poor detection of small targets is that the shallow layers do not have enough semantic information. Motivated by [13,19], we designed the shallow feature enhancement module for the shallower layers. Specifically, as shown in Figure 3a, we used the two similar branches to deepen the shallow layers. In the left branch, we used $k \times k$ group convolution followed by 1×1 convolution, which can broaden the receptive field and learn more non-linear relations. It's worth noting that we resolved the $k \times k$ convolutional layer into a $1 \times k$ and a $k \times 1$ convolutional layer for maintaining the receptive field as well as saving the calculation time of the SFE module. The only thing that differs in the other branch is the inversion of the $1 \times k$ and $k \times 1$ convolution layers.

In addition, the basic network of the SSD is followed by an auxiliary structure consisting of a series of convolutional layers, which forms a set of feature maps with progressively enhancing the extraction of semantic information and broadening the receptive field. For the auxiliary structure, we think it is not representative enough, and the deeper feature maps may ignore a lot of important details about the input image. In our implementation, we maintained the same cascade structure of SSD, and used the DFE module to replace the first few convolutional layers in the auxiliary structure. The inner structure of DFE module is shown in Figure 3b. We used the basic unit in the Dual Path Networks [20] as our DFE module, which combines the advantages of DenseNet [18] constantly exploring new

features and ResNet [19] implicitly reusing features, so more details of the input image can be obtained. The proposed Deep Feature Enhancement in Figure 3b can be expressed as Equation (2):

$$\begin{aligned} \mathrm{DFE}_{output} &= concat(\mathrm{Res}(x), \mathrm{Dense}(x)) \\ \mathrm{Res}(x) &= add(\lambda_1(x), \delta(x)) \\ \mathrm{Dense}(x) &= concat(\lambda_2(x), \delta(x)) \end{aligned} \tag{2}$$

where x is the feature maps input into the DFE module, $\lambda_1(\cdot)$ and $\lambda_2(\cdot)$ represent a BN + Relu + 1×1 convolutional layer operation, and $\delta(\cdot)$ is the main operation of deep Feature Fusion Module.

4. Experiments

In order to evaluate the detection performance of the proposed FFESSD, we trained the proposed method on the union of PASCAL VOC2007 trainval and PASCAL VOC2012 trainval [16], which includes 16,551 images with 40,058 targets and we evaluated the results on the PASCAL VOC2007 test, which contains 4952 images with 12,032 targets. In VOC 2007 and VOC 2012, a predicted bounding box is positive if its match with the ground truth is higher than a threshold (0.5). The backbone of FFESSD is the reduced VGG16 [14], which is pre-trained on the ILSVRC CLS-LOC [28] dataset. We conducted all the experiments on a machine with a single NVIDIA GeForce GTX 1080Ti graphics processing unit (GPU) and our code was based on PyTorch [29]. The metric to evaluate target detection accuracy was the mean Average Precision (mAP).

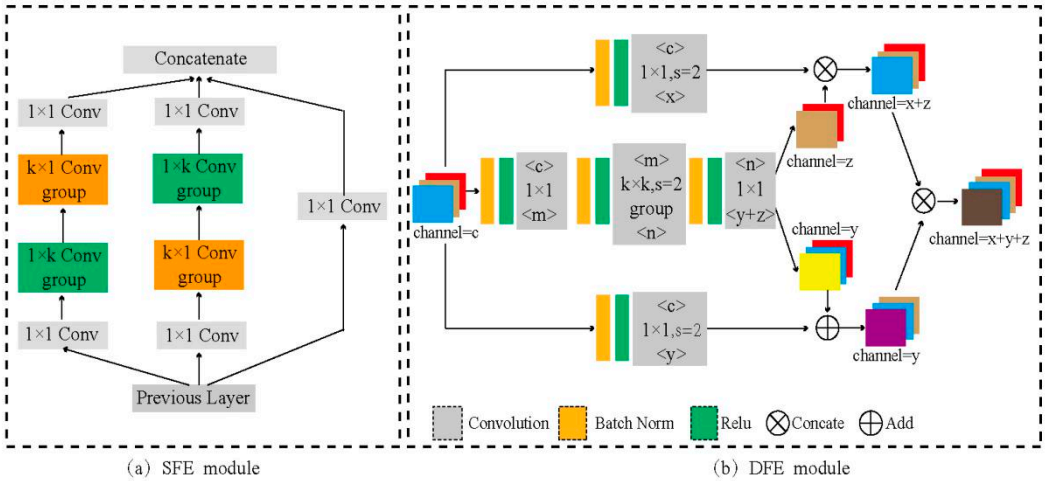


Figure 3. Feature Enhancement Module, in which each conv box denotes a Conv + BN + ReLu processing. (**a**) Shallow Feature Enhancement Module and (**b**) Deep Feature Enhancement Module.

4.1. Experimental Parameter Settings

In this experiment, we trained our model on PASCAL VOC 2007 trainval and VOC 2012 trainval. In order to obtain a more robust model for the size and shape of various input targets, data enhancement operations, such as random cropping and flipping of the input image should be carried out for each training image. For fair comparison, we used the same training settings as the SSD. For the model, when the input size was 300, the batch size was set to 32, and for a larger input size of 512, considering the memory limitations of GPU, we set the batch size to 12. We use a "warm-up" strategy to train our model. We set the learning rate at 10^{-3} to train for the first 80 k iterations, then continue training for 20 k iterations with a 10^{-4} learning rate, and finally we set the learning rate at 10^{-5} for another 20 k iterations. Following [10], we fine-tuned the entire model with a weight decay of 0.0005 and a momentum of 0.9.

4.2. Evaluation Metrics on Pascal VOC 2007 Test Set

The mean average precision (mAP) and the frames per second (FPS) commonly used in the field of target detection were used to compare the detection performance of different methods.

(1) Mean average precision (mAP): The mean AP represents the average value of all category AP, and the AP computes the average value of the precision over the interval from recall = 0 to recall = 1, which can be formulated as:

$$AP = \int_0^1 p(r)dr \tag{3}$$

where, p is the value of precision and r denotes the value of recall. The precision indicator can be seen as a measure of exactness or fidelity, and the recall indicator is a measure of completeness. The precision and recall indicators were formulated as follows:

$$precision = \frac{TP}{(\mathrm{TP} + \mathrm{FP})}, recall = \frac{TP}{TP + FN} \tag{4}$$

where, TP, FP, and FN represent the number of true-positive, false-positive, and false-negative respectively.

(2) Frames per second (FPS): In order to calculate the computing time of the proposed method and compare it with the existing techniques, we used the same method to calculate the detection efficiency of each method. Specifically, we set the batch size to 1, took the sum of the feature extraction time and the predicted time of 4952 images, and divided by 4952 to calculate the detection time of a single image.

4.3. Results on Pascal VOC 2007 Test Set

As shown in Table 1, we compared the proposed FFESSD with several other detectors, such as R-CNN, SSD, DSSD, FFSSD methods, on the PASCAL VOC 2007 test dataset. For the two-stage detectors, R-CNN [5] is the first target detection algorithm that combines region proposals with CNN. It generates about 2000 region proposals based on the selective search [26] method, which requires a large amount of space in memory. In addition, the normalization process of region proposals makes the algorithm lose a lot of features and it features semantic information, resulting in detection accuracy of 50.2% mAP and detection efficiency of 0.07 FPS. Fast R-CNN [6] and SPP-NET [8] also have the same significant drawbacks as R-CNN [5]. In order to reduce the operation time and improve the detection accuracy of the algorithm, Faster R-CNN [8] introduces a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. The mAP achieves 73.2% and the FPS achieves 7.0. HyperNet [30], Online hard example mining (OHEM) [31], and ION [32] also have problems of poor accuracy and real-time performance, making it difficult to satisfy the needs of real-time detection of complex large data sets. For the single-stage detectors, YOLO [9] is the first target detection algorithm to achieve real-time detection. It removes the selection of regional proposal scheme and makes the network structure simpler than the two-stage detectors. However, YOLO [9] only uses the top layer of the feature maps to detect targets of different sizes, which has great limitations on the size and location of the targets. The detection performance of mAP achieves only 63.4%, especially for small targets. In SSD [10], both low-level feature maps and top-level feature maps are used to improve the detection performance of small targets, but the low-level features have the shortcoming of insufficient semantic information. This may result in inaccurate detection of small targets. The mAP on the testing dataset achieves 77.2% with the input size 300 × 300 and 79.5% with the input size 512 × 512. Recently, various methods have attempted to improve the accuracy of SSD, especially for small targets, DSSD [11] uses ResNet-101 [17] instead of VGG16 [14] to achieve higher accuracy, when the input size is 300 × 300, the detection accuracy is improved by 1.4% compared with SSD, and the DSSD achieves 81.5% mAP with the input size 512 × 512, which is improved by 1.7% compared with SSD512. FSSD [13] inherits the network structure of SSD and introduces a lightweight and efficient Feature Fusion Module into it, which can improve the performance over SSD with just a little speed drop. R-SSD [15] improves the detection accuracy by

using the pooling operation and deconvolution operation to enhance the correlation between different feature layers in the feature pyramid structure of SSD. ESSD [33] uses two-way transfer of feature information and feature fusion to enhance the network and proposes a visual reasoning method that utilized fully the relationships between targets to assist further with target detection. DOSD [34] and feature-fused SSD [35] have improved the detection accuracy by combining feature maps with other layers.

Table 1. Comparisons of speed and accuracy based on PASCAL VOC2007 tests. (For fair comparison, as introduced in Section 3, we have improved on the basis of SSD and only improved the network structure).

Method	Backbone	Input Size	GPU	SPEED (FPS)	mAP (%)
R-CNN [5]	AlexNet	1000 × 600	-	0.07	50.2
SPP-NET [8]	AlexNet	224 × 224	-	0.5	63.1
Fast R-CNN [6]	VGG16	1000 × 600	Titan X	0.5	70.0
Faster R-CNN [7]	VGG16	1000 × 600	Titan X	7.0	73.2
HyperNet [30]	VGG16	1000 × 600	Titan X	0.9	76.3
OHEM [31]	VGG16	1000 × 600	Titan X	7.0	74.6
ION [32]	VGG16	1000 × 600	Titan X	1.3	76.5
YOLO [9]	GoogleNet	448 × 448	Titan X	45.0	63.4
SSD300 [10]	VGG16	300 × 300	1080Ti	71.0	77.2
DSSD321 [11]	ResNet-101 [17]	321 × 321	Titan X	9.5	78.6
FSSD300 [13]	VGG16	300 × 300	1080Ti	65.8	78.8
R-SSD300 [15]	VGG16	300 × 300	Titan X	35.0	78.5
ESSD300 [33]	VGG16	300 × 300	-	54	78.7
DOSD300 [34]	DenseNet [18]	300 × 300	Titan X	17.4	77.7
Feature-fused SSD [35]	VGG16	300 × 300	Titan X	43.0	78.9
FFESSD300 (ours)	VGG16	300 × 300	1080Ti	54.3	79.1
SSD512 [10]	VGG16	512 × 512	Titan X	19.0	79.8
DSSD513 [11]	ResNet-101 [17]	513 × 513	Titan X	5.5	81.5
FSSD512 [13]	VGG16	512 × 512	1080Ti	35.7	80.9
R-SSD512 [15]	VGG16	512 × 512	Titan X	16.6	80.8
ESSD512 [33]	VGG16	512 × 512	-	20.5	81.7
FFESSD512 (ours)	VGG16	512 × 512	1080Ti	30.2	81.8

In FFESSD, we used the FFM module to fuse the feature maps of different scales in the feature pyramid, the SFE module to enhance the semantic information of the shallower layers, and the DFE module to make the deep feature maps have more details about the input image. It improved the detection accuracy and obtained a relatively fast reasoning speed, which is beneficial in practical applications. From Table 1, we know that we have achieved good results. For the 300 input model, the proposed FFESSD was 1.9% improvement in the accuracy with 79.1% mAP compared to the SSD [10], and for the 512 input model, the FFESSD achieved 81.8% mAP, which is 2.0% better than the SSD. The proposed FFESSD shows state-of-the-art mAP, which is better than the SSD and other advanced detectors, when the input size is 300 × 300, the detection accuracy was improved by 0.3% compared with FSSD [13], and the FFESSD results in mAP of 81.8% with the input size 512 × 512, which is improved by 0.9% compared with FSSD512. In addition, the proposed FFESSD showed a better performance than the existing methods, such as R-SSD [15] and ESSD [33], thus further demonstrating that our Feature Fusion Module and feature enhancement module are effective. Table 1 shows more details about the test results of the exiting methods on the PASCAL VOC2007 test set.

4.4. Detection Examples

In order to intuitively compare the performance of FFESSD and SSD, as shown in Figure 4, a certain number of images were randomly sampled from the PASCAL VOC2007 test set and the outcomes were

compared. As shown in Figure 4, in the upper three rows of images, we can see that SSD often detects a single object with various overlapping boxes. However, the detection results of the proposed FFESSD algorithm in the same picture did not show the Box-in-Box status. FFESSD uses the Feature Fusion Module (FFM) to combine the feature maps of each layer in the feature pyramid, so as to obtain more image context information that is conducive to improving the detection accuracy. In addition, as far as we know, PASCAL VOC2007 test set contains 567 small targets (area $<32 \times 32$ pixel) that we selected to better evaluate our model. Our FFESSD300, improved its mAP response by 2.8% compared to the conventional SSD. We visualized some of the results, and as shown in the lower three rows of Figure 4, it can be shown that SSD misses a few small targets, and we can clearly see that our FFESSD has a better detection effect on small targets than SSD algorithm, because the proposed feature enhancement module (SFE and DFE) can improve the ability to extract semantic information of small targets.

(a) SSD detection results (b) FFESSD detection results

Figure 4. Detection examples on the Pascal VOC 2007 test dataset with SSD300/FFESSD300 model. (**a**) The images are the results of the SSD. (**b**) The images are the results of the Single Shot Object Detection with Feature Enhancement and Fusion (FFESSD).

4.5. Ablation Study on Pascal VOC 2007

In this section, we set up different models and tested them on the PASCAL VOC 2007 test dataset to verify the impact of each module on the detection performance. The results are shown in Tables 2 and 3.

4.5.1. SSD with Feature Fusion Module

In Table 2, we verify the SSD with and without the Feature Fusion Module (FFM) for the detection performance. In terms of general target detection, the detection performance of the model under the condition of different input sizes achieved 78.8% and 81.1% after applying the Feature Fusion module to the SSD and the detection performance improved by 1.6% and 1.7% compared with the conventional SSD. Especially for samples with similar backgrounds and targets, the conventional SSD causes the problem that a single target is detected by two overlapping boxes because each layer in the conventional SSD is independent and cannot reflect appropriate contextual information from different layers. Using the Feature Fusion Module to fuse the local contextual information at different layers, it is possible to understand some complex scenes and detect the targets from the background.

Table 2. Test results of SSD and SSD + FFM.

Method	Backbone	mAP	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
SSD300	VGG16	77.2	78.8	85.3	75.7	71.5	49.1	85.7	86.4	87.8	60.6	82.7
SSD + FFM300	VGG16	78.8	83.2	85.8	78.2	73.8	50.4	87.6	88.1	88.9	63.5	84.4
SSD512	VGG16	79.5	84.8	85.1	81.5	73.0	57.8	87.8	88.3	87.4	63.5	85.4
SSD + FFM512	VGG16	81.2	85.3	87.6	82.6	74.6	59.0	88.9	88.8	89.3	64.8	85.6
Method	**Backbone**	**mAP**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**Tv**
SSD300	VGG16	77.2	76.5	84.9	86.7	84.0	79.2	51.3	77.5	78.7	86.7	76.2
SSD + FFM300	VGG16	78.8	78.8	86.1	88.3	85.2	80.1	52.1	77.3	78.9	87.5	77.3
SSD512	VGG16	79.5	73.2	86.2	86.7	83.9	82.5	55.6	81.7	79.0	86.6	81.0
SSD + FFM512	VGG16	81.2	77.2	87.5	88.9	87.5	83.7	57.1	83.9	82.2	87.7	81.8

4.5.2. SSD with Shallow Enhancement Module and Deep Enhancement Module

Table 3 shows the performance of the SSD with and without the shallow feature enhancement module (SFE) and deep feature enhancement module (DFE). As displayed in Table 3, the original SSD with the shallow feature enhancement module achieved a 78.7% mAP when the input size was 300×300. By simply replacing the first few convolution layers in the auxiliary structure of the original SSD with the deep feature enhancement module, we can see that the result achieved a 78.5% mAP. For a larger input size of 512, the original SSD with the SFE module achieved 81.4% mAP, and the result of the original SSD with the DFE module was 81.2% mAP, exceeding SSD by 1.9 and 1.7 points respectively, which indicates that the SFE module and DFE module are effective in detection.

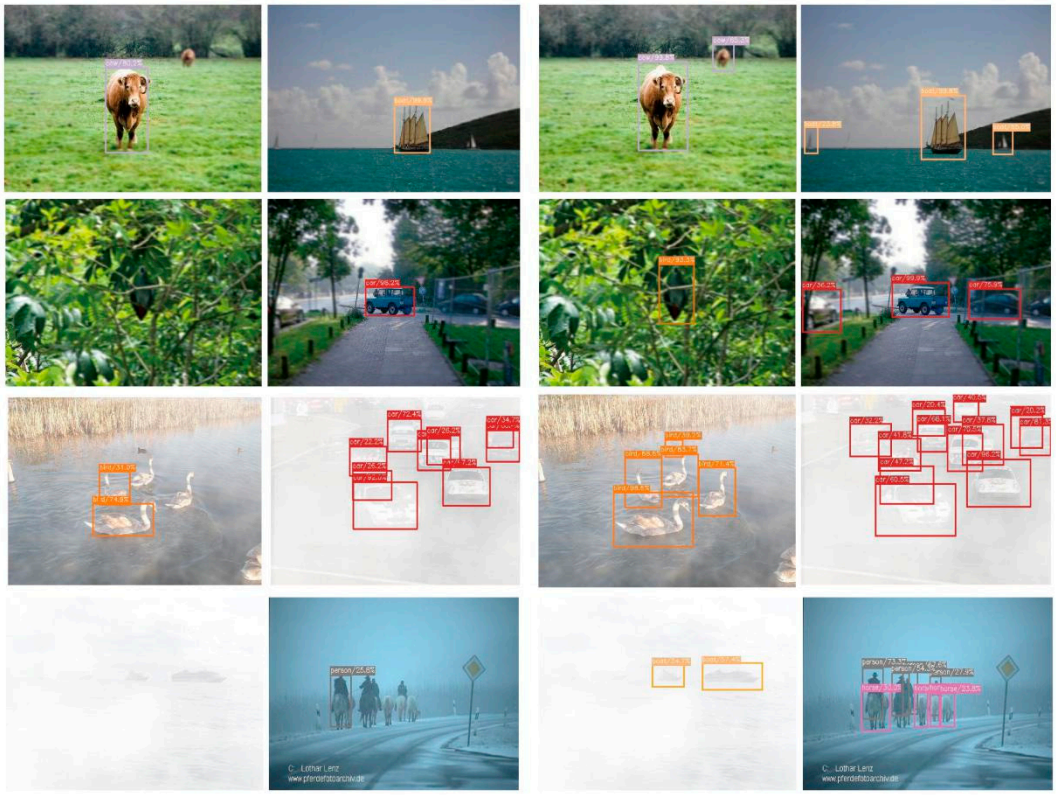
Table 3. Test results of SSD, SSD + SFE, and SSD + DFE.

Method	Backbone	mAP	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
SSD300	VGG16	77.2	78.8	85.3	75.7	71.5	49.1	85.7	86.4	87.8	60.6	82.7
SSD + SFE300	VGG16	78.7	84.1	86.4	77.6	73.0	50.6	87.2	88.3	88.4	62.9	84.0
SSD + DFE300	VGG16	78.5	80.2	85.9	75.9	72.8	48.5	87.1	88.4	88.1	62.5	83.7
SSD512	VGG16	79.5	84.8	85.1	81.5	73.0	57.8	87.8	88.3	87.4	63.5	85.4
SSD + SFE512	VGG16	81.4	86.5	88.0	82.6	74.2	59.1	88.3	88.9	89.4	66.1	85.7
SSD + DFE512	VGG16	81.2	86.1	87.4	81.7	73.8	57.6	88.1	89.1	89.3	65.7	85.3
Method	**Backbone**	**mAP**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**Tv**
SSD300	VGG16	77.2	76.5	84.9	86.7	84.0	79.2	51.3	77.5	78.7	86.7	76.2
SSD + SFE300	VGG16	78.7	78.6	85.3	88.6	85.1	79.8	52.6	77.6	78.6	87.1	77.5
SSD + DFE300	VGG16	78.5	77.5	85.1	89.2	84.7	79.4	54.6	77.4	80.3	87.8	78.3
SSD512	VGG16	79.5	73.2	86.2	86.7	83.9	82.5	55.6	81.7	79.0	86.6	80.0
SSD + SFE512	VGG16	81.4	77.3	87.4	89.3	86.8	83.5	57.9	84.2	82.1	87.2	81.8
SSD + DFE512	VGG16	81.2	76.8	87.0	90.1	86.2	83.2	60.2	84.1	82.6	88.0	82.4

4.6. Fuzzy Target Detection

In addition to comparing the performance of the algorithms in the standard train and test dataset, the FFESSD and SSD algorithms were used to detect the fuzzy images caused by focusing and haze

problems. The detection results are shown in Figure 5. From the figure, we can clearly see that our proposed FFESSD is beneficial to the detection of fuzzy targets. This is because the Feature Fusion Module is added in FFESSD, which can more comprehensively understand the feature information from context information so that it can better distinguish between the background and the fuzzy targets, and determine the location of fuzzy targets through the contextual information.



(a) SSD detection results (b) FFESSD detection results

Figure 5. Fuzzy target detection examples on the Pascal VOC 2007 test dataset with the SSD300/FFESSD300 model. (**a**) The images are the results of the SSD and (**b**) the images are the results of the FFESSD.

5. Conclusions

In this paper, we propose an accurate and efficient one-stage target detection method, named FFESSD. The FFM module is introduced to improve the performance by fusing features of different layers. In addition, we utilized the SFE module and the DFE module to enhance semantic information of shallow features and detail information of deep features respectively. We validated FFESSD on the PASCAL VOC 2007 benchmark dataset. From the experimental results, the proposed FFESSD method enhances the ability to express features and achieves better results. The proposed FFESSD shows state-of-the-art mAP, which is better than the SSD, FSSD, ESSD, feature-fused SSD, and other advanced detectors. In addition, on extended experiment, the performance of FFESSD in fuzzy target detection was better than the conventional SSD.

In the future research, we will enhance our FFESSD with much deeper and stronger backbone networks, which may be more conducive to enhancing the effect of target detection, especially for small targets, and we will study how to improve the accuracy of small targets detection by fusing contextual information.

Author Contributions: W.S. contributed towards the algorithms and the analysis. As the supervisor of W.S., he proofread the paper several times and provided guidance throughout the whole preparation of the manuscript. S.B. and D.T. contributed towards the algorithms, the analysis, and the simulations and wrote the paper and critically revised the paper. All authors read and approved the final manuscript.

Funding: This work was supported by the Natural Science Foundation of Sichuan Education Department (No.15ZA0152). The New Generation of Artificial Intelligence Major Program in Sichuan Province (No.2018GZDZX0036). The Key Program of Sichuan Science and Technology Department (No.2018SZ0040).

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

References

1. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face Recognition with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041. [CrossRef] [PubMed]
2. Lowe, D.G. Distinctive Image Features from Scale-Invariant Key points. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
3. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
4. Lienhart, R.; Maydt, J. An extended set of Haar-like features for rapid object detection. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; pp. 900–903.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
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 Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 346–361.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the International Conference on Computer Vision & Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
11. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
12. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2016**, arXiv:1612.03144.
13. Li, Z.; Zhou, F. FSSD: Feature fusion single shot multibox detector. *arXiv* **2017**, arXiv:1712.00960.
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. Jeong, J.; Park, H.; Kwak, N. Enhancement of SSD by concatenating feature maps for object detection. *arXiv* **2017**, arXiv:1705.09587.
16. Everingham, M.; Eslami, S.M.A.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
19. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 Jun 2015; pp. 1–9.

20. Chen, Y.; Li, J.; Xiao, H.; Jin, X.; Yan, S.; Feng, J. Dual path networks. In Proceedings of the Conference and Workshop on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4470–4478.
21. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 343–3440.
22. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
23. Dai, J.; He, K.; Li, Y.; Ren, S.; Sun, J. Instance-sensitive fully convolutional networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 534–549.
24. Mou, L.; Xiao, X.Z. Vehicle Instance Segmentation From Aerial Image and Video Using a Multitask Learning Residual Fully Convolutional Network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6699–6711. [CrossRef]
25. Bai, M.; Urtasun, R. Deep watershed transform for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2858–2866.
26. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
27. Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 391–405.
28. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
29. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the NIPS 2017 Autodiff Workshop, Long Beach, CA, USA, 9 December 2017.
30. Kong, T.; Yao, A.; Chen, Y.; Sun, F. Hypernet: Towards accurate region proposal generation and joint object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 845–853.
31. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 761–769.
32. Bell, S.; Lawrence Zitnick, C.; Bala, K.; Girshick, R. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2874–2883.
33. Leng, J.; Liu, Y. An enhanced SSD with feature fusion and visual reasoning for object detection. *Neural Comput. Appl.* **2018**, *1*, 1–10. [CrossRef]
34. Shen, Z.; Liu, Z.; Li, J.; Jiang, Y.-G.; Chen, Y.; Xue, X. 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.
35. Cao, G.; Xie, X.; Yang, W.; Liao, Q.; Shi, G.; Wu, J. Feature-fused SSD: Fast detection for small objects. In Proceedings of the 9th International Conference on Graphic and Image Processing (ICGIP 2017), Qingdao, China, 14–16 October 2017; p. 106151E.