Article

SSD7-FFAM: A Real-Time Object Detection Network Friendly to Embedded Devices from Scratch

Qing Li , Yingcheng Lin and Wei He *

Chongqing Key Laboratory of Space Information Network and Intelligent Information Fusion, School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400030, China; liqing-cqu@cqu.edu.cn (Q.L.); linyang@cqu.edu.cn (Y.L.)

* Correspondence: hewei007@cqu.edu.cn

Abstract: The high requirements for computing and memory are the biggest challenges in deploying existing object detection networks to embedded devices. Living lightweight object detectors directly use lightweight neural network architectures such as MobileNet or ShuffleNet pre-trained on large-scale classification datasets, which results in poor network structure flexibility and is not suitable for some specific scenarios. In this paper, we propose a lightweight object detection network Single-Shot MultiBox Detector (SSD)7-Feature Fusion and Attention Mechanism (FFAM), which saves storage space and reduces the amount of calculation by reducing the number of convolutional layers. We offer a novel Feature Fusion and Attention Mechanism (FFAM) method to improve detection accuracy. Firstly, the FFAM method fuses high-level semantic information-rich feature maps with low-level feature maps to improve small objects' detection accuracy. The lightweight attention mechanism cascaded by channels and spatial attention modules is employed to enhance the target's contextual information and guide the network to focus on its easy-to-recognize features. The SSD7-FFAM achieves 83.7% mean Average Precision (mAP), 1.66 MB parameters, and 0.033 s average running time on the NWPU VHR-10 dataset. The results indicate that the proposed SSD7-FFAM is more suitable for deployment to embedded devices for real-time object detection.

Keywords: object detection; feature fusion; attention mechanism; lightweight network; embedded devices

Citation: Li, Q.; Lin, Y.; He, W. SSD7-FFAM: A Real-Time Object Detection Network Friendly to Embedded Devices from Scratch. *Appl. Sci.* **2021**, *11*, 1096. <https://doi.org/10.3390/app11031096>

Received: 29 December 2020

Accepted: 22 January 2021

Published: 25 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As one of the fundamental visual recognition problems of computer vision, object detection is the basis of many other computer vision tasks, such as instance segmentation [1,2] and object tracking [3]. Object detection must identify not only the object's categories but also locate each item. Object detection has been extensively studied in the literature. In recent years, benefited by the rapid development of Deep Convolutional Neural Networks (DCNN), object detectors based on deep learning have achieved significant breakthroughs. Object detection has been widely used in the real world, such as robot vision, video surveillance, and autonomous driving. In order to improve the detection accuracy, most research focuses on the design of increasingly complex object detectors such as R-CNN [1], Single-Shot MultiBox Detector (SSD) [4], You Only Look Once (YOLO) [5], and their variants [2,6–9]. Although they have achieved high detection accuracies, such object detection networks are usually challenging to handle for embedded devices due to computational and memory limitations. Therefore, the design and development of more efficient deep neural networks for real-time embedded object detection are highly expected.

Current state-of-the-art object detectors with deep learning can be mainly divided into two major categories: two-stage detectors [1,2,6,7] and one-stage detectors [4,5]. Two-stage detectors firstly generate a set of preselected boxes for the object to be detected and then perform object category prediction. Therefore, Two-stage detectors often reported state-of-the-art results on many public benchmark datasets. However, they are relatively slow.

One-stage detectors directly train convolutional neural network models to map image pixels to the coordinates of the bounding box, so they are much faster and more desired for real-time object detection applications. Despite the fact that one-stage detectors have achieved good results in the trade-off between accuracy and speed, it is not realistic to port them directly to embedded applications because the network models are often large.

Recently, research on lightweight object detection networks that can be applied to embedded devices has attracted more and more researchers. Different artificially designed lightweight neural network architectures have been utilized for object detection, such as the lightweight network architecture MobileNet proposed by Google [10,11], which uses deep separable convolution instead of standard convolution. ShuffleNet [12] is proposed through the point-by-point group convolution kernel channel shuffling technology by Face++. The Fire module introduced by Iandola et al. in SqueezeNet [13] consists of two parts, a compression layer (squeeze) and an expansion layer (expand), which reduces the amount of calculation required for the entire model by reducing the number of channels in the squeeze layer. The AF-SSD [14] applied MobileNetV2 [11] and extra convolutional layers with basic units in ShuffleNetV2 [15] and depthwise separable convolutions as the lightweight backbone. Experimental results show that AF-SSD is a fast and accurate detector with few parameters. Many other studies [16–24] have shown that object detectors using these lightweight networks as the backbone achieved state-of-the-art results. However, these lightweight networks need to be pre-trained on universal datasets such as ImageNet before using them as backbone networks for object detection. Pre-training is usually performed on the datasets of general image classification tasks, so it is difficult to transplant them to specific application scenarios like medical image detection. Simultaneously, these pre-trained network models have a large number of parameters and fixed structures, which makes it difficult to optimize the structure.

In order to solve the above problems, we propose a lightweight real-time object detection network SSD7-Feature Fusion and Attention Mechanism (FFAM), which is friendly to some different memory and computationally limited embedded platforms such as phone, robots, and drones and can be started from scratch. The contributions of this paper are summarized as follows:

1. We propose a seven-layer convolutional lightweight real-time object detection network, SSD7-FFAM, that can be started from scratch to solve the problems that arise when the structures of the existing lightweight object detectors with pre-trained network models as the backbone are fixed, difficult to optimize, and not suitable for specific scenarios.
2. A novel feature fusion and attention mechanism method is proposed to solve the problem of reduced detection accuracy caused by the decrease in the number of convolutional layers. It first combines high-level semantic information-rich feature maps with low-level feature maps to improve the detection accuracy of small targets. At the same time, it cascades the channel attention module and spatial attention module to enhance the contextual information of the target and guide the convolutional neural network to focus more on the easily identifiable features of the object.
3. Compared with existing state-of-the-art lightweight object detectors, the proposed SSD7-FFAM has fewer parameters and can be applied to various specific embedded real-time detection scenarios.

The remainder of this paper is organized as follows. The related work is described in Section 2. In Section 3, the details of the proposed SSD7-FFAM are stated. The experimental results and discussions are then reported in Section 4. Finally, Section 5 concludes this paper.

2. Related Work

2.1. Single-Stage Detectors

Different from the two-stage detection algorithm, the single-stage detector directly trains the convolutional neural network model to map image pixels to the coordinates of the

bounding box. In 2016, Redmon et al. proposed a real-time detector called You Only Look Once (YOLO) [5]. YOLO divides the image space into a fixed number of grids and then predicted each grid. YOLO achieves relatively good detection results at the time. However, YOLO only uses the last feature map to predict, which is not suitable for multi-scale object detection. Later, to solve the limitations of YOLO, Liu et al. proposed another single-stage object detector—Single-Shot MultiBox Detector (SSD) [4]. SSD and YOLO both use a convolution neural network for prediction. Still, SSD is different from YOLO in (1) using multi-scale feature map detection; the advantage of this is that a relatively large feature map is used to detect relatively small objects, and small feature maps are responsible for detecting large objects; (2) unlike YOLO, which uses full connectivity, SSD directly uses convolution for detection; (3) SSD draws on the concept of anchor in Faster R-CNN [6], and each cell sets prior boxes with different scales as the basis of the predicted bounding box, which reduces the difficulty of training. Some other single-stage detectors also be proposed recently, e.g., Single-shot Refinement [25], RetinaNet [26], and CornerNet [27].

SSD uses the adjusted VGG as the backbone network and then adds additional convolution layers to obtain more feature maps for detection. A basic SSD model is shown in Figure 1. SSD predicts objects on multiple feature maps, and each feature map predicts objects of different scales. In the prediction module, it is divided into two parts: positioning and classification. The redundant prediction frame is then filtered through an algorithm called non-maximum suppression to form the final prediction result. Using multiple feature maps for object detection, the detection accuracy of SSD is comparable to Faster R-CNN, and the speed is faster than it. However, because a pre-trained VGG or even deeper network is required as the backbone network, SSD is not ideal for embedded devices with limited memory. Meanwhile, it is not suitable for detecting small objects as each feature map is predicted separately.

Figure 1. The Single-Shot MultiBox Detector (SSD) model consisting of a VGG and additional convolution layers.

2.2. Deep Feature Fusion

Deep neural networks have a strong expressive ability by extracting deep feature maps. SSD uses the pyramid structure of the convolutional network to predict the target and achieves high-precision detection. A network called FPN [7], which combines high-level semantic information with low-level high-resolution feature information, is proposed to achieve higher accuracy. FPN is a fully convolutional network and its backbone uses ResNet. FPN uses nearest-neighbor interpolation to sample from top to bottom and then connects it horizontally with the feature map after 1×1 convolution in the way of element-wise summation. The experiments of applying FPN to RPN and Fast R-CNN [28] respectively show that after fusing different feature layers, the accuracy of object detection is improved, especially the detection of small objects.

To detect small objects quickly and accurately, a multi-level feature fusion method that introduces contextual information into the SSD is proposed in the article Feature-Fused SSD: Fast Detection for Small Objects [29]. Gao et al. designed two different feature fusion modules: the connection module and the element summation module. Compared with SSD, its detection accuracy and speed have been improved to a certain extent. However, Feature-Fused SSD is also not suitable for embedded devices because of the increased computational complexity over SSD.

2.3. Visual Attention Mechanism

In cognitive science, the visual attention mechanism is essential for the human visual system to have such amazing data processing capabilities. In computer vision, the main research problem is how to establish a suitable calculation model to explain this attention mechanism. Introducing the attention mechanism in computer vision information processing can not only allocate limited computing resources to important targets but also produce results that meet human visual cognition requirements. Therefore, the visual attention mechanism has become a research hotspot in the field of computer vision.

In recent years, most of the research work on the combination of deep learning and visual attention mechanism has focused on using masks to form attention mechanisms. The mask uses new weights to identify critical features in input pictures. Through training, the deep neural network learns the areas that need attention in each new picture and forms attention. This kind of attention pays close attention to the spatial domain [30] or the channel domain [31], which can be directly generated through the network after learning. Besides, this kind of attention is differentiable, and models can learn the weight of attention through neural network calculation of gradient and forward propagation and backward feedback [32].

Inspired by these excellent feature fusion and attention mechanism methods, we propose the SSD7-FFAM. The SSD7-FFAM is formed by adding the proposed FFAM based on SSD7 [33]. The details of SSD7-FFAM is introduced in the next section.

3. Proposed Method

In this section, each part of the proposed SSD7-FFAM in Figure 2 is illustrated in detail. First, we introduce the specific structure of the entire network in Section 3.1. Then, the feature fusion module is described in Section 3.2. Next, we illustrate the attention module combined with feature fusion in Section 3.3. Immediately afterward, Section 3.4 presents the prediction layers. Finally, in Section 3.5, the loss function during training is explained.

Figure 2. The specific structure of the proposed lightweight real-time object detection network SSD7-Feature Fusion and Attention Mechanism (FFAM) using a feature fusion and attention mechanism. Two feature maps from different convolutional layers firstly pass the feature fusion module to form a new feature map and then pass the attention module to generate a feature map for prediction and input to the prediction layers.

3.1. Specific Structure of SSD7-FFAM

Figure 2 depicts the specific structure of the proposed SSD7-FFAM. In SSD [4], the feature maps extracted by VGG and additional convolutional layers are respectively used for object location and classification. However, the initial shallow feature maps lack important semantic information, and this problem causes the detection accuracy to be inferior to the two-stage detectors. Therefore, SSD is not conducive to the detection of small objects. Unlike SSD7 [33], the proposed SSD7-FFAM adopts two novel modules: the feature fusion module and the attention module, based on SSD7, to compensate for the reduction of detection accuracy caused by the decrease in convolutional layers. The feature fusion module combines two feature maps of different scales into a new feature map after transformation. This module enhances the semantic information of shallow feature maps. The attention module is a lightweight module that combines channel attention and spatial attention. It dramatically improves network performance while bringing a small amount of calculation and parameters.

3.2. Feature Fusion Module

SSD7 [33] uses four feature maps of different sizes to predict objects independently, and there is no connection between each feature map. The deep low-resolution feature map has undergone many convolution operations and can extract rich semantic information, which helps distinguish the object and the background. However, due to excessive downsampling, a lot of detailed information is lost. The shallow, high-resolution feature map contains detailed information of the object, which is conducive to accurate object positioning. However, because it performs fewer convolution operations, the shallow feature map cannot extract enough high-level features, and the semantic information is insufficient. Therefore, combining shallow feature maps and deep feature maps to generate a feature map with high resolution and full semantic information can be beneficial to the detection of small targets.

The schematic diagram of the semantic interpolation processing proposed in FPN [7] is shown in Figure 3. The dark color indicates that it contains richer semantic information. The low-level feature maps are passed to high-level through convolution, and the high-level feature maps are given to the low-level through upsampling, and then the new feature maps generated by the element-sum module stitching between the two are used to predict the object.

Figure 3. The schematic diagram of the semantic interpolation processing, where $\oplus$ represents the element-wise sum operation.

Unlike FPN, in SSD7-FFAM, we use 2×2 deconvolutions instead of nearest-neighbor interpolation for upsampling. The deconvolution layer introduces non-linearity and therefore helps to represent the ability of network feature expression. In order to make the size of the feature map consistent, some deconvolutions are followed by zero-padding, which is to add pixels with a value of zero to the row or column. Similarly, the two adjusted feature maps are spliced by element-wise summation so that the in-depth semantic information is propagated to the shallow feature maps. An operation called Rectified Linear Unit (ReLU) follows the element-wise summation module. At the same time, the smooth convolution with the size of the 3×3 convolution kernel is used to reduce aliasing caused by the enlargement of the feature map size. Figure 4 shows an example of the structure of the feature fusion module used in the proposed SSD7-FFAM. The new Conv4 feature map is passed to the attention module along with the other two feature maps.

Figure 4. An example of the structure of the feature fusion module used in the proposed SSD7-FFAM.

3.3. Attention Module

3.3.1. Channel Attention Module

In order to effectively understand the scope of the object, many previous works [34,35] adopted the average pooling method. In this paper, SSD7-FFAM uses both maximum pooling and average pooling on the feature map after feature fusion so that the detector channel focuses on the object and its contextual information. The schematic diagram of the channel attention module is shown in Figure 5a. Firstly, we perform average pooling and maximum pooling on the feature map $F \in \mathbb{R}^{H \times W \times C}$ to obtain two $1 \times 1 \times C$ channel descriptors. Secondly, they are sent to a two-layer parameter-sharing neural network. The number of neurons in the two layers is C/r and C , where r is the reduction ratio, and the activation function is ReLU. Then, the weight coefficient M_c is obtained by adding the two features through a Sigmoid activation function. In summary, the calculation formula of channel attention is as follows:

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (1)$$

Figure 5. Schematic diagram of channel attention module and spatial attention module. (a) Channel attention module, where $\oplus$ represents summation operation. (b) Spatial attention module, which is given the output of the channel attention module as the input.

3.3.2. Spatial Attention Module

The authors of [36] point out that the merge operation in the channel dimension can highlight the feature map's information area. After the channel attention module, we introduce a spatial attention module to focus on where features are meaningful. Similar to channel attention, given the feature map F' after the channel attention module, the spatial attention module first performs the average pooling and maximum pooling of the channel dimensions to obtain two $H \times W \times 1$ channel descriptions and stitch these two descriptions together according to the channels. Then, the weight coefficient M_s is obtained after a 7×7 convolutional layer, and the activation function is Sigmoid. The spatial attention module is shown in Figure 5b and is computed as:

$$M_s(F') = \sigma(\text{Conv}_{7 \times 7}([\text{AvgPool}(F'), \text{MaxPool}(F')])) \quad (2)$$

Figure 6 shows the structure of the attention module used in SSD7-FFAM, where F' is the new feature obtained by multiplying M_c and the input feature, and F'' is the result of multiplying M_s by F' . The final feature map F_f is the result of adding F and F'' to the ReLU activation function.

$$F' = F \times M_c(F) \quad (3)$$

$$F'' = F' \times M_s(F') \quad (4)$$

$$F_f = \sigma(F + F'') \quad (5)$$

Figure 6. The structure of the attention module used by SSD7-FFAM.

3.4. Prediction Layers

SSD [4] draws on the anchor concept in Faster R-CNN [6] and sets prior boxes with different scales in each unit of the feature map. The predicted bounding boxes are based on these prior boxes, which can reduce the training difficulty to a certain extent. In SSD7-FFAM, we send the four feature maps after feature fusion and attention module to the prediction layer for object classification and location. Their scales are (37,37), (18,18), (9,9), and (4,4); the number of prior boxes set for each feature map is different. For the scale of the prior box, it follows the linear increasing rule: as the size of the feature map decreases, the scale of the prior box increases linearly:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), k \in [1, m] \quad (6)$$

where s_k represents the ratio of the prior box size to the feature map, $s_{\min}$ and $s_{\max}$ represent the minimum and maximum values of the ratio, and 0.2 and 0.9 are used in the paper, m refers to the number of feature maps.

Each prior box of each unit outputs a set of independent prediction values, which are mainly divided into two parts: the confidence of each category and the location of

the bounding box. In the prediction process, the category with the highest confidence is the category to which the bounding box belongs. In particular, when the first confidence value is the highest, it means that the bounding box does not contain the object because the background is also regarded as a particular category. The location of the bounding box contains four values (cx, cy, w, h) , which respectively represent the center coordinates and width and height of the bounding box.

For an $m \times n$ feature map, SSD7-FFAM sets k prior boxes in each unit, then each unit needs $(c + 4) \times k$ prediction values, so a total of $(c + 4) \times k \times m \times n$ predicted values, where c is the number of categories. Similarly, SSD7-FFAM uses convolution for prediction and needs $(c + 4) \times k$ convolution kernels to complete the detection process of this feature map.

3.5. Loss Function

In the training process, we firstly determine which prior box matches the ground truth, and the bounding box corresponding to the matching prior box is responsible for predicting it. If a prior box does not match any ground truth, then the prior box can only match the background, which is called a negative sample. Intersection over Union (IoU) is used in the matching process. It is the ratio of the area where the two boxes intersect to the combined area. The principle of matching has the following two points: First, we match each ground truth with the prior box with the highest IoU between it to ensure that each ground truth must match a certain prior box; second, for the remaining unmatched prior boxes, the prior box is also matched the ground truth if the IoU is higher than the threshold (0.5 in our experiments). The IoU of ground truth and the prior box can be formulated as follows:

$$IoU(gt_box, priorbox) = \frac{area(gt_box) \cap area(priorbox)}{area(gt_box) \cup area(priorbox)} \quad (7)$$

There are a lot more negative samples than positive samples because ground truths are less than prior boxes. SSD7-FFAM uses hard negative mining to sample negative samples and arrange them in descending order of confidence loss and selects top-k with larger loss as negative samples to ensure that the ratio of positive and negative samples is close to 1:3.

After the training samples are determined, the loss function needs to be calculated. SSD7-FFAM uses the weighted sum of location loss (loc) and confidence loss (conf) as the loss function:

$$L(x, c, l, g) = \frac{1}{N} \left(L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right) \quad (8)$$

where N is the number of positive samples and the weight parameter α is set to 1 through cross validation; $x_{ij}^p \in \{1, 0\}$ is an indicator parameter. When $x_{ij}^p = 1$, it means that the i -th prior box matches the j -th ground truth, the ground truth category is p ; c is the category confidence prediction value, l is the location prediction value, and g is the location parameter of the ground truth.

For the confidence loss, the proposed SSD7-FFAM uses SoftMax loss:

$$L_{conf}(x, c) = - \sum_{i \in Pos} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad (9)$$

$$\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (10)$$

For the location loss, SSD7-FFAM employs Smooth L1 loss, which is calculated as follows:

$$L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k smooth_{L1} \left(l_i^m - \hat{g}_j^m \right) \quad (11)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (12)$$

4. Experimental Results and Discussions

We conducted experiments on two widely used datasets: NWPU VHR-10 [37] and Pascal VOC [38]. First, we compare the proposed SSD7-FFAM with other state-of-the-art methods on the NWPU VHR-10 dataset containing many small targets. Second, we conducted ablation experiments on the Pascal VOC dataset to understand the role of each module in the proposed FFAM. Finally, we added the proposed FFAM to the original SSD300 to form SSD300-FFAM and carry out comparative experiments with other state-of-the-art detectors on the Pascal VOC dataset to verify the effectiveness of the FFAM. We adopted the standard mean Average Precision (mAP) to measure the object detection performance.

4.1. Datasets and Evaluation Metric

4.1.1. Datasets Description

The NWPU VHR-10 dataset is a 10-class geospatial object detection dataset. The dataset contains 3775 object instances, including 757 airplanes, 302 ships, 655 storage tanks, 390 baseball diamonds, 524 tennis courts, 159 basketball courts, 163 ground track fields, 224 harbors, 124 bridges, and 477 vehicles. These object instances are manually marked with horizontal borders. It contains a total of 800 images, and we only used 650 of them with annotated images in this experiment.

The Pascal VOC [38] dataset is a standardized dataset with good image quality and complete labels. It is widely used to evaluate object detection and instance segmentation algorithms. It contains 20 common objects, namely an airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, and TV Monitor. Each image in the dataset has a one-to-one correspondence with its annotation file (class label and bounding box). At the same time, there may be multiple different object categories in an image. The training set and test set were divided according to a ratio of approximately 1:1. Pascal VOC 2007 contains 9936 labeled images with 24,640 objects, of which about 5011 were used for training and validating, and the remaining 4925 images were used for testing. Pascal VOC 2012 contains 11,540 labeled images with 27,450 objects. In this experiment, all the labeled images in Pascal VOC 2012 were used for training and validating [39].

4.1.2. Evaluation Metric

Like most other object detection methods, mean Average Precision (mAP) was adopted as the evaluation metric. In object detection, mAP is the average of the APs in multiple categories. The AP depends on the precision and recall score and is calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (13)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (14)$$

$$AP = \int_0^1 \text{precision}(r) dr \quad (15)$$

$$mAP = \left(\frac{1}{N}\right) \sum_{i=1}^N AP_i \quad (16)$$

where N means the number of all the classes. AP is computed as the area under the precision-recall curve. FP indicates the false positives; TP indicates the true positives, i.e., $\text{IoU} > 0.5$; FN indicates the false negatives.

4.2. Experimental Details

The experimental platform is a PC with a 3.4 GHz CPU, 8.0 GB RAM, and Windows 10 operating system. The proposed method was implemented by Python 3.6 based on the Pytorch framework, accelerated by two NVIDIA GeForce GTX 2080Ti GPUs with 11 GB GPU memory, CUDA10.0, and cuDNN7.0.

The proposed SSD7-FFAM network was trained using Stochastic Gradient Descent (SGD) from scratch with a Gamma value of 0.1. The momentum parameter and weight decay were chosen as 0.9 and 0.0005. In the training process, the batch size we used was 32, which can be adjusted according to the capacity of the GPU used. Unlike SSDs that use backbone networks, all convolution layers of SSD7-FFAM were initialized with the “Xavier” method [40]. The max iteration was set to 120 k. The initial learning rate was set to 0.001 and decreases by 10% at 80 and 100 k, respectively. For the training procedure, making the training model and the test model have similar input sizes has been proven to be the best way to train a multi-scale object detection network [41]. In these experiments, all networks’ input image size was set to $300 \times 300 \times 3$. SSD7-FFAM uses the NMS with a Jaccard overlap of 0.45 per class and a confidence threshold of 0.01. We maintained the top 200 detections per image. To further limit the number of predictions to parse, we set top_k to 5 in the testing phase. Other experimental settings were the same as the original SSD.

4.3. Experimental Results

4.3.1. Results on NWPU VHR-10

In this experiment, we compared the proposed SSD7-FFAM with other state-of-the-art detectors on the NWPU VHR-10 dataset. The results are shown in Table 1. The proposed SSD7-FFAM achieved an mAP of 83.7%, which is a 13% improvement over the original SSD7 [33]. SSD7-FFAM has higher accuracy than the state-of-the-art two-stage methods RICNN [42], R-P-Faster R-CNN [43], and Faster R-CNN [6].

Table 1. The comparison result of the proposed SSD7-FFAM and state-of-the-art methods on the NWPU VHR-10 dataset.

Methods	mAP (%)	Average Running Time (s)
COPD [37]	54.6	1.070
RICNN [42]	72.6	8.770
R-P-Faster R-CNN [43]	76.5	0.150
Faster R-CNN [6]	80.9	0.430
SSD7	70.7	0.178
NEOON [44]	77.5	0.059
AF-SSD [14]	88.7	0.035
SSD7-FFAM	83.7	0.033

The SSD7-FFAM (0.033 s) also has the shortest average running time than other single-stage methods such as NEOON [44] and AF-SSD [14]. Although the mAP of AF-SSD is higher than that of SSD7-FFAM, the parameters of the SSD7-FFAM model only account for 29% of AF-SSD (1.66 vs. 5.7 MB). The results show that the proposed SSD7-FFAM is a lightweight real-time object detection network more suitable for embedded devices. Figure 7 shows some prediction results of the proposed SSD7-FFAM.

Figure 7. Qualitative detection examples on the NWPU VHR-10 dataset with the proposed SSD7-FFAM (83.7% mean Average Precision (mAP)). For each pair, the left (a,c,e) is the original image and right (b,d,f) is the result of the SSD7-FFAM. Each color corresponds to an object category in that image.

4.3.2. Ablation Study on Pascal VOC 2007

In order to test the effects of the feature fusion module and the attention mechanism module, we ran models with different modules added to the SSD7 on the Pascal VOC 2007 test set and discussed each method's results. All experimental settings were consistent, except for the network structure and certain components.

First, we tested the effect of the attention mechanism module. Table 2 shows that adding the attention mechanism to different networks can get different results. Comparing the original SSD7 (row 1) and adding the attention module on SSD7 (row 2), using the attention mechanism on SSD7 can get 37.6% mAP while SSD7 can only achieve 36.6%.

At the same time, we compared the number of parameters and find that the attention mechanism only increases by 0.01 MB on average. In the following experiments, we used the attention mechanism to obtain better detection results.

Table 2. Effects of the feature fusion module and the attention mechanism module on the SSD7. FF indicates feature fusion module; AM indicates attention mechanism module, Deconv indicates deconvolution module.

Methods	Feature Fusion	Deconv	Parameters (MB)	mAP (%)
SSD7 [33]	×	×	1.21	36.6
SSD7 + AM	×	×	1.22	37.6
FPSSD7 [45]	Ele-sum	×	1.54	41.7
SSD7 + FF	Ele-sum	✓	1.92	44.2
FPSSD7 + AM	Ele-sum	×	1.55	44.2
SSD7-FFAM	Ele-sum	✓	1.93	44.7

Then, we conducted comparative experiments to test the effects of the feature fusion module on our experimental platform. The results are also listed in Table 2. The SSD7-FFAM used the feature fusion module introduced in Section 3, while the FPSSD7 [45] used bilinear interpolation for upsampling during feature fusion. The mAP has an improvement of 7.6% using the proposed feature fusion method (row 4) compared to the method using FPSSD7 [45] is only 5.1% (row 3). The results demonstrate that using the feature fusion module with deconvolution contributes more to the detection accuracy than the bilinear interpolation. Compared with the FPSSD7 [45], the combination of FPSSD7 [45] and attention mechanism results in a higher mAP with 44.2% (row 5). The results show that the SSD7-FFAM (row 6) we proposed has an 8.1% improvement compared to the original SSD7.

Table 3 shows that the proposed SSD7-FFAM achieves the second-lowest average running time among the six different models, only 0.145 s slower than FPSSD7 [45], while the accuracy of SSD-FFAM is 3% higher. The average running time of SSD7-FFAM in comparison to other detectors is shown in Figure 8.

Table 3. The average running time of different networks on the Pascal VOC 2007 test set.

Methods	Average Running Time (s)
SSD7 [33]	0.988
SSD7 + AM	0.993
FPSSD7 [45]	0.595
SSD7 + FF	0.802
FPSSD7 + AM	0.902
SSD7-FFAM	0.740

Figure 8. Accuracy and average running time comparison of the proposed SSD7-FFAM with other methods on the Pascal VOC 2007 test set. SSD7-FFAM is more accurate compared with other models with little compromise on speed.

Therefore, the quantitative results in Tables 2 and 3 show that the FFAM has a good effect on improving detection accuracy and can more accurately detect various objects in the dataset.

4.3.3. Extended Research on SSD300

To further verify the effectiveness of the proposed FFAM method, in this experiment, we also added it to the original SSD [4], which is called SSD300-FFAM, and compared it with other state-of-the-art detectors on the Pascal VOC 2007 test set. Except for the different sizes of detection feature maps, other experimental settings are consistent with SSD300 [4].

We first compared the proposed SSD300-FFAM with other state-of-the-art detectors on the Pascal VOC 2007 test set. All detectors used VGG-16 as the backbone network, and their input image size was 300×300 . Since our methods are mainly modified based on SSD300 [4], we chose SSD300 [4] as the baseline method. The detection results are summarized in Table 4. The results show that SSD300-FFAM achieves 78.7% mAP, which has an improvement of 1.2% over the baseline method SSD300 and is 5.5% higher than the Faster R-CNN. SSD300-FFAM is slightly better than SSD-TSEFFM300 [46] (78.7% vs. 78.6%). Compared to other detectors, we also achieved much better performance. SSD300-FFAM shows a great improvement in testing tasks with specific backgrounds and some small objects, e.g., airplane (82.6%), chair (61.7%), motorbike (86.3%), and train (87.8%). This shows that SSD300-FFAM improves the detection effect of SSD on small objects to a certain extent. Therefore, the proposed feature fusion and attention mechanism method plays a useful role in object detection.

Table 4. Detection results of the proposed SSD300-FFAM and state-of-the-art detectors on the Pascal VOC 2007 test set. All models were trained with 07+12 (VOC 2007 trainval + VOC 2012 trainval). The entries with the best APs for each object category are bold-faced.

Methods	mAP (%)	Aero	Bike	Bird	Boat	Bottle	Bus	Car
R-CNN [47]	50.2	67.1	64.1	46.7	32.0	30.5	56.4	57.2
Fast R-CNN [28]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6
Faster R-CNN [6]	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7
ION [23]	76.5	79.2	79.2	77.4	69.8	55.7	85.2	84.2
MR-CNN [48]	78.2	80.3	84.1	78.5	70.8	68.5	88.0	85.9
SSD300 [4]	77.5	79.5	83.9	76.0	69.6	50.5	87.0	85.7
RON [21]	76.6	79.4	84.3	75.5	69.5	56.9	83.7	84.0
SSD-TSEFFM300 [46]	78.6	81.6	94.6	79.1	72.1	50.2	86.4	86.9
SSD300-FFAM	78.7	82.6	86.8	78.5	70.3	55.9	86.0	86.5
Methods	mAP (%)	Cat	Chair	Cow	Table	Dog	Horse	Mbike
R-CNN [47]	50.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9
Fast R-CNN [28]	70.0	86.7	42.8	78.8	68.9	84.7	82.0	76.6
Faster R-CNN [6]	73.2	86.4	52.0	81.9	65.7	84.8	84.6	77.5
ION [23]	76.5	89.8	57.5	78.5	73.8	87.8	85.9	81.3
MR-CNN [48]	78.2	87.8	60.3	85.2	73.7	87.2	86.5	85.0
SSD300 [4]	77.5	88.1	60.3	81.5	77.0	86.1	87.5	83.9
RON [21]	76.6	87.4	57.9	81.3	74.1	84.1	85.3	83.5
SSD-TSEFFM300 [46]	78.6	89.1	60.3	85.6	75.7	85.6	88.3	84.1
SSD300-FFAM	78.7	86.9	61.7	85.1	76.4	85.9	87.5	86.3
Methods	mAP (%)	Person	Plant	Sheep	Sofa	Train	Tv	
R-CNN [47]	50.2	53.6	26.7	56.5	38.1	52.8	50.2	
Fast R-CNN [28]	70.0	69.9	31.8	70.1	74.8	80.4	70.4	
Faster R-CNN [6]	73.2	76.7	38.8	73.6	73.9	83.0	72.6	
ION [23]	76.5	75.3	49.7	76.9	74.6	85.2	82.1	
MR-CNN [48]	78.2	76.4	48.5	76.3	75.5	85.0	81.0	
SSD300 [4]	77.5	79.4	52.3	77.9	79.5	87.6	76.8	
RON [21]	76.6	77.8	49.2	76.7	77.3	86.7	77.2	
SSD-TSEFFM300 [46]	78.6	79.6	54.6	82.1	80.2	87.1	79.0	
SSD300-FFAM	78.7	79.3	53.5	80.1	79.4	87.8	77.1	

We also compared the proposed SSD300-FFAM with state-of-the-art two-stage detectors and one-stage detectors using different backbone networks. For the results in Table 5, we discuss as follows:

Table 5. Comparison of the proposed SSD300-FFAM with state-of-the-art detectors on the Pascal VOC 2007 test set. All models were trained with VOC 2007 trainval and VOC 2012 trainval.

Methods	mAP (%)	#Boxes
Faster R-CNN(VGG-16)	73.2	300
R-FCN(ResNet-101)	79.5	300
SSD300(VGG-16)	77.5	8732
YOLOv3(DarkNet-53)	79.6	10,647
SSD300-FFAM(VGG-16)	78.7	9032

- The proposed SSD300-FFAM has 300 more boxes than the baseline method SSD300 because the sizes of detection feature maps used are 38, 20, 10, 6, 3, 1, respectively. However, its detection accuracy is higher.
- Two-stage methods like Faster R-CNN and R-FCN have a candidate frame extraction part, so the number of boxes is much less than that of one-stage methods YOLOv3 [9], SSD, or SSD300-FFAM.
- Compared with the one-stage method YOLOv3 [9], the deeper the backbone network used, the higher the object detector's detection accuracy.

4.4. Discussions

In the research of lightweight object detection network, previous works have achieved some remarkable success. With the rapid development of the Internet of Things (IoT), some efforts [49–51] have been made to reduce the need for storage space and computational complexity on edge devices by using the IoT and cloud-based services. These existing efforts send algorithms and databases to AWS services hosted in the cloud for saving storage space on the edge devices. However, these works also have the limitations of long data transmission time, slow data transmission speed, and high demand on the equipment's network condition. Compared with these works, the proposed SSD7-FFAM can overcome these limitations by directly deploying it to edge devices to realize real-time object detection. Similarly, SSD7-FFAM also has the limitation of insufficient processing power.

Aiming at realizing real-time object detection on embedded devices, we reduced the number of convolutional layers of the original SSD. A novel method of feature fusion and attention mechanism is proposed to improve the detection accuracy. The experimental results comparatively demonstrate the compactness of SSD7-FFAM.

In addition to the results given, it should be noted that FFAM is a universal scheme that provides a general solution for designing and implementing a fast and high detection accuracy object detector. Except for remote sensing image detection, SSD7-FFAM also has the potential to be applied in other fields, including but not limited to autonomous driving, military object detection, and medical image detection.

Although compared to other SSD-based lightweight object detection networks, our SSD7-FFAM can start from scratch. One of the limitations of SSD7-FFAM is its low accuracy on large-scale object detection datasets, which guides the direction of our future study.

SSD7-FFAM only pays attention to the feature extraction part, and the prediction part is not optimized. In future work, we will focus on the overall optimization of SSD7-FFAM.

5. Conclusions

In this paper, we propose a lightweight real-time object detection network SSD7-FFAM for embedded devices that can be used in specific scenarios from scratch. The proposed novel feature fusion and attention mechanism method can effectively improve the accuracy of object detection. Firstly, through deconvolution and feature fusion, low-level feature maps and high-level feature maps are combined to obtain feature maps with

more substantial semantic information for prediction, which is conducive to detecting small objects. Secondly, the channel attention module and the spatial attention module are used to enhance the contextual information of the detection target so that the detector focuses more on the object instead of the background. The experimental results on the NWPU VHR-10 dataset show that the proposed SSD7-FFAM can reach 83.7% mAP and predict an image at an average of 0.033 s. The proposed SSD7-FFAM has a parameter value of 1.66 M, which is easier to deploy to embedded devices than other lightweight object detection networks because it does not require a pre-trained model.

In our future work, in addition to addressing the above limitations, we will transplant the SSD7-FFAM to embedded devices such as drones, and we will test more methods and further optimize our model to enhance scalability.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization, and writing—original draft preparation, Q.L.; resources, writing—review and editing, supervision, project administration, and funding acquisition, Y.L. and W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2020YFC0832700.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. 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. (In English) [[CrossRef](#)]
2. He, K.M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988. (In English) [[CrossRef](#)]
3. Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; et al. T-CNN: Tubelets with Convolutional Neural Networks for Object Detection From Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 2896–2907. (In English) [[CrossRef](#)]
4. 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—Eccv 2016, Pt I, Amsterdam, The Netherlands, 11–14 October 2016; Volume 9905, pp. 21–37. (In English) [[CrossRef](#)]
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. (In English) [[CrossRef](#)]
6. Ren, S.Q.; He, K.M.; 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 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28. (In English)
7. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (Cvpr 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. (In English) [[CrossRef](#)]
8. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (Cvpr 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. (In English) [[CrossRef](#)]
9. Redmon, J.; Farhadi, A. YOLOv3: An Incremental improvement. *arXiv* **2018**, arXiv:1804.02767. (In English)
10. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for mobile vision applications. In Proceedings of the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Cvpr), Honolulu, HI, USA, 21–26 July 2017; pp. 432–445. (In English) [[CrossRef](#)]
11. Sandler, M.; Howard, A.; Zhu, M.L.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Cvpr), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520. (In English) [[CrossRef](#)]

12. Zhang, X.; Zhou, X.Y.; Lin, M.X.; Sun, R. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Cvpr), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856. (In English) [[CrossRef](#)]
13. Forrest, N.I.; Song, H.; Matthew, W.M.; Khalid, A.; William, J.D. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
14. Yin, R.; Zhao, W.; Fan, X.; Yin, Y. AF-SSD: An Accurate and Fast Single Shot Detector for High Spatial Remote Sensing Imagery. *Sensors* **2020**, *20*, 6530. [[CrossRef](#)] [[PubMed](#)]
15. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. Shufflenet V2: Practical guidelines for efficient cnn architecture design. In Proceedings of the 2018 European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 116–131.
16. Womg, A.; Shafiee, M.J.; Li, F.; Chwyl, B. Tiny SSD: A Tiny Single-Shot Detection Deep Convolutional Neural Network for Real-Time Embedded Object Detection. In Proceedings of the 2018 15th Conference on Computer and Robot Vision (CRV), Toronto, ON, Canada, 8–10 May 2018; pp. 95–101. [[CrossRef](#)]
17. Wang, R.J.; Li, X.; Ling, C.X. Pelee: A Real-Time Object Detection System on Mobile Devices. In Proceedings of the Advances in Neural Information Processing Systems 31 (Nips 2018), Montréal, QC, Canada, 3–8 December 2018; Volume 31. (In English)
18. Singh, B.; Davis, L.S. An Analysis of Scale Invariance in Object Detection—SNIP. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Cvpr), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3578–3587. (In English) [[CrossRef](#)]
19. Peng, C.; Xiao, T.; Li, Z.; Jiang, Y.; Zhang, X.; Jia, K.; Yu, G.; Sun, J. MegDet: A Large Mini-Batch Object Detector. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Cvpr), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6181–6189. (In English) [[CrossRef](#)]
20. Kong, T.; Yao, A.B.; Chen, Y.R.; Sun, F.C. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), Las Vegas, NV, USA, 27–30 June 2016; pp. 845–853. (In English) [[CrossRef](#)]
21. Kong, T.; Sun, F.C.; Yao, A.B.; Liu, H.P.; Lu, M.; Chen, Y.R. RON: Reverse Connection with Objectness Prior Networks for Object Detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (Cvpr 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 5244–5252. (In English) [[CrossRef](#)]
22. Bosquet, B.; Mucientes, M.; Brea, V.M. STDnet: Exploiting high resolution feature maps for small object detection. *Eng. Appl. Artif. Intell.* **2020**, *91*. (In English) [[CrossRef](#)]
23. Bell, S.; Zitnick, C.L.; Bala, K.; Girshick, R. Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), Las Vegas, NV, USA, 27–30 June 2016; pp. 2874–2883. (In English) [[CrossRef](#)]
24. Wu, B.C.; Iandola, F.; Jin, P.H.; Keutzer, K. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 446–454. (In English) [[CrossRef](#)]
25. Zhang, S.; Wen, L.Y.; Bian, X.; Lei, Z.; Li, S.Z. Single-Shot Refinement Neural Network for Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Cvpr), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4203–4212. (In English) [[CrossRef](#)]
26. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.M.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2999–3007. (In English) [[CrossRef](#)]
27. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *Int. J. Comput. Vis.* **2020**, *128*, 642–656. (In English) [[CrossRef](#)]
28. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. (In English) [[CrossRef](#)]
29. Cao, G.M.; Xie, X.M.; Yang, W.Z.; Liao, Q.; Shi, G.M.; Wu, J.J. Feature-Fused SSD: Fast Detection for Small Objects. In Proceedings of the Ninth International Conference on Graphic and Image Processing (ICGIP 2017), Qingdao, China, 13–15 October 2017; Volume 10615. (In English) [[CrossRef](#)]
30. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28. (In English)
31. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Cvpr), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141. (In English) [[CrossRef](#)]
32. Zhao, B.; Wu, X.; Feng, J.S.; Peng, Q.; Yan, S.C. Diversified Visual Attention Networks for Fine-Grained Object Classification. *IEEE Trans. Multimed.* **2017**, *19*, 1245–1256. (In English) [[CrossRef](#)]
33. Pierluigiferrari. keras_ssd7. Available online: https://github.com/pierluigiferrari/ssd_keras/blob/master/models/keras_ssd7.py (accessed on 3 May 2018).
34. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E.H. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. (In English) [[CrossRef](#)] [[PubMed](#)]
35. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929. (In English) [[CrossRef](#)]

36. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In Proceedings of the ICLR 2017, Toulon, France, 24–26 April 2017.
37. Cheng, G.; Han, J.; Zhou, P.; Guo, L. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm.* **2014**, *98*, 119–132. [[CrossRef](#)]
38. Everingham, M.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. (In English) [[CrossRef](#)]
39. Wang, J.; Hu, H.; Lu, X. ADN for object detection. *IET Comput. Vis.* **2020**, *14*, 65–72. [[CrossRef](#)]
40. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, Italy, 13–15 May 2010; pp. 249–256.
41. Qin, H.W.; Li, X.; Wang, Y.G.; Zhang, Y.B.; Dai, Q.H. Depth Estimation by Parameter Transfer with a Lightweight Model for Single Still Images. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 748–759. (In English) [[CrossRef](#)]
42. Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. an Remote Sens.* **2016**, *54*, 7405–7415. [[CrossRef](#)]
43. Han, X.; Zhong, Y.; Zhang, L. An Efficient and Robust Integrated Geospatial Object Detection Framework for High Spatial Resolution Remote Sensing Imagery. *Remote Sens.* **2017**, *9*, 666. [[CrossRef](#)]
44. Xie, W.; Qin, H.; Li, Y.; Wang, Z.; Lei, J. A Novel Effectively Optimized One-Stage Network for Object Detection in Remote Sensing Imagery. *Remote Sens.* **2019**, *11*, 1376. [[CrossRef](#)]
45. Yamashige, Y.; Aono, M. FPSSD7: Real-time Object Detection using 7 Layers of Convolution based on SSD. In Proceedings of the 2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Yogyakarta, Indonesia, 20–21 September 2019; pp. 1–6. [[CrossRef](#)]
46. Hwang, Y.J.; Lee, J.G.; Moon, U.C.; Park, H.H. SSD-TSEFFM: New SSD Using Trident Feature and Squeeze and Extraction Feature Fusion. *Sensors* **2020**, *20*, 3630. [[CrossRef](#)] [[PubMed](#)]
47. Ryu, J.; Kim, S. Chinese Character Boxes: Single Shot Detector Network for Chinese Character Detection. *Appl. Sci.* **2019**, *9*, 315. (In English) [[CrossRef](#)]
48. Gidaris, S.; Komodakis, N. Object detection via a multi-region & semantic segmentation-aware CNN model. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 1134–1142. (In English) [[CrossRef](#)]
49. Mehra, M.; Sahai, V.; Chowdhury, P.; Dsouza, E. Home Security System using IOT and AWS Cloud Services. In Proceedings of the 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 20–21 December 2019; pp. 1–6. [[CrossRef](#)]
50. Guillermo, M.; Billones, R.K.; Bandala, A.; Vicerra, R.R.; Sybingco, E.; Dadios, E.P.; Fillone, A. Implementation of Automated Annotation through Mask RCNN Object Detection model in CVAT using AWS EC2 Instance. In Proceedings of the 2020 IEEE Region 10 Conference (TENCON), Osaka, Japan, 16–19 November 2020; pp. 708–713. [[CrossRef](#)]
51. Seal, A.; Mukherjee, A. Real Time Accident Prediction and Related Congestion Control Using Spark Streaming in an AWS EMR cluster. In Proceedings of the 2019 SoutheastCon, Huntsville, AL, USA, 11–14 April 2019; pp. 1–7. [[CrossRef](#)]