symmetry

MDPI

Article

Reinforced Neighbour Feature Fusion Object Detection with Deep Learning

Ningwei Wang, Yaze Li and Hongzhe Liu *

Beijing Key Laboratory of Information Service Engineering, College of Robotics, Beijing Union University, Beijing 100101, China; 191083510919@buu.edu.cn (N.W.); 181083520411@buu.edu.cn (Y.L.)
* Correspondence: liuhongzhe@buu.edu.cn

Abstract: Neural networks have enabled state-of-the-art approaches to achieve incredible results on computer vision tasks such as object detection. However, previous works have tried to improve the performance in various object detection necks but have failed to extract features efficiently. To solve the insufficient features of objects, this work introduces some of the most advanced and representative network models based on the Faster R-CNN architecture, such as Libra R-CNN, Grid R-CNN, guided anchoring, and GRoIE. We observed the performance of Neighbour Feature Pyramid Network (NFPN) fusion, ResNet Region of Interest Feature Extraction (ResRoIE) and the Recursive Feature Pyramid (RFP) architecture at different scales of precision when these components were used in place of the corresponding original members in various networks obtained on the MS COCO dataset. Compared to the experimental results after replacing the neck and RoIE parts of these models with our Reinforced Neighbour Feature Fusion (RNFF) model, the average precision (AP) is increased by 3.2 percentage points concerning the performance of the baseline network.

Keywords: computer vision; object detection; feature extraction; region of interest; feature pyramid network

Citation: Wang, N.; Li, Y.; Liu, H. Reinforced Neighbour Feature Fusion Object Detection with Deep Learning. *Symmetry* **2021**, *13*, 1623. https://doi.org/10.3390/sym13091623

Academic Editor: José Carlos R. Alcantud

Received: 31 July 2021
Accepted: 30 August 2021
Published: 3 September 2021

1. Introduction

Target detection is an essential task in deep learning; it answers the question "what objects are located where". Traditional object detection algorithms mainly use artificially designed feature modeling to extract geometric information such as edges, colors, and textures and then detect them through support vector machines. The method has some obvious shortcomings. For example, the detection accuracy error is large in some complex scenes, such as significant alterations in the background and object scale or occlusion. With the advancement of deep learning, detection algorithms using convolutional neural networks have been gradually proposed, and the detection accuracy has been greatly improved. It has a potential impact on the development of the fundamentals of deep learning techniques, and it may help to reduce the amount of required labeled data in many deep learning tasks, such as recognition, instance segmentation, etc. [1]. Object detection has many applications in self-driving vehicles, medical image analysis, business analytics, and face identification. Object detection in transportation situations is still a challenging difficulty which is the key to supervising traffic order and maintaining road safety. The existing deep learning-based object detection algorithms are mainly divided into one-stage detection and two-stage detection. The one-stage object detection algorithm does not require a region of interest suggestion network, and the features extracted by the deep convolutional network are directly classified and the object position coordinate value, such as SSD [2], YOLOv1 [3], YOLOv2 [4], YOLOv3 [5], YOLOv4 [6], RetinaNet [7], CornerNet [8], CornerNet-Lite [9], CenterNet [10], FCOS [11], ExtremeNet [12], etc.

Since the one-stage detection network does not use candidate regions to generate the network, the scale is small, so the detection speed is faster than the two-stage network, and the accuracy is low. Based on one-stage target detection, Region-based Convolutional Neural Network (RCNN) [13] introduced region proposals. It uses a priori box to filter out the fields where objects could exist and use selective search means to merge these regions to generate candidate regions finally. Perform position and classification regression in the detector. Some R-CNN [13] frameworks, such as Libra R-CNN [14], Generic Region of Interest Extractor (GRoIE) [15], CBNet [16], ThunderNet [17], and CSPNet [18], fuse features from different levels to obtain one-level features that simultaneously include semantic information and location information. Some networks, such as Cascaded R-CNN [19], improve the average precision (AP) by extracting features many times, and guided anchoring [20] modifies the process of anchor frame generation to improve the AP. Feature Pyramid Network (FPN) [21] based on top-down, independently detects each layer of features and introduces Faster RCNN [22]. After FPN, there have been many feature fusion methods based on FPN, such as PANet [23], ThunderNet [17], Balanced FPN in Libra RCNN [14], BiFPN in NAS-FPN [24] and EfficientDet [25], etc. Nonetheless, the current algorithm does not completely solve the multi-scale problem, and there is still a loss of position and semantic information. In the top-down process, the background information gradients of small-scale features will generate enormous errors, thus exacerbating the scale imbalance in the feature fusion stage in the neck of the network. For the neck part, we report NFPN experiments conducted on LISA [26], and Table 1 shows the enhancement of the AP and the advantages for objects of different scales. Then, we report experiments conducted to test the Recursive Feature Pyramid (RFP) and ResRoIE methods using the Faster R-CNN architecture proceeding the MS COCO [27], including comparisons with Faster R-CNN in Table 2 and several other processes using the Faster R-CNN architecture in Table 3.

Table 1. Module-wise ablation analysis on the LISA Traffic Sign Dataset.

Method	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Baseline	65.2	76.1	74.8	66.8	71.0	70.5
BFP	63.9	75.9	74.0	63.4	71.2	77.9
BiFPN	58.4	70.9	69.8	55.4	65.1	83.5
BiFPN × 2	49.2	59.9	58.4	49.1	56.6	78.5
NFPN (ours)	66.0	77.4	75.7	65.8	72.1	75.5
ResRoIE + NFPN (ours)	67.2	78.7	77.3	68.0	74.1	76.0
NFPN × 2 + ResRoIE (ours)	69.8	78.0	76.2	66.1	72.3	80.5
NFPN + RFP + ResRoIE (ours)	67.5	78.8	77.5	68.3	73.8	81.0

Table 2. Comparison with Faster R-CNN and PANet proceeding the MS COCO.

Method	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Baseline	37.4	58.1	40.4	21.2	41.0	48.1
PANet	37.5	58.6	40.8	21.5	41.0	48.6
NFPN (ours)	37.9	58.2	41.1	21.1	41.3	49.5
NFPN + ResRoIE (ours)	39.0	59.8	42.5	23.1	42.6	50.5
RNFF (ours)	39.3	60.0	42.4	22.9	42.2	50.6

Table 3. Comparison experiments involving the application of the RNFF method in combination with other networks.

Method	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Baseline	37.4	58.1	40.4	21.2	41.0	48.1
Libra R-CNN	38.7	59.9	42.0	22.5	41.1	48.7
Grid R-CNN	40.4	58.5	43.6	22.7	43.9	53.0
Guided anchoring	39.6	58.7	43.4	21.2	43.0	52.7
GRoIE	37.5	59.2	40.6	22.3	41.5	47.8
Libra R-CNN + RNFF (ours)	39.3	59.2	42.8	22.6	42.7	51.2
Grid R-CNN + RNFF (ours)	40.6	59.5	43.7	24.1	44.6	52.7
Guided anchoring + RNFF (ours)	40.5	59.3	44.1	23.1	44.2	54.0
GRoIE + RNFF (ours)	40.1	59.7	43.5	23.1	44.5	52.2

2. Related Work

The existing deep learning-based target detection algorithms are divided into one-stage detection and two-stage detection.

2.1. One-Stage Detection

Contemporary researches concentrate on developing object detectors from several aspects: Scale awareness, spatial awareness, and task awareness. YOLOv1 [3] takes the image to be detected as the input of the network and classifies and regresses the features in the output layer to obtain the prediction frame and category of the object. YOLOv2 [4] optimizes the speed and accuracy of the model based on YOLOv1 and expands to be able to detect 9000 categories at the same time, so it is also called YOLO9000. The YOLOv3 [5] model draws on the ideas of ResNet and extracts features based on the Darknet-53 backbone network. It achieves faster speed, and better performance than ResNet [28]. At the same time, compared to YOLOv2, it uses the FPN feature pyramid to optimize multi-scale object detection. YOLOv4 [6] object detection based on the CSP method balances both up and down and regards to small and large networks while sustaining optimal speed and precision. The most common model scaling technique is to change the depth (number of convolutional layers in a CNN) and width (number of convolutional filters in a convolutional layer) of the backbone and then train CNNs suitable for different devices [29].

Given that FPN makes network structure complex, brings memory burdens, and slows down the detectors, we offer a mild but highly efficient way without using FPN to address the optimization problem differently, denoted as YOLOF [30]. In this paper, the issue associated with nudity detection at various scales and backgrounds were addressed [31]. CornerNet [8] concludes that the advantage of anchor frames, particularly in one-stage detectors, has drawbacks, such as slowing down the training speed, and introducing additional hyperparameters. CenterNet [10] further improves CornerNet and detects each object by submitting another critical point as a triplet of crucial points in the proposed center. FCOS [11] uses the idea of semantic segmentation to resolve the difficulty, abandoning the standard anchor boxes and object proposals in object detection, making it unnecessary to tune the hyper-parameters involving anchor boxes and object proposals. ExtremeNet [12] turns the target detection problem into a simple appearance information-based key point predicting situation, thus cleverly avoiding region classification and specific feature learning. First, the extreme points can reflect the object information better than the bounding box, compared with the existing object detection model. Secondly, the author also proposed that a more detailed octagonal segmentation estimation result can be obtained by using a simple trick. Finally, if you are not satisfied, you can use it in combination with Deep Extreme Cut [32] to convert extreme points into segmentation masks. An algorithm based on a two-stage target detection network is proposed to realize the classification and detection of people, vehicles, pets, etc., to achieve the detection of objects far away and surrounding [33].

2.2. Two-Stage Detection

Fast RCNN abandons RCNN's method of extracting features for each suggested region and introduces the RoI pooling algorithm to select features of the entire image. The purpose is to resolve the time-consuming problem of RCNN repeatedly calculating the features of each candidate region. Faster RCNN proposes a Region Proposal Network (RPN) to generate candidate regions, which greatly improves generating candidate regions. The possible feature extraction methods in the backbone part mainly include ResNet [28], ResNeXt [34], Res2Net [35], and HRNet [36], while the feature fusion in the neck part integrates the output features from each level of the backbone. The feature extraction method of SSD [2] is shown in Figure 1a. SSD is a typical method that uses multi-scale features without fusion. SSD uses features of different resolutions for detection to avoid the exponential drop in resolution caused by the CNN layer's deeper. FPN in Figure 2 introduces a top-down fusion method in the feature fusion part, which significantly promotes large objects with deep features. However, the location information is lost due to the reduced resolution. While the position information is kept sufficient at the time of the object, the semantic information is not extracted. PANet shown in Figure 1b uses a bottom-up approach to further feature fusion after FPN. Based on FPN, the position information of the shallow layer is propagated to the deep layer. However, there are similar problems with FPN. The error caused by FPN upsampling will continue to propagate in the bottom-up process with the downsampling of PANet, and even amplify the error. Various features nearby to each other might be picked; non-maximal suppression will be brought out after detection to remove those not-so-significant feature points [37]. ZigZagNet [38] improves on PANet so that in the top-down and bottom-up fusion process, information interaction between each layer of features is also carried out so that the multi-scale context information in both directions is enhanced. ThunderNet [17] simplifies FPN and proposes a Context Enhancement Module (CEM) module. CEM uses global pooling to pool FPN features and then scale them to three-layer features for detection. Libra RCNN scales the feature to the C_4 level and then restores it. The feature fusion part is named the Balanced FPN (BFP) model in shown in Figure 1d. The bidirectional FPN (BiFPN) Figure 1c [25] offers double-way fusion methods. In the feature fusion part, CSPNet proposes an Exact Fusion Model (EFM) in Figure 1f, which can better aggregate feature pyramids. ThunderNet design a Global Fusion Model (GFM) in Figure 1e to compare with the proposed EFM. We connect the anchor-based and anchor-free branches with symmetric structures. Compared with a single unit, symmetry is applied to combine knowledge extracted from two components. Moreover, the parallel anchor-based branch and anchor-free branch run in symmetry to select the most desirable trait and anchor box [39].

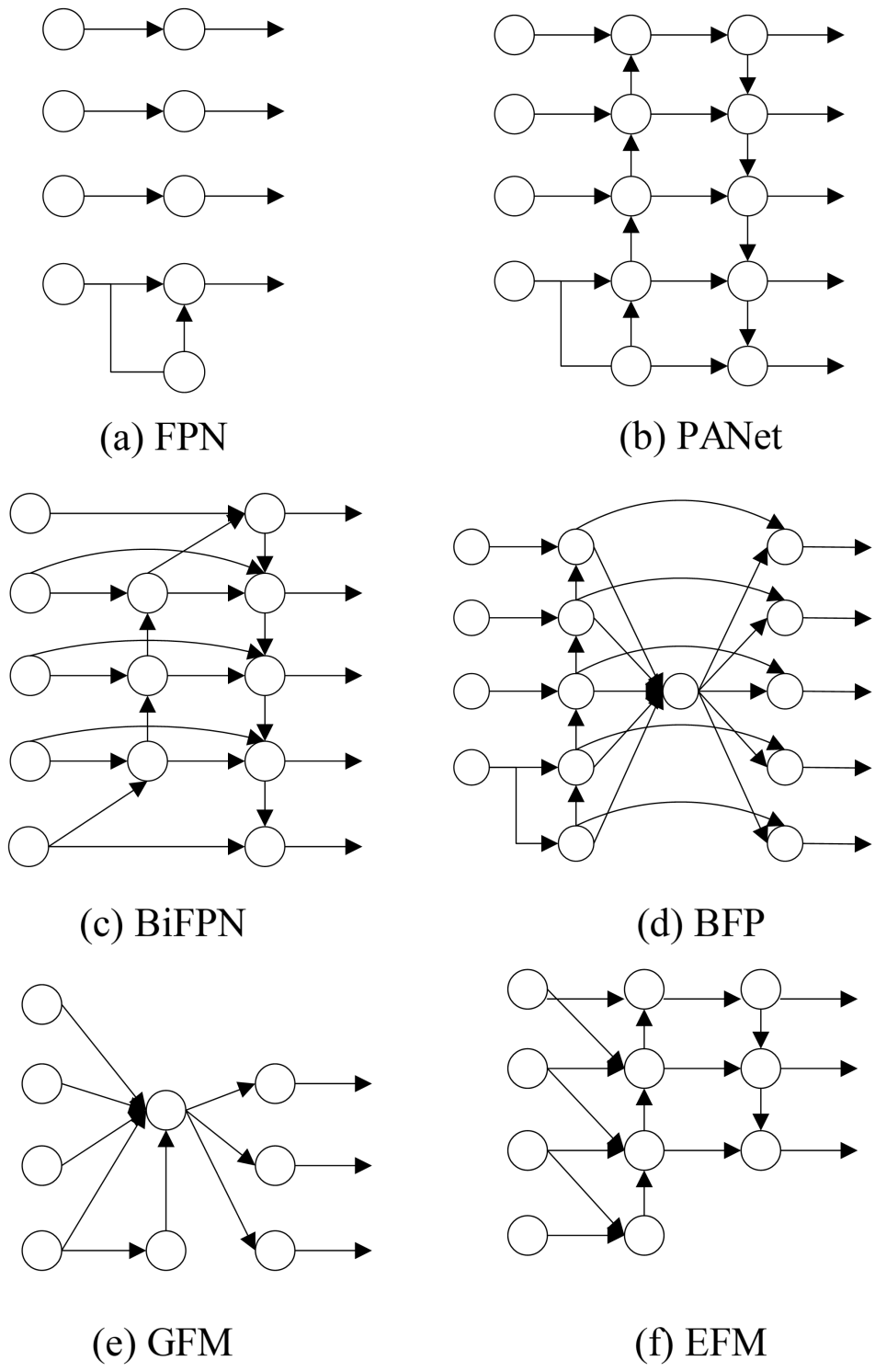

Figure 1. Panel (**a**) shows a single way fusion method; (**b**,**c**) show double way fusion methods; and (**d**–**f**) show other fusion methods.

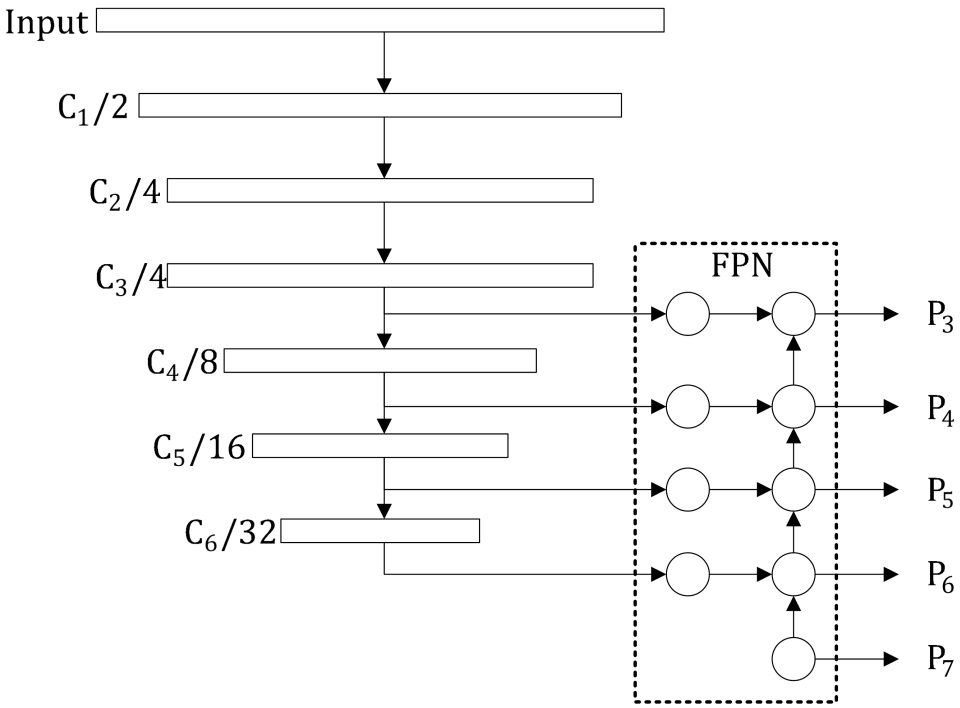

Figure 2. Backbone manuscript. and neck structures in Faster R-CNN. C_1 and C_2 are the output features of the stem layer, C_3–C_6 are the output features of the ResLayers, and P_3–P_7 are the output features of an FPN structure that includes an extra layer.

3. Reinforced Neighbour Feature Fusion (RNFF)

Figure 3 shows the overall flow of our model. Our goal is to reduce the feature imbalance caused by changes in the object scale. We design the following three methods to solve this problem. We first create a Neighbour Feature Pyramid Network (NFPN) architecture to verify the imbalance among different object scales at different feature levels for feature extraction. Then, we propose a Recursive Feature Pyramid (RFP) fusion method, which reduces the imbalance while integrating feature information from different layers. Finally, we offer ResNet Region of Interest Feature Extraction (ResRoIE) to reduce the mutual influence on the gradients caused by objects of different scales.

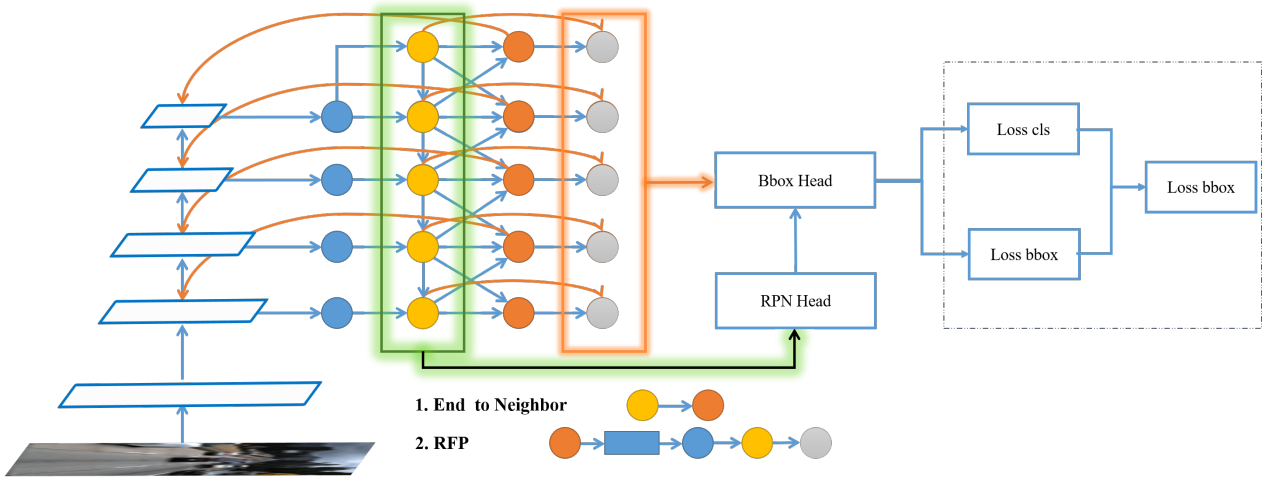

Figure 3. Overview of the proposed Reinforced Neighbour Feature Fusion (RNFF) network consists of three parts: Neighbour Feature Pyramid Network (NFPN) performs feature fusion from a neighbor. The orange lines represent recursive FPN connections Recursive Feature Pyramid (RFP). Features are returned to the backbone, extracted, and then added to ResNet Region of Interest Feature Extraction (ResRoIE).

3.1. Neighbour Feature Pyramid Network (NFPN)

For the neck part of the network, we design the NFPN architecture Figure 4 to verify the deterioration in the AP caused by small objects and the improvement caused by large objects after multiple up-sampling operations for objects of different sizes between different layers. NFPN is to fuse the features of two adjacent levels. The lower-level features are up-sampled or interpolated and then convolved to optimize the interpolated features. The upper-level features use convolution with a step size of 2 after pooling. Then use the convolution with a step size of 1, add the features of this layer for fusion, and add the combined characteristics to the initial input features to conquer the small object knowledge loss caused by upsampling. In a word, make full use of the feature to integrate it into the next layer better to reduce feature loss and interference. In the algorithm below, P_i denotes the output feature from the i-th layer of an FPN structure Figure 4, O_i is the output from the i-th layer of NFPN. When it comes to P_{i+1} relative to O_i, it stands for up-sampling, the nearest impending interpolation of bilinear interpolation. As for P_{i-1}, the O_i process represents pooling, either average pooling, maximum pooling, or minimum pooling. The former part of O_3 is the up-sampling of P_4 plus P_3, and the two are subjected to a convolution to get a result. The latter part is the feature P_3 of this layer. The primary part of O_7 is the pooled P_6 plus P_7, and the two undergo a convolution operation. The following part is the feature P_7 of this layer. For O_4, O_5, and O_6 are also require two sections; the first part is the feature P_i of this layer, the second part is P_i plus the up-sampling, the convolution of P_{i+1}, and the convolution of P_{i-1} after pooling. After the process of fusion of adjacent layers, the specific algorithm is as follows.

$$O_3 = P_3 + conv(P_3 + Resize(P_4)) \tag{1}$$

$$O_4 = P_4 + conv(P_4 + conv(Resize(P_3)) + Resize(P_5)) \tag{2}$$

$$O_5 = P_5 + conv(P_5 + conv(Resize(P_4)) + Resize(P_6)) \tag{3}$$

$$O_6 = P_6 + conv(P_6 + conv(Resize(P_5)) + Resize(P_7)) \tag{4}$$

$$O_7 = P_7 + conv(P_7 + Resize(P_6)) \tag{5}$$

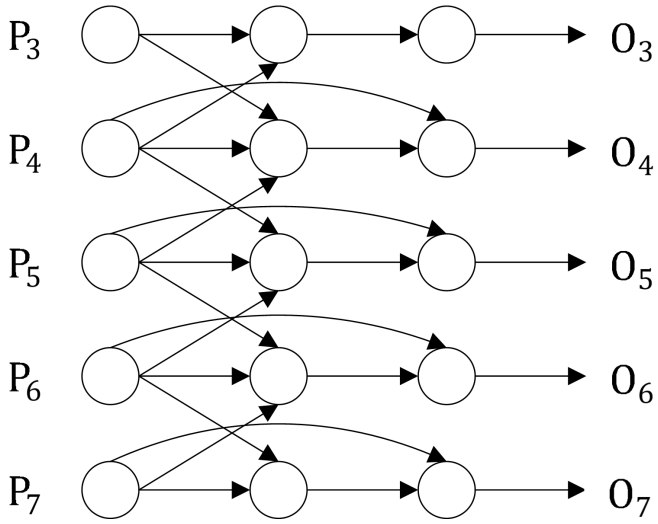

Figure 4. Neighbour Feature Pyramid Network (NFPN) architecture.

3.2. Resnet Region of Interest Feature Extraction (Resroie)

In the RoIE (RoI Extractor) part of the network in Figure 5, because only FPN output features are extracted, there will still be an imbalance among multi-scale objects. Due to the increase in the number of network layers of adjacent feature fusion, it impacts the gradient of the object with a small scale itself and produces the phenomenon of gradient

disappearance. Compared with Faster RCNN RPN50, the accuracy of large-scale objects in a row of BiFPN × 2 is lower because the fusion network is too extensive, which leads to a certain degree of gradient disappearance problem, which reduces the accuracy of objects at all scales. Nevertheless, the accuracy of its large-scale objects is still higher than Faster RCNN, which verifies the feature enhancement effect of the top-down process on large-scale objects and the feature weakening effect on small-scale objects. In the RoIE stage, map to the corresponding feature layer according to the region's suggested area, convolve the layer to obtain 7 × 7 features, and perform classification and regression. The features obtained in this way only contain the information obtained by one fusion method. To quickly and efficiently confirm this guess, we conducted experiments on the Laboratory for Intelligent, and Safe Automobiles Traffic Sign Dataset (LISA) [26], in which the resolution varies from 6 × 6 to 167 × 168. After the first ResNet [28] layer in the network backbone, the feature resolution has been reduced to 1/4. Large objects will have location information, whereas small objects, such as 6 × 6 objects, may have only weak semantic information because their remaining resolution will be only 1.5 × 1.5 if convolution is not considered. In addition, the resolution will be only 4 × 4 if 3 × 3 convolutions are used twice, and only 1 × 1 pixel will not be influenced by background pixels; that is, 15/16 pixels will be influenced by invalid pixels, which can cause incorrect gradients for deeper-level features when back-propagation is applied in top-down feature fusion. Due to the top-down fusion method, the fused features at shallow levels contain many invalid pixels, which the RoIE layer will also extract. Therefore, we extract the FPN features in our neck structure so that the gradient can bypass the top-down process to reduce the influence of invalid background pixels. For example, the features fused using the bottom-up method may be beneficial to large objects, but the accuracy of small objects will be lost. However, only using the top-down procedure to fuse information, it is impossible to use the two-way fusion to have a beneficial impact on the large-scale object. As in Figure 5, using the idea of shortcut in ResNet, the sum of FPN-out and NFPN-out is used as the output of the neck part to solve imbalance among multi-scale objects. After the adjacent features are merged in the neck stage, the FPN features are output simultaneously.

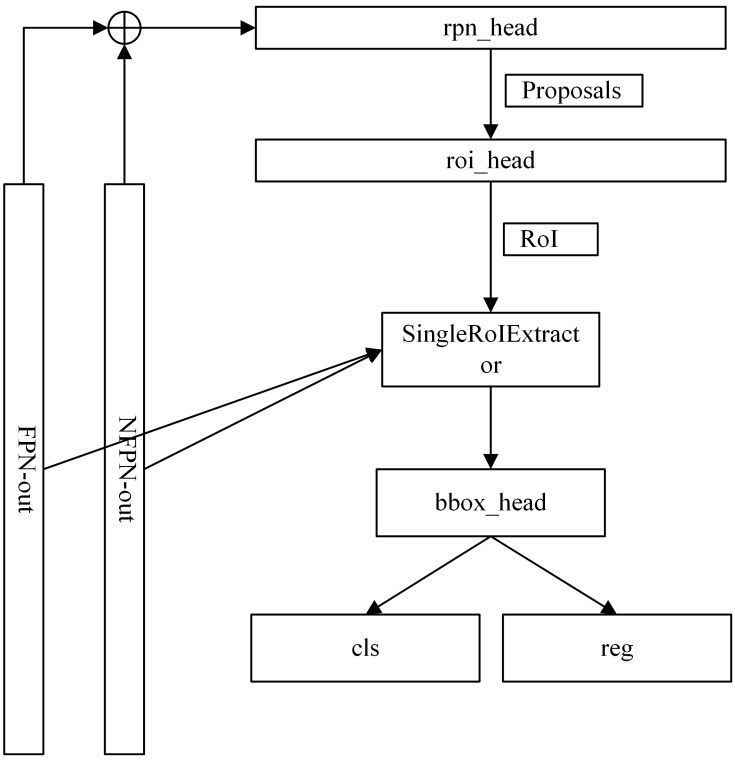

Figure 5. ResNet Region of Interest Feature Extraction (ResRoIE).

3.3. Recursive Feature Pyramid (RFP)

Because the fused features after the top-down operation are generated based on the currently existing pixels rather than the original pixels, if 3×3 convolutions are used, the influence of background pixels will still exist due to the enlargement of the receptive field. To address this issue, we refer to the idea applied in DetectoRS Figure 6 [39], namely, the idea of "thinking twice" for detection. It is built on the FPN, which combines additional feedback connections from the FPN to the bottom-up backbone layer. For the fused features, the backbone is used to select the parts again, as shown in Figure 7, and then the extracted features are fused with the features before extraction. After the backbone extracts the components, the parts before extraction are fused. After combining the features from each layer, we use a convolution block corresponding to the depth in the backbone to extort features again to reduce the impact of the background pixels after up-sampling and 3×3 convolutions. Then, the extracted features are summed with the output features from NFPN fusion. By merging additional feedback links of the FPN to the bottom-up backbone layer. RFP serves as the feature pyramid network, which takes level 3–7 features C_3, C_4, C_5, C_6, C_7 from the backbone network and recursively applies top-down and bottom-up bidirectional feature fusion. We demonstrated a method that improved object detection performance by building a more powerful robust feature extract module RNFF that recursively inputs the first extracted features in the neural network to remove the elements again. However, unlike DetectoRS, we do not need to build a new backbone and use the original image; instead, after the neighbor feature fusion step of the feature fusion process, the same backbone is used again for extraction to reduce parameters of the model.

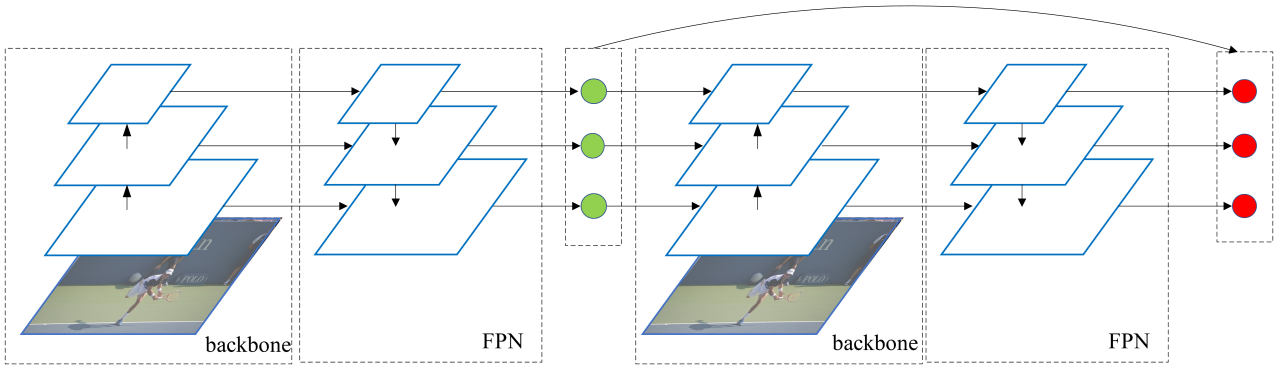

Figure 6. The Recursive Feature Pyramid (RFP) architecture in DetectoRS. In DetectoRS, a new backbone is created to extract the FPN output features.

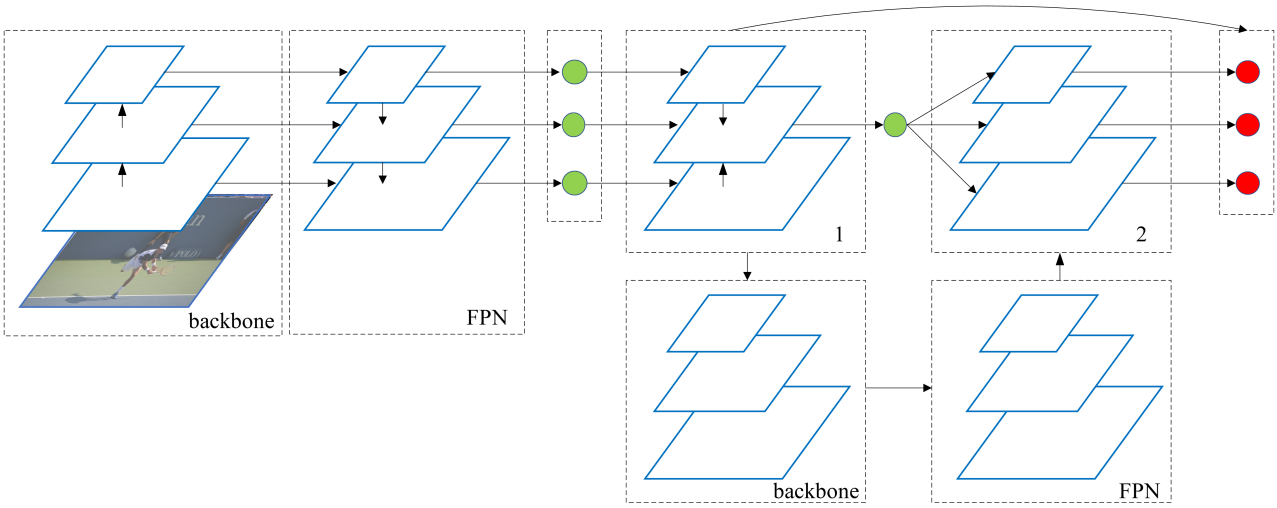

Figure 7. The RFP architecture in our model: 1 and 2 are the neighbor feature fusion method (NFPN). We extract the network features by reusing the backbone.

4. Experiments and Results

In this part, we report two groups concerning comparative experiments. The first group supports the effectiveness of addressing feature imbalance, using the NFPN architecture as the basis for comparison. The second group verifies the improvement achieved with our method. In the third group, the application of our Reinforced Neighbour Feature Fusion (RNFF) method in combination with other recent networks is investigated by replacing the original neck and RoIE parts of the networks. We use an auxiliary multi-scale feature enhancement module to assist in extracting multi-scale shallow features and merging them with the components selected from the backbone, which dramatically improves the expression ability of small objects [40].

4.1. Dataset

We conducted experiments on the LISA Traffic Sign Dataset for the NFPN architecture, and on the MS COCO dataset, [27] for the RNFF architecture. The LISA Traffic Sign Dataset contains 6k images divided into 47 traffic signal sign categories, and the MS COCO dataset contains more than 11k images of 80 classes.

4.2. Implementation Details

Our default experimental configuration was as follows unless otherwise specified. The experimental platform had 8 GPUs (TITAN V). The total number of epochs was 24, and we selected the best mAP among all epochs. The learning rate was 0.02, the weight decay rate was 0.0001, and the batch size on each GPU was 2. We used the Faster R-CNN architecture and ResNet-50 as the backbone. We used the COCO evaluation metrics in the MMDetection framework to evaluate and compare RNFF and the other methods. In the two-stage network experiment, taking Faster RCNN and its variant network as an example, after replacing the corresponding module in the model, whether it is the experiment with Res50 as the backbone network and the Res101 series as the backbone, the average accuracy and each, the scale accuracy has been improved. In the experiment with ResNet50, the mAP increase was the highest when compared with GRoIE, from 37.5% to 40.1%. The mAP has increased by 2.6 percentage points, and small, medium, and large objects have increased by 1.4, 1.2, and 4.4 percentage points, respectively.

4.3. Module-Wise Ablation Analysis

Toward the MS COCO dataset, we observed the act of NFPN fusion, ResRoIE, and the RFP architecture at different precision scales and the improvements in precision when these components were used in place of the corresponding original components in various networks. On the LISA Traffic Sign Dataset, we used FPN, BFP, and BiFPN to observe the influence of these different fusion methods on precision and used two stacked BiFPN modules to observe the influence of top-down fusion on precision.

In the NFPN architecture, we performed ablation experiments on two datasets. The LISA Traffic Sign Dataset experiments were performed to observe the impact of top-down and bottom-up fusion on objects at different scales and the improvement enabled by our NFPN architecture. We used the five feature layers generated by FPN for feature fusion. The fusion process is divided into three steps. First, the neighboring layers of the reference layer are resized to the exact resolution as the reference layer through pooling or interpolation. Then, a 3×3 convolution check is used to convolve the reconstructed features and add them to the reference layer features. Finally, a 1×1 convolution kernel is used for convolution. We used the sum (+) method to fuse two parts with the exact resolution. Through experimental data, it is found that the NFPN structure can improve mAP on the Lisa data set, but the detection effect on objects of different scales varies greatly. For the large scale, although it is an increase of 5.2% compared to the benchmark network, the improvement is even more significant when other network fusion methods are used in the benchmark network. For the medium scale, NFPN increased by 1.1 percentage points. Except for RNFF, which can increase by 0.2 percentage points, other methods have

decreased significantly. For small objects, the Faster RCNN network using FPN performs the best, the NFPN reduces the least, and the accuracy of the remaining networks reduces significantly. For this phenomenon, this article believes that the reason is that after the top-down (FPN) process. However, the background pixels cause the first interference to the small-scale object; due to the lateral conv, there will be no original features in the up-sampled backbone network. Convolution and addition, thereby reducing this interference. After FPN, feature fusion is performed, that is, the bottom-up process, which transmits the position information of the shallow layer to the deep layer. Because the large object has a larger resolution, its influence is much more significant than the influence of the surrounding background pixels. The enhancement process from semantics (bottom-up) to location information (top-down) is verified by NFPN $\times$ 2 in Table 1.

In the experiment using ResRoIE + NFPN, mAP is increased by 2.0% for medium objects, while the accuracy of small objects is increased by 1.2%. The common point of all experiments in Table 3 is that the network of the feature fusion part increases, and the accuracy of small and medium objects decreases. For the large-scale experiment, NFPN $\times$ 2 + ResRoIE, the accuracy is 4.7 percentage points higher than that of the external NFPN + ResRoIE. Compared with BiFPN $\times$ 2 in Table 1, both increase network layers and decrease the accuracy. It shows that ResRoIE can reduce the feature fusion part and the grade fading made by increasing the network. In the experiment of the proposed RFP + ResRoIE method, after using RFP, the experiment is increased by 0.2%. The accuracy of each scale object is also improved, which verifies the hypothesis that there is noise in the feature fusion section. The RFP algorithm can reduce this noise. Subsequently, an adaptive weight standardization strategy was used to reduce the mutual influence between different scale features. The accuracy of the small, medium, and large objects were slightly improved. When SEnet is used to add attention to the feature fusion of each layer, the accuracy drops by 0.1%, indicating that the features of different layers are fused. While paying attention to the object feature, it also gives the same attention as the noise. Compared with some mainstream networks based on two stages, the experiment uses a combination of NFPN, ResRoIE, and RFP. Experiments prove that the algorithm offered in this paper can increase the precision of target detection. Balanced FPN imports the neck module in Libra RCNN into Faster RCNN, and the rest remains unchanged.

In addition, weight normalization (ConvAWS) is added to the network, and the idea of using weight fusion features in BiFPN in EfficientDet is referred to. The attention mechanism is used to add learnable weights when different layers of features are fused. Experiments show that the combination of ResRoIE and RFP introduced can realize more reliable performance, significantly improving compared to some current two-stage methods. For small and medium scales, in the experimental results, PANet, Balanced FPN, GRoIE, and RFP + ResRoIE have been upgraded in turn. Indicating that in a larger data set with a relatively balanced distribution of categories and scales, the more feature fusion is, the more accurate the detection of small and ordinary targets is presented in Figure 8. It is meriting that feature fusion may lead to information loss and noise interference in a small-scale data set due to the scale and category distribution imbalance, thereby triggering gradient descent. For large objects, the accuracy of GRoIE is lower than Faster RCNN. The reason is that features of all scales are used in its detection. The shallow features of small-scale objects are easily disturbed by upsampling, while large objects use disturbing features. So it manages to decrease the precision of the large object, as is shown in Figure 9.

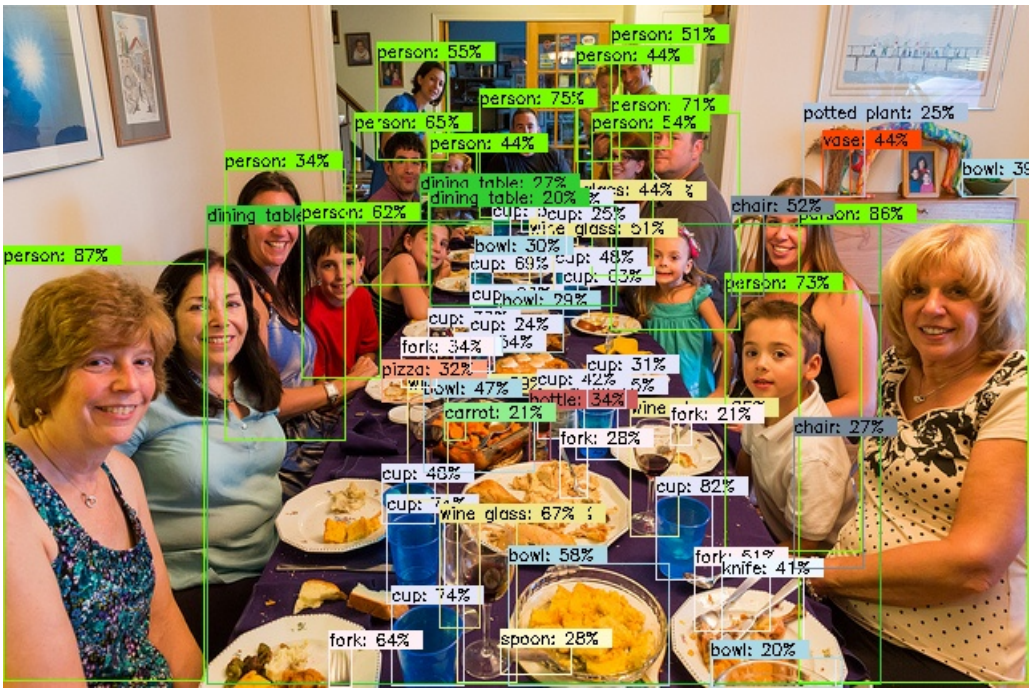

Figure 8. Detection of medium-sized and small objects.

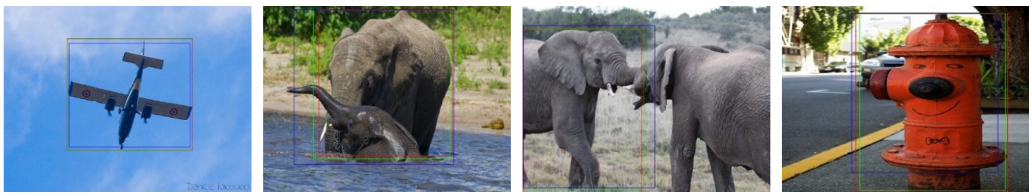

Figure 9. Detection of large objects.

5. Conclusions

To solve the insufficient features of objects, this work introduces some of the most advanced and representative network models based on Faster R-CNN, such as Libra R-CNN, Grid R-CNN, guided anchoring, and GRoIE, which we observed the performance of NFPN fusion, ResRoIE and the RFP architecture at different scales of precision. ResRoIE was added to the NFPN network on the Lisa data set, which verified that NFPN has vanishing gradients, and ResRoIE can alleviate this problem. Then, experiments were performed on other data sets of different scales. Among them, the small and medium data sets are the same as Lisa's conclusion that ResRoIE can alleviate the problem of gradient disappearance, thereby improving the accuracy of objects at various scales. The two-stage network experiment took Faster RCNN and its variant network as a reference after replacing the corresponding module in the model. In the experiment with ResNet50 as the backbone network, the mAP increase was the highest compared with GRoIE, from 37.5% to 40.1%. In a large-scale data set such as MS COCO, the NFPN + ResRoIE in this article can improve the detection accuracy in the current most advanced two-stage network. Some well-known methods in recent object detection are selected for comparison. In the experiment based on those methods, we keep ResNet50 as a fixed backbone network to observe the performance improvement by the feature RNFF. Experiments show that the algorithm proposed in this paper can improve the accuracy of object detection.

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

Funding: This work was supported, the National Natural Science Foundation of China (Grant No. 61871039, 62102033, 62171042, 61906017, 61802019), the Beijing Municipal Commission of Education Project (No. KM202111417001, KM201911417001), the Collaborative Innovation Center for Visual Intelligence (Grant No. CYXC2011), the Academic Research Projects of Beijing Union University(No. ZB10202003, ZK40202101, ZK120202104).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated and analysed during the study are included in this published article.

Acknowledgments: The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

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

References

1. Zhao, D.; Ding, B.; Wu, Y.; Chen, L.; Zhou, H. Unsupervised Learning from Videos for Object Discovery in Single Images. *Symmetry* **2021**, *13*, 38. [CrossRef]
2. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In E*uropean Conference on Computer Vision*; Springer: Dordrecht, The Netherlands, 2016; pp. 21–37.
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
4. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
5. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
6. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
7. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
8. Law, H.; Deng, J. CornerNet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
9. Law, H.; Teng, Y.; Russakovsky, O.; Deng, J. CornerNet-Lite: Efficient keypoint based object detection. *arXiv* **2019**, arXiv:1904.08900.
10. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6569–6578.
11. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9626–9635.
12. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 850–859.
13. 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.
14. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards balanced learning for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 821–830.
15. Rossi, L.; Karimi, A.; Prati, A. A novel region of interest extraction layer for instance segmentation. *arXiv* **2020**, arXiv:2004.13665.
16. Liu, Y.; Wang, Y.; Wang, S.; Liang, T.; Zhao, Q.; Tang, Z.; Ling, H. Cbnet: A novel composite backbone network architecture for object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11653–11660. [CrossRef]
17. Qin, Z.; Li, Z.; Zhang, Z.; Bao, Y.; Yu, G.; Peng, Y.; Sun, J. Thundernet: Towards real-time generic object detection on mobile devices. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6718–6727.

18. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. Cspnet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
19. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]
20. Wang, J.; Chen, K.; Yang, S.; Loy, C.C.; Lin, D. Region proposal by guided anchoring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 2965–2974.
21. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
23. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
24. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
25. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
26. Mogelmose, A.; Trivedi, M.M.; Moeslund, T.B. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1484–1497. [CrossRef]
27. Lin, Ts.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. *arXiv* **2014**, arXiv:1405.0312.
28. 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, 26 June–1 July 2016; pp. 770–778.
29. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 21–24 June 2021; pp. 13029–13038.
30. Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; Sun, J. You only look one-level feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 21–24 June 2021; pp. 13039–13048.
31. AlDahoul, N.; Abdul Karim, H.; Lye Abdullah, M.H.; Ahmad Fauzi, M.F.; Ba Wazir, A.S.; Mansor, S.; See, J. Transfer Detection of YOLO to Focus CNN's Attention on Nude Regions for Adult Content Detection. *Symmetry* **2021**, *13*, 26. [CrossRef]
32. Maninis, K.K.; Caelles, S.; Pont-Tuset, J.; Van Gool, L. Deep extreme cut: From extreme points to object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 616–625.
33. Zhang, Y.; Hu, C.; Lu, X. Improved YOLOv3 Object Classification in Intelligent Transportation System. *arXiv* **2020**, arXiv:2004.03948. .
34. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
35. Gao, S.; Cheng, M.; Zhao, K.; Zhang, X.; Yang, M.; Torr, P.H.S. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, 43, 652–662. [CrossRef] [PubMed]
36. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [CrossRef] [PubMed]
37. Kok, K.Y.; Rajendran, P. A Descriptor-Based Advanced Feature Detector for Improved Visual Tracking. *Symmetry* **2021**, *13*, 1337. [CrossRef]
38. Lin, D.; Shen, D.; Shen, S.; Ji, Y.; Lischinski, D.; Cohen-Or, D.; Huang, H. Zigzagnet: Fusing top-down and bottom-up context for object segmentation. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7490–7499.
39. Zhou, Z.; Lai, Q.; Ding, S.; Liu, S. Novel Joint Object Detection Algorithm Using Cascading Parallel Detectors. *Symmetry* **2021**, *13*, 137. [CrossRef]
40. Liang, H.; Yang, J.; Shao, M. FE-RetinaNet: Small Target Detection with Parallel Multi-Scale Feature Enhancement. *Symmetry* **2021**, *13*, 950. [CrossRef]