



Article

Road Traffic Sign Detection Method Based on RTS R-CNN Instance Segmentation Network

Guirong Zhang , Yiming Peng and Hai Wang * 

School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China; 2222104147@stmail.ujs.edu.cn (G.Z.); 3200401204@stmail.ujs.edu.cn (Y.P.)

* Correspondence: wanghai1019@163.com

Abstract: With the rapid development of the autonomous driving industry, there is increasing research on related perception tasks. However, research on road surface traffic sign detection tasks is still limited. There are two main challenges to this task. First, when the target object's pixel ratio is small, the detection accuracy often decreases. Second, the existing publicly available road surface traffic sign datasets have limited image data. To address these issues, this paper proposes a new instance segmentation network, RTS R-CNN, for road surface traffic sign detection tasks based on Mask R-CNN. The network can accurately perceive road surface traffic signs and provide important information for the autonomous driving decision-making system. Specifically, CSPDarkNet53_ECA is proposed in the feature extraction stage to enhance the performance of deep convolutional networks by increasing inter-channel interactions. Second, to improve the network's detection accuracy for small target objects, GR-PAFPN is proposed in the feature fusion part, which uses a residual feature enhancement module (RFA) and atrous spatial pyramid pooling (ASPP) to optimize PAFPN and introduces a balanced feature pyramid module (BFP) to handle the imbalanced feature information at different resolutions. Finally, data augmentation is used to generate more data and prevent overfitting in specific scenarios. The proposed method has been tested on the open-source dataset Ceymo, achieving a Macro F_1 -score of 87.56%, which is 2.3% higher than the baseline method, while the inference speed reaches 23.5 FPS.

Keywords: deep learning; autonomous driving; instance segmentation; road traffic sign detection



Citation: Zhang, G.; Peng, Y.; Wang, H. Road Traffic Sign Detection Method Based on RTS R-CNN Instance Segmentation Network. *Sensors* **2023**, *23*, 6543. <https://doi.org/10.3390/s23146543>

Academic Editor: Xiangao Xia

Received: 12 May 2023

Revised: 17 July 2023

Accepted: 18 July 2023

Published: 20 July 2023



Copyright: © 2023 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

In recent years, with the rapid development of parallel computing technology based on image data and the rapid iteration of sensors, research on autonomous driving perception algorithms has been greatly promoted. The information that the environmental perception system of the smart car needs to extract is the dynamic and static obstacles and the road surface environment. Dynamic obstacles mainly refer to other vehicles around the vehicle, pedestrians, and dynamic or static objects on the ground that may affect the safe driving of the vehicle; road pavement environment mainly refers to the geographical environment around the vehicle and road traffic information, such as lane lines, road traffic signs, etc. Among them is the accurate detection of traffic signs on the road surface, especially the accurate perception of straight arrows, left arrows, right arrows, straight-left arrows, straight-right arrows, pedestrian crossings, and slow signs, which is beneficial for intelligent vehicles to make normative braking during driving and, when combined with high-precision maps, can assist in path planning.

Research on traffic sign detection on road surfaces can be roughly divided into two categories: one is manual feature methods and object detection methods based on deep learning. Most of the traditional methods for road traffic sign detection manually extract the basic features of the target object, such as color, edge, and texture, which largely rely on the method designed by the authors. For example, Tang et al. [1] utilized histograms of

oriented gradients (HOG) [2] and support vector machines (SVM) [3] with region of interest (ROI) constraints, which demonstrated good performance on the dataset. Compared with manual methods, deep learning-based methods show better results and stability in the feature extraction of road signs.

In deep learning-based computer vision, classification can be defined as predicting the class of an object in an image or providing a list of classes of objects in an image based on its classification score. Object detection or localization is a gradual process from coarse inference to fine inference, which not only provides the category of the image object but also gives the location of the classified image object in the form of a bounding box or center. The purpose of semantic segmentation is to obtain accurate inference results by predicting the label of each pixel in the image. Each pixel is classified and labeled according to the object or region in which it is located. Instance segmentation is essentially a more fine-grained visual perception task that combines two classic computer vision environment understanding subtasks: object detection and semantic segmentation. Therefore, instance segmentation can not only distinguish pixel categories but also accurately segment different instance objects belonging to the same semantic category, thereby providing rich perceptual information for downstream decision planning and other tasks. In the actual autonomous driving scene, for the detection task of road traffic sign the autonomous driving perception system not only needs to accurately identify the position of each type of road traffic signs, but also needs to accurately classify each pixel on the road traffic sign area. Therefore, using the instance segmentation method to detect road traffic signs can not only obtain the positions of different traffic signs but also classify each type of traffic sign at the pixel level.

However, the pixel ratio of road traffic signs in traffic scenes is relatively small, and their features are relatively sparse, which greatly limits the accuracy of instance segmentation algorithms. In addition, the public data for the road traffic sign detection task is very scarce, and most of the public data sets are small data sets, and the instance segmentation algorithm needs a lot of training to have better robustness. Based on these two challenges, this paper proposes the RTS R-CNN instance segmentation network, which is improved on the basis of Mask R-CNN [4]. The feature extraction network used by Mask R-CNN is ResNet [5]. Although the network stacks many convolutional layers, its effective depth is low, which easily leads to poor detection performance of small target objects, so we use efficient channel attention. The module ECA [6] further optimizes CSPDarknet53 and proposes CSPDarkNet53_ECA to replace the feature extraction module in the original network. At the same time, in order to improve the defects of FPN [7] and further improve the detection accuracy of the network for small target objects, we propose GR-PAFPN on the basis of PAFPN [8] and use the residual feature augmentation module (RFA) [9] to improve the top-down first-level features. Indicates that atrous spatial pyramid pooling (ASPP) [10] is introduced to obtain multi-scale feature information, and the balanced feature pyramid module (BFP) is used to enhance the expressive ability of each level of feature maps. Aiming at the small amount of road traffic sign data, we flipped and adjusted the color space on the training set pictures to increase the number of images from the small data set and changed the pixel coordinates of the flipped pictures.

In summary, the main contributions of this paper are as follows: Firstly, we proposed CSPDarkNet53_ECA to replace the original feature extraction module in the network. It enhances the channel features of the input feature map and improves the performance of deep convolutional networks by adding cross-channel interactions. Secondly, we proposed GR-PAFPN, which optimizes PAFPN using Residual Feature Aggregation (RFA) modules and Atrous Spatial Pyramid Pooling (ASPP) and introduces the Balanced Feature Pyramid (BFP) module to handle imbalanced information in different resolutions, thus improving the detection accuracy of small objects in the network. Finally, we increased the number of images from a small dataset by using data augmentation techniques such as flipping and color space adjustment, which helps to avoid overfitting specific scenarios.

2. Related Works

In this part, we review the road traffic sign detection algorithm based on deep learning and the instance segmentation algorithm, respectively.

2.1. Road Traffic Sign Detection

With the further development of deep learning, CNN-based object detection significantly improves the performance of road traffic sign detection. A convolutional neural network model is used in [11] that combines ResNet-101 [5] and pyramid pooling ensembles to obtain lane and road sign semantic segmentation output. Their architecture achieves average results on the TRoM [11] dataset, which can be considered a performance baseline. VPGNet [12] is an end-to-end model based on CNN architecture for simultaneous detection of lanes and road traffic signs. They regard pavement traffic sign detection as a grid regression task and then use grid sampling and box clustering as post-processing techniques to merge grid cells. However, they focus more on lane detection and vanishing point prediction tasks and only provide experimental results for four pavement traffic sign categories. It is worth mentioning that the authors released a new dataset that was publicly collected under various weather conditions in Korea. Hoang et al. [13] detected and classified arrow and bicycle markings on the road based on adaptive ROI and RetinaNet [14]. The results show that Adaptive ROI outperforms other methods. In pursuit of real-time detection, Zhang et al. [15] propose a method consisting of three modules: preprocessing, road traffic sign detection, and segmentation. In the second stage, a lightweight network combined with a Siamese attention mechanism is employed to improve accuracy and enhance sensitivity to road traffic signs. For the segmentation module, the segmented objects can achieve pixel-level accuracy at a lower computational cost. Ye et al. [16] proposed a two-stage model combining YOLOv2 [17] with a spatial transformation network (STN) [18] to address the distortion of road markings. Even for two-stage models, the proposed method achieves good performance with less computation. In conclusion, deep learning-based methods are more robust and stable than traditional feature extraction methods and can be applied to different scenarios with higher accuracy.

2.2. Instance Segmentation Algorithm

Instance segmentation algorithms based on deep learning can be mainly divided into four mainstream technical routes [19]: bottom-up methods based on semantic segmentation, detection-based methods, direct instance segmentation methods, and query-based methods. Semantic segmentation-based methods include SGN [20], which first generates segmentation masks with fixed semantic categories by pixel-by-pixel classification and then groups pixels into individual instances by means of clustering, metric learning, etc. This method can better retain the shallow, fine-grained features and geometric shape information, but its post-processing operation is more cumbersome, and the accuracy rate is low in complex working conditions with large objects and many categories. In order not to rely on methods such as bounding boxes or pixel embeddings, the direct instance segmentation method uses PolarMask [21] by directly predicting instance masks and semantic categories. Based on the intuition that the center position and size of the instance object are different, the SOLO series [22,23] divides the mask according to the image position. Specifically, the input image is divided into several grids, and the grid unit where the center of the instance is located is responsible for assigning a category to each pixel of the instance. Although this method effectively improves the detection speed, when the centers of multiple objects fall on the same grid, the predicted mask and category will be confused, which will affect the driving safety of autonomous vehicles. Detection-based methods first locate prior instances via powerful detectors [24–26] and then perform segmentation within the resulting regions of interest [27–30]. Thanks to the development of object detection, such methods are usually more accurate. The query-based approach utilizes the Transformer mechanism and expresses the object of interest through a query. ISTR [31] is the first end-to-end instance segmentation using low-dimensional mask embeddings.

According to the required order of object localization and mask generation, current detection-based instance segmentation methods can be divided into two types: single-stage methods and two-stage methods. The two-stage algorithm based on the region candidate network has high accuracy, but the small batch training of the model consumes a lot of memory resources and takes a long time for reasoning, such as Cascade Mask R-CNN [32]. A single-stage algorithm such as YoLACT++ [33], which performs localization and segmentation at the same time, has a faster inference speed, but its detection accuracy is obviously lacking.

3. Methodology

The instance segmentation network Mask R-CNN is mainly composed of the feature extraction module, the feature fusion module, the region candidate module RPN, and the prediction module. The road traffic sign detection algorithm model RST R-CNN proposed in this paper is based on Mask R-CNN. The feature extraction module and the feature fusion module are optimized separately. Specifically, the feature extraction network ResNet50 of the original model is replaced by the CSPDarkNet53, and on this basis, an Efficient Channel Attention module (ECA) is integrated. Considering that road traffic signs account for fewer pixels in the original image problem, we improve the Channel Enhanced Feature Pyramid Network PAFPN by enhancing the information flow and propose the GR-PAFPN module to replace the feature extraction network FPN in the original model. As shown in Figure 1, the network framework diagram of RTS R-CNN.

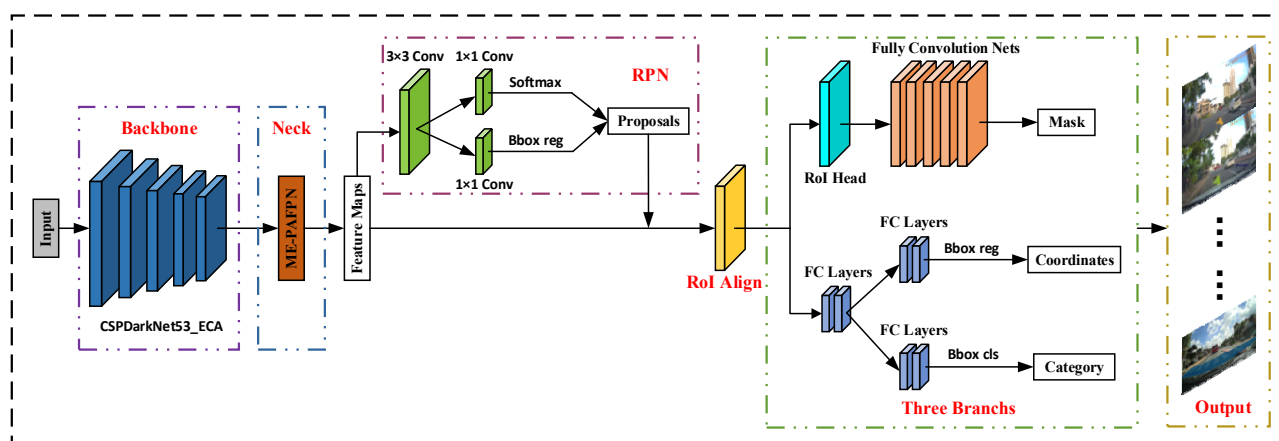


Figure 1. The network framework diagram of RTS R-CNN.

3.1. Backbone

CSPDarknet53 has demonstrated excellent feature extraction capabilities in YOLOv4. It mainly includes 5 CSP_Res modules combined with a cross-stage partial network (CSP) and a residual network. It uses its large residual edge to expand the receptive field and Integrating local context information can improve the feature extraction ability of the convolutional network without losing detection accuracy, improve detection accuracy, and at the same time reduce the calculation loss of the entire model. Since road traffic signs are small targets, this puts high demands on the detection accuracy of the network model, so we use the efficient channel attention module ECA to further optimize CSPDarknet53. As shown in Figure 2, the structure diagram of CSPDarknet53_ECA. We add four high-efficiency channel attention modules ECA between the input and output of five CSP_Res modules and strengthen the channel features of the input feature map by increasing cross-channel interaction. Improving the performance of deep convolutional networks.

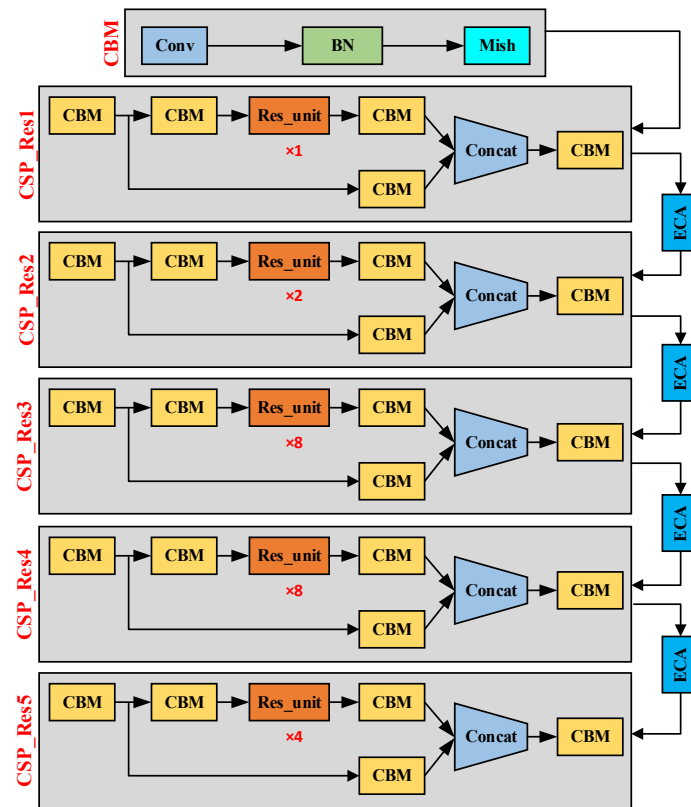


Figure 2. The structure of CSPDarknet53_ECA.

ECA removes the fully connected layer on the basis of the Squeeze-and-Excitation Network (SENet) and uses a 1×1 convolutional layer directly after the global average pooling layer so as to avoid learning channel attention information. The time channel dimension is reduced, and the number of parameters is reduced. In the conventional convolution operation, the size of the convolution kernel will affect the size of the receptive field. In order to extract different ranges of features from different input feature maps, ECA uses a dynamic convolution kernel to do 1×1 convolution to learn between different channels. importance. The term dynamic convolution kernel means that the size of the convolution kernel is adaptively changed through a function. In a layer with a large number of channels, a larger convolution kernel is used to perform 1×1 convolution, which enables more cross-channel interaction. In the layer with a small number of channels, a smaller convolution kernel is used to perform 1×1 convolution, so that there is less cross-channel interaction. The convolution kernel adaptive function is defined as follows:

$$k = \Phi(C) = \left\lfloor \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right\rfloor_{odd} \quad (1)$$

where, k indicates the size of the convolution kernel; C indicates the number of channels; $\lfloor \cdot \rfloor_{odd}$ indicates that k can only take odd numbers; $\gamma, b = 1$.

A schematic diagram of the structure of the ECA attention module is shown in Figure 3. The specific implementation process is: (1) first input the feature map, and its dimension is $H \times W \times C$; (2) perform spatial feature compression on the input feature map and use the global average pooling GAP in the spatial dimension to obtain $1 \times 1 \times C$ feature map; (3) Convolve the compressed feature map through 1×1 to learn the importance between different channels. At this time, the output dimension is still $1 \times 1 \times C$; (4) Finally, the channel Attention combination, the feature map $1 \times 1 \times C$ of channel attention, and the original input feature map $H \times W \times C$ are multiplied channel by channel, and finally the feature map with channel attention is output.

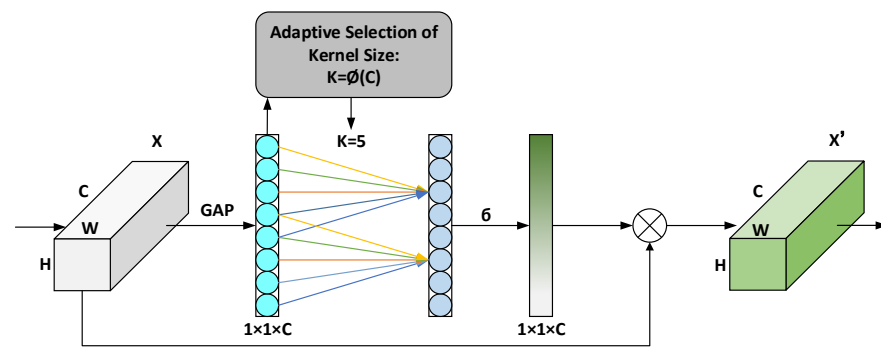


Figure 3. The structure of ECA module.

3.2. Feature Fusion Module

Mask R-CNN uses FPN as a feature extraction network. FPN has the following problems with the fusion of information features: (1) The information on the feature map is lost. FPN is a top-down propagation method. From the low-level feature map to the highest feature map, it must go through multiple network layers, and the network increases the difficulty of obtaining the initial image. (2) Multi-scale information is lost. FPN does not fully consider the difference in semantic information between different layers. Direct fusion between different features will reduce the ability for multi-scale feature representation. (3) The semantic information of non-adjacent layers cannot be fused. In FPN, only the semantic information of adjacent layers can be directly fused, and the semantic information of non-adjacent layers is diluted, which easily leads to the problem of imbalanced semantic features.

As shown in Figure 4, since a high response to object edges or instance parts is a strong indicator for accurately locating instances, PAFPN enhances the overall feature layer by constructing a bottom-up feature fusion network that reduces the propagation path of the main feature layer. Positioning ability: the red dotted line indicates that in the FPN network, the main feature map passes through multiple network layers from top to bottom, and the information of the main feature map is seriously lost. The purple dashed line represents the bottom-up feature fusion. Shallow features are connected to P2 by fusing the raw FPN at the bottom layer and then transferred from P2 to the top layer with bottom-up feature fusion. The number of layers is less than 10, which can better preserve shallow feature information. Although PAFPN improves the problem of loss of main feature information, it does not solve the problem of multi-scale information loss in FPN, and non-adjacent layer semantic information cannot be directly fused. To solve the problem of multi-scale information loss, we use Residual Feature Augmentation (RFA) and Atrous Spatial Pyramid Pooling (ASPP) to optimize PAFPN; in order to balance semantic features, we introduce the balanced feature pyramid Module (BFP) to deal with unbalanced information in each resolution.

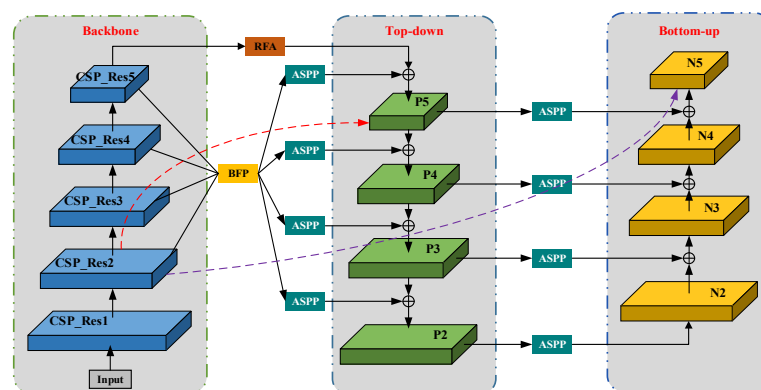


Figure 4. The structure of GR-PAFPA.

3.2.1. Residual Feature Augmentation Module

In FPN, the dimensionality reduction operation is performed when the low-level features are fused from the highest layer to reduce the channel features, resulting in the loss of multi-scale information. To make up for this shortcoming, we use RFA to improve the top-down, one-level feature representation and fuse the feature information without information loss in CSP_Res5 into P5. As shown in Figure 5, firstly, the feature map output by CSP_Res5 is divided into three feature layers of different scales using Ratio-invariant Adaptive Pooling (RAP), and the scales are 0.1 times, 0.2 times, and 0.3 times that of the input feature map. Each feature map is then passed through a 1×1 convolutional layer, the number of channels C is reduced to 256, and finally bilinear interpolation is used for upsampling to restore the feature map to the scale size of the original input feature map. Considering the aliasing effect caused by interpolation, Adaptive Spatial Fusion (ASF) is used to adaptively combine the generated spatial weight probability map and the upsampled hierarchical feature map so that the output feature map has multi-scale feature information.

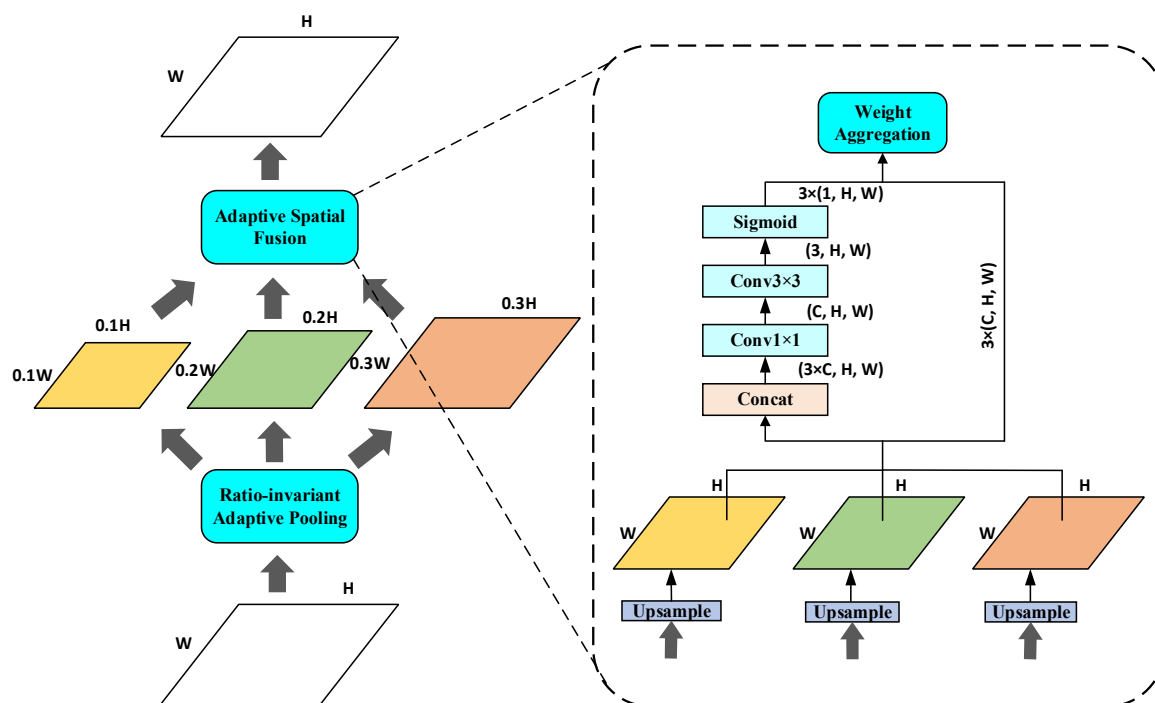


Figure 5. The structure of RFA.

3.2.2. Atrous Spatial Pyramid Pooling

FPN reduces the number of channels of the feature map through a 1×1 convolution operation, which also suppresses the representation of multi-scale features, so we introduce ASPP to obtain multi-scale feature information. As shown in Figure 6, the four parallel branches of ASPP generate images of different scales through atrous convolution [34], using different atrous convolution rates and convolution kernels, where the convolution rates are 1, 3, and 6, respectively, and the convolution kernels sizes are 1×1 and 3×3 , respectively. Using the ASPP module to improve the model's performance in the fusion stage of each branch effectively expands the receptive field of feature images and improves the network's ability to extract abstract information.

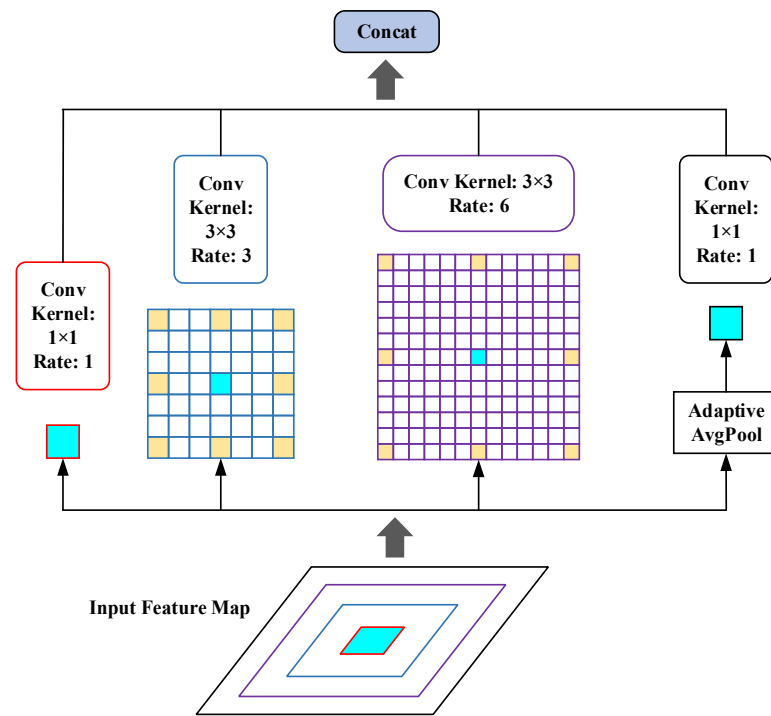


Figure 6. The structure of ASPP.

3.2.3. Balance Feature Pyramid Module

High-level semantic information and shallow detail information can complement each other to improve the target detection effect, and the fused features need to deal with unbalanced information at each resolution. Therefore, we introduce the BFP module to scale, integrate, and refine the four-level feature maps and use the feature map information of multiple levels to enhance the expressive ability of each level of feature maps and realize the ability to deal with unbalanced information. As shown in Figure 7, in order to aggregate multi-level features, upsampling interpolation is used for small-size feature layers, adaptive maximum pooling is used for large-size feature maps, and multi-level features $\{C_2, C_3, C_4, C_5\}$ are adjusted To C_4 size, a balanced semantic is obtained by simple integrating, which can be expressed as:

$$\text{Integrate} = \frac{1}{L} \sum_{l=l_{\min}}^{l_{\max}} C_l \quad (2)$$

where, L represents the total number of layers of the predicted feature layer, and the level of the predicted feature layer is defined as C_l .

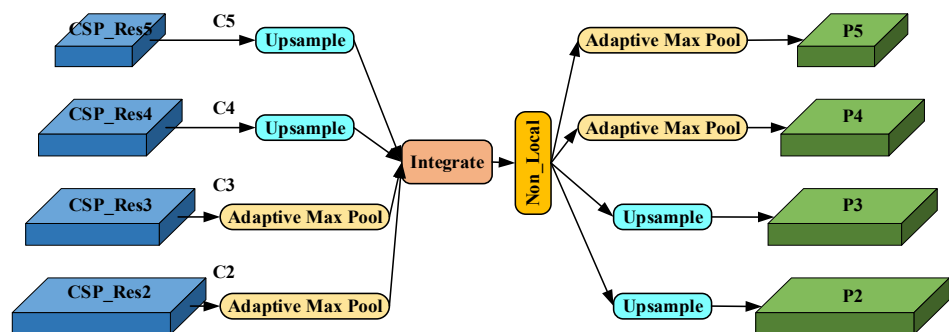


Figure 7. The structure of BFP.

Then the non-local network [35] is used to further optimize the balance semantics to enhance the fusion features. As shown in Figure 8, first linearly map the input feature map to obtain the three features θ , ϕ , g after channel compression; then perform feature merging through the reshape operation; then perform matrix point multiplication on θ and ϕ and use the activation function Get the weight, that is, the attention coefficient; then multiply it with the feature g to expand the number of channels; finally, sum the residual with the input feature map and output it.

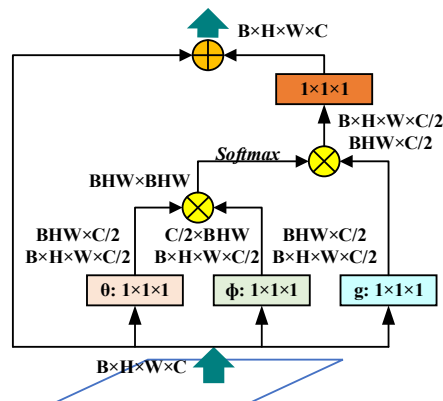


Figure 8. The structure of non-local module.

After the non-Local module, we rescale the obtained features using the same but reversed process of semantic balancing to enhance the original features. During this process, each predictive feature layer obtains the same information from the other predictive feature layers.

3.3. Data Augmentation

Data augmentation is a common method for deep learning models that aims to increase the number of images from small datasets to avoid overfitting specific scenarios. This section will detail data augmentation on images. Data augmentation on labels due to changes in pixel coordinates will also be addressed in this section.

3.3.1. Data Augmentation on Image

Deep learning networks usually require a large amount of training data to achieve better results. In the case of limited data acquisition, data augmentation techniques are used to generate more data from existing datasets, thereby increasing the diversity of original images and making up for the lack of data. Common techniques for data augmentation include: (1) geometric transformation: randomly flip, crop, rotate, shear, or translate the image, (2) color space transformation: change the color channel space or try to map RGB to other color spaces, (3) Noise injection: A matrix of random values sampled from a Gaussian distribution is added to the pixels of the image; (4) Kernel filter: A kernel filter on the image for convolution operations such as sharpening and blurring. After data augmentation, what people see with their eyes is still easily recognizable as the same image, but to the deep learning model, these processed images are completely new images.

Considering the simple features and monotonous colors of road traffic signs, they do not contain diverse structural features for object detection models. Therefore, this paper uses flipped and color space-adjusted (brightness and contrast) training datasets. Flipping is an effective approach and has been shown to be useful for improving the performance of deep learning models. Furthermore, color space adjustments are the easiest and most common technique for changing the brightness of an image. In the road surface environment, the diversity of lighting conditions and weather conditions has an impact on the accuracy of the model, so data augmentation is an important technique to change the image through color space adjustment. Flipping the image horizontally and vertically is a common method in geometric transformation. Additionally, brightness adjustment is

implemented on the training data to transform brightness-related channels according to value settings. Therefore, it can make the image slightly brighter or darker to enhance the lighting conditions of the image. Contrast adjustment is also one of the data augmentation techniques used to rescale the range of intensity values in an image. Contrast is the ratio between the lightest and darkest areas of an image. The larger the ratio, the more shades of gray there are from black to white, which makes objects or boundaries in the image more distinguishable. As a result, the contrast of white road markings on black asphalt roads is enhanced, improving visual perception. Finally, quadruple image copies are generated using data augmentation techniques that increase the amount of data from the original image without adding additional time cost.

3.3.2. Data Augmentation on Label

The ground-truth labels of target objects are crucial for supervised learning networks. Before training the model, it is necessary to label the target object labels of the dataset as ground truth. However, changing more original images through data augmentation requires a new labeling of the labels of the original images, which is a very time-consuming task. Therefore, this study increases the amount of data by performing data augmentation and homography transformation on images in a limited dataset. After data augmentation and homography transformation, annotations do not need to be manually labeled again. In view of the change in pixel coordinates, some labels of the augmented data need to be modified. After data augmentation such as brightness and contrast adjustments, the labels are the same as the original annotation files, while the pixel coordinates of flipped images need to be flipped horizontally. The pixel coordinates of the flipped image are transformed from the left part to the right part.

3.4. Loss Function during Training

The loss function used in this article mainly continues the loss function in Mask R-CNN, namely:

$$L = L_{cls} + L_{box} + L_{mask} \quad (3)$$

They are the classification loss L_{cls} , the regression box loss L_{box} and the mask loss L_{mask} . The first two losses are consistent with the object detection network Fast-RCNN [36]. The calculation formula of L_{mask} is as follows:

$$L_{mask} = \frac{1}{m^2} \sum_i^k \binom{1^k}{1} \sum_1^{m^2} [-y \log \text{sigmoid}(x) - (1 - y) \log(1 - \text{sigmoid}(x))] \quad (4)$$

where, 1^k means that when the k th channel corresponds to the true category of the target, it is 1, otherwise it is 0; y means the label value of the mask at the current position; the output value of the current position of x , $\text{sigmoid}(x)$ means that the output x has passed through sigmoid. The result after the function transformation; m^2 represents the dimension.

4. Experiments and Results

4.1. Introduction to Public Dataset

The public dataset used in this experiment is Ceymo [37], a dataset for road traffic sign detection, which consists of 2887 images, of which 4706 road sign instances belong to 11 categories. The images have a high resolution of 1920×1080 and capture a wide range of traffic, lighting, and weather conditions. This dataset uses the labelme annotation tool [38] to manually annotate road markings belonging to 11 categories into polygons. Each image has a JSON file that contains the coordinates of the polygons that enclose the road markings in that image. In addition to polygon annotations in JSON format, bounding box annotations in XML format and pixel-level segmentation masks in PNG format are provided to facilitate different road marking detection methods.

This dataset uses F_1 -score and Macro F_1 -score as rating indicators. Calculate the IoU between the prediction and the road truth. When the IoU is greater than 0.3, the

corresponding prediction is regarded as the real prediction. The total number of true positives (TP), false positives (FP), and false negatives (FN) is used to calculate precision, recall and F_1 -score as follows:

$$\text{precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (6)$$

The Macro F_1 -score is calculated as the average of the individual F_1 -score of the 7 categories in the dataset, as follows:

$$\text{Macro } F_1\text{-score} = \frac{1}{C} \sum_{i=1}^C F_1\text{-score}_i \quad (7)$$

where, C represents different classes in the dataset, and the Macro F_1 -score has the same importance for all classes, no matter how frequently they appear in the dataset.

4.2. Experimental Details

Instance segmentation models perform a lot of complex matrix operations and floating-point operations during training to find the optimal solution. The instance segmentation model proposed in this paper for road traffic sign detection uses the Pytorch1.13.1 deep learning framework for distributed data parallel training and synchronous batch normalization based on 64-bit NVIDIA (NVIDIA Co., Ltd., Santa Clara, CA, USA) and Ubuntu20.04 (Canonical Co., Ltd., London, UK), with two GTX3090 graphics cards (NVIDIA Co., Ltd.). We use stochastic gradient descent (SGD) as the optimizer, with the weight decay rate set to 4×10^{-4} and the momentum set to 0.9.

The choice of learning rate, batch size, and loss function affects the training speed of the segmentation model and the final segmentation accuracy. A higher learning rate will lead to a sharp increase in the loss while increasing the training speed, so it is more suitable to use a higher learning rate in the initial stage of training. If the learning rate is set too low, the convergence speed of the model will be slower, and it will be easier to find the optimal solution of the model, but the model will appear overfitting, so setting a smaller learning rate in the later stages of training will be more suitable. Considering the appeal factor, we use a multi-adaptive learning rate, that is, the learning rate will continue to decrease as the number of iterations increases, which is expressed as follows:

$$l_r = \ln l_r \left(1 - \frac{\text{epoch}}{\text{epoch}_{\max}} \right)^{\text{power}} \quad (8)$$

where, $\ln l_r$ represents the initial learning rate, which is set to 0.1; epoch represents the number of current epochs; epoch_{\max} represents the maximum number of epochs, which is set to 150; and power represents exponential decay, which is set to 1. Similarly, the size of the batch setting will also affect the training speed and segmentation accuracy; if the batch size is too large, the training speed will be significantly improved, but the utilization of GPU memory will be affected, resulting in a decrease in segmentation accuracy. In this paper, we will set the batch size to 6.

4.3. Experimental Results

In this section, we compare the results of the proposed model on the dataset Ceymo with some of the current advanced methods, including MaskLab [39], Cascade Mask R-CNN, RetinaNet, and YoLACT++. This paper chooses Mask R-CNN as the baseline model (Baseline). The first two grids are instance segmentation methods based on a two-stage framework, and the quality of their position mask generation is highly dependent on the object localization network. The latter two networks are instance segmentation methods

based on a single-stage framework. Using the global mask does not require the processes of clipping and RoI Align, but directly predicts an instance from the entire feature map. The model proposed in this paper is based on a two-stage framework. For the reliability of the experimental comparison, we use networks based on different frameworks for comparison. The inference speeds of all the above algorithms follow the default configuration, and the comparison results are shown in Table 1.

Table 1. Comparison results with mainstream schemes on the Ceymo dataset.

Category	Masklab	Cascade Mask R-CNN	RetinaNet	YoLACT++	Ours
Straight Arrow	87.51	87.39	86.98	87.23	89.31
Left Arrow	74.27	73.97	72.11	71.06	75.36
Right Arrow	90.64	91.93	90.21	89.47	91.32
Straight-Left Arrow	80.21	81.92	79.39	80.11	82.69
Straight-Right Arrow	82.51	79.06	78.72	81.49	80.11
Slow	95.56	95.70	96.01	93.38	94.34
Pedestrian Crossing	94.95	95.44	93.56	92.61	96.86
FPS	14.2	13.3	23.6	24.0	23.5
Params(M)	75.32	77.1	37.74	35.29	36.85
Macro F_1 -score	86.52	86.49	85.28	85.05	87.56

It is observed that the instance segmentation model based on the two-stage framework has an overall better instance segmentation effect than the single-stage method, but the network reasoning speed is significantly slower. Compared with the compared network, the instance segmentation performance of the method proposed in this paper is excellent overall, and the Macro F_1 -score can reach 87.14%, especially in targets with relatively small proportions of pixels, such as going straight, turning left, and decelerating signs. F_1 -score reached 89.31%, 75.36%, and 94.34%, respectively. It is obvious that our network model has more advantages, for instance, for the segmentation effect of small targets. In terms of inference speed, although it is slightly lower than YoLACT++ based on the single-stage framework, it has a greater advantage compared with the two-stage instance segmentation network.

In order to further verify the effectiveness of the proposed method, we selected 4 pictures from the verification set for visualization, and the results are shown in Figure 9. It can be seen that the method proposed in this paper works well in congested, cloudy, rainy, and dark scenes. A better instance segmentation effect is obtained. At the same time, we also visually compared our method with the RetinaNet of the single-stage framework and the Cascade Mask R-CNN of the two-stage framework. The results are shown in Figure 10. It can be seen intuitively that there are small The problem of missed detection of targets and false detection of large targets, as well as the target of worn-out ground traffic signs cannot be accurately identified. Qualitative results show that the proposed method significantly improves the misdetection of large-sized traffic signs, reduces the misdetection rate of small traffic signs in the foreground area to a certain extent, and cannot accurately predict the wear and tear of traffic signs.



Figure 9. Comparison chart of visualization results on the Ceymo dataset.

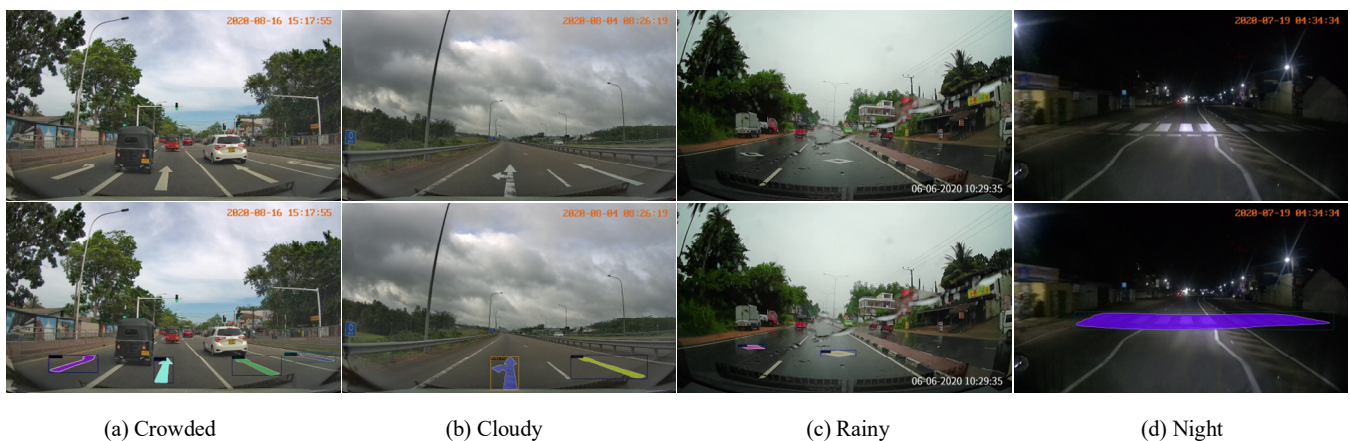


Figure 10. Visualization in extreme scenarios.

4.4. Ablation Experiment

In order to verify the effectiveness of each component module of the proposed method, we use Mask R-CNN as the Baseline to add each module in turn and test them on the Ceymo verification set, respectively. The performance results are shown in Table 2. By replacing the feature extraction network ResNet in the original network with CSPDarknet53_ECA, the Macro F_1 -score of the model increased by 0.81%, which effectively verified that integrating the ECA attention module into CSPDarknet53 as the backbone network can effectively improve the detection and segmentation performance of the model. Replace the feature fusion module FPN in the original network with GR-PAFPN. This feature fusion is designed and integrated into the original PAFPN to incorporate the residual feature augmentation module RFA, the atrous spatial pyramid pooling ASPP, and the BFP that can balance semantic features. The Macro F_1 -score of the model It has been improved by 0.16%, 0.79%, and 0.09% in turn, which proves that the feature fusion module we designed improves the problem that the information of traditional FPN feature maps and multi-scale information is easily lost and avoids the non-fusion of semantic information in non-adjacent layers.

Finally, data enhancement was used, and the Macro F_1 -score of the model was increased by 0.55%, which proved that flipping and color space adjustment on images and labels can effectively improve the detection and segmentation effects of the model.

Table 2. Ablation experiments on the Ceymo dataset. “√” indicates that the current module was used in the ablation experiment.

Baseline	CSPDarknet53_ECARFA	ASPP	BFP	Macro F_1 -Score
√				85.26
√	√			85.97
√	√	√		86.13
√	√	√	√	86.92
√	√	√	√	87.56

5. Conclusions

This paper proposes an instance segmentation framework, RTS R-CNN, based on the Mask R-CNN algorithm for road traffic sign detection tasks. The CSPDarknet53_ECA feature extraction network is proposed, which further optimizes CSPDarknet53 using the efficient channel attention module to strengthen the channel features of the input feature maps and improve the performance of deep convolutional networks by increasing cross-channel interaction. To improve the detection accuracy of small targets, GR-PAFPN is proposed in the feature fusion part, which optimizes PAFPN using the Residual Feature Aggregation module (RFA) and the Atrous Spatial Pyramid Pooling (ASPP) and introduces a Balanced Feature Pyramid module (BFP) to handle the imbalanced feature information in various resolutions. To address the issue of small road traffic sign image datasets, data augmentation techniques such as flipping and color space adjustment are used to increase the number of images from small datasets and prevent overfitting of algorithms to specific scenarios. The results on the publicly available Ceymo dataset demonstrate that the proposed algorithm is significantly better than the original method, with a Macro F_1 -score of 87.56%, which is 2.3% higher than the baseline method. Moreover, compared with other advanced instance segmentation networks, our proposed method significantly improves the false detection of large traffic signs and reduces the under-detection rate of small traffic signs in the far-field area to some extent. However, accurate predictions cannot be made for traffic signs that are heavily worn out. Future work will focus on improving the detection accuracy of these signs. Moreover, the issue of insufficient publicly available datasets for road traffic sign detection can be addressed by collecting a large number of images using onboard cameras on real transport vehicles and annotating them using professional annotation software. Since road traffic signs are small objects with a low pixel ratio, attention mechanisms can be designed in the future to improve the segmentation performance of small targets. This would help to further enhance the detection accuracy of small road traffic signs.

Author Contributions: Methodology, G.Z.; Software, Y.P.; Validation, G.Z. and H.W.; Writing—original draft preparation, G.Z. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (52072160), Key Research and Development Program of Jiangsu Province (BE2020083-3).

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. Tang, Z.; Boukerche, A. An improved algorithm for road markings detection with SVM and ROI restriction: Comparison with a rule-based model. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
2. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
3. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [\[CrossRef\]](#)
4. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
6. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
7. Lin, T.Y.; 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.
8. 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.
9. Guo, C.; Fan, B.; Zhang, Q.; Xiang, S.; Pan, C. Augfpn: Improving multi-scale feature learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12595–12604.
10. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [\[CrossRef\]](#)
11. Liu, X.; Deng, Z.; Lu, H.; Cao, L. Benchmark for road marking detection: Dataset specification and performance baseline. In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
12. Lee, S.; Kim, J.; Shin Yoon, J.; Shin, S.; Bailo, O.; Kim, N.; Lee, T.H.; Seok Hong, H.; Han, S.H.; So Kweon, I. Vpgnet: V anishing point guided network for lane and road marking detection and recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1965–1973.
13. Hoang, T.M.; Nam, S.H.; Park, K.R. Enhanced Detection and Recognition of Road Markings based on Adaptive Region of Interest and Deep Learning. *IEEE Access* **2019**, *7*, 109817–109832. [\[CrossRef\]](#)
14. 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.
15. Zhang, W.; Mi, Z.; Zheng, Y.; Gao, Q.; Li, W. Road Marking Segmentation based on Siamese Attention Module and Maximum Stable External Region. *IEEE Access* **2019**, *7*, 143710–143720. [\[CrossRef\]](#)
16. Ye, X.Y.; Hong, D.S.; Chen, H.H.; Hsiao, P.Y.; Fu, L.C. A Two-Stage Real-Time YOLOv2-based Road Marking Detector with Lightweight Spatial Transformation-Invariant Classification. *Image Vis. Comput.* **2020**, *102*, 103978. [\[CrossRef\]](#)
17. 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.
18. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2017–2025.
19. Gu, W.; Bai, S.; Kong, L. A review on 2D instance segmentation based on deep neural networks. *Image Vis. Comput.* **2022**, *120*, 104401. [\[CrossRef\]](#)
20. Liu, S.; Jia, J.; Fidler, S.; Urtasun, R. Sgn: Sequential grouping networks for instance segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
21. Xie, E.; Sun, P.; Song, X.; Wang, W.; Liu, X.; Liang, D.; Shen, C.; Luo, P. Polarmask: Single shot instance segmentation with polar representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
22. Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; Li, L. Solo: Segmenting objects by locations. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020.
23. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. Solov2: Dynamic and fast instance segmentation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17721–17732.
24. 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, Republic of Korea, 27 October–2 November 2019.
25. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
26. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
27. Tian, Z.; Shen, C.; Chen, H. Conditional convolutions for instance segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020.

28. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask scoring r-cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
29. Kirillov, A.; Wu, Y.; He, K.; Girshick, R. Pointrend: Image segmentation as rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
30. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
31. Hu, J.; Cao, L.; Lu, Y.; Zhang, S.; Wang, Y.; Li, K.; Huang, F.; Shao, L.; Ji, R. Istr: End-to-end instance segmentation with transformers. *arXiv* **2021**, arXiv:2105.00637.
32. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1483–1498. [[CrossRef](#)] [[PubMed](#)]
33. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact++: Better real-time instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1108–1121. [[CrossRef](#)] [[PubMed](#)]
34. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2015**, arXiv:1511.07122.
35. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
36. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
37. Jayasinghe, O.; Hemachandra, S.; Anhetigama, D.; Kariyawasam, S.; Rodrigo, R.; Jayasekara, P. Ceymo: See more on roads—a novel benchmark dataset for road marking detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 3104–3113.
38. Torralba, A.; Russell, B.C.; Yuen, J. Labelme: Online image annotation and applications. *Proc. IEEE* **2010**, *98*, 1467–1484. [[CrossRef](#)]
39. Chen, L.C.; Hermans, A.; Papandreou, G.; Schroff, F.; Wang, P.; Adam, H. Masklab: Instance segmentation by refining object detection with semantic and direction features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4013–4022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.