Article

Interactive Efficient Multi-Task Network for RGB-D Semantic Segmentation

Xinhua Xu 1,2, Jinfu Liu 1,* and Hong Liu 1

1 Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Shenzhen 518055, China; xuxh55@mail2.sysu.edu.cn (X.X.); hongliu@pku.edu.cn (H.L.)
2 School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 518107, China
* Correspondence: liujinfu@stu.pku.edu.cn

Abstract: Semantic segmentation is significant for robotic indoor activities. However, relying solely on RGB modality often leads to poor results due to limited information. Introducing other modalities can improve performance but also increases complexity and cost, making it unsuitable for real-time robotic applications. To address the balance issue of performance and speed in robotic indoor scenarios, we propose an interactive efficient multitask RGB-D semantic segmentation network (IEMNet) that utilizes both RGB and depth modalities. On the premise of ensuring rapid inference speed, we introduce a cross-modal feature rectification module, which calibrates the noise of RGB and depth modalities and achieves comprehensive cross-modal feature interaction. Furthermore, we propose a coordinate attention fusion module to achieve more effective feature fusion. Finally, an instance segmentation task is added to the decoder to assist in enhancing the performance of semantic segmentation. Experiments on two indoor scene datasets, NYUv2 and SUNRGB-D, demonstrate the superior performance of the proposed method, especially on the NYUv2, achieving 54.5% mIoU and striking an excellent balance between performance and inference speed at 42 frames per second.

Keywords: semantic segmentation; RGB-D semantic segmentation; efficient semantic segmentation; attention mechanism

1. Introduction

In recent years, improving robot perception capabilities [1,2] has become a hotspot in various applications. Recent research employs sensors such as lasers or LIDAR [3,4], but the absence of visual information limits optimal performance. Consequently, semantic segmentation [5], which effectively utilizes visual data, has gained prominence. It aims to classify images at the pixel level, assigning each pixel to specific categories. With the emergence of deep learning, an increasing number of high-performance models [6–9] have shown remarkable results, gradually replacing traditional semantic segmentation methods. However, in complex indoor scenes, relying solely on the RGB modality is insufficient for refined segmentation due to the lack of object boundary and geometric information. However, depth data, which contains rich position, contour, and geometric information, is easily collected using RGB-D sensors such as Microsoft Kinect, making it an excellent modality. Thus, various approaches for RGB-D semantic segmentation [10–12] have been studied. Nonetheless, these methods possess limited interaction within cross-modality features and often ignore the impact of noise of both modalities.

Moreover, most RGB-D semantic segmentation works focus on feature fusion [13–15] between RGB and depth modality. Although these methods provide instructive fusion modules to integrate the two kinds of information, fully utilizing both modalities for detailed semantic segmentation remains a challenge.

On the other hand, the increased complexity introduced by depth information substantially raises inference time, making it challenging for real-time applications. In semantic...
segmentation, numerous studies [16–18] employ different techniques to minimize computational costs. However, for RGB-D semantic segmentation, most methods do not focus on the balance issue of performance and speed, causing slow inference speed.

To solve the above problems, we propose an interactive efficient multi-task RGB-D semantic segmentation network, IEMNet. By introducing a cross-modality feature rectification module [15], feature interaction and rectification are increased by leveraging channel-wise and spatial-wise correlations. In addition, we also propose a coordinate attention fusion module for capturing long-range interactions and achieving comprehensive feature fusion. Meanwhile, lightweight and efficient architecture [19] is utilized to ensure efficient inference speed. Without affecting inference speed, we also use a multi-task decoder [19], training the model with semantic segmentation and instance segmentation tasks simultaneously, further enhancing semantic segmentation performance. Evaluations of two indoor scene datasets NYUv2 [20] and SUNRGB-D [21] demonstrate that IEMNet achieves an optimal balance between performance and speed compared to other advanced methods.

2. Related Work

RGB-D semantic segmentation. Incorporating depth modality in RGB modality can increase segmentation performance. CNN-based RGB-D semantic segmentation has achieved great advances in recent years. Various methods have been proposed for higher accuracy in different aspects. CANet [10] employed a three-branch encoder comprising RGB, deep, and mixed branches, effectively complementing information via a co-attention module. PGDENet [11] utilized depth data with a dual-branch encoder and depth enhancement module. FRNet [12] proposed cross-level enriching and cross-modality awareness modules to boost representative information and obtain rich contextual information. FSFNet [13] introduced a symmetric cross-modality residual fusion module. CEN [14] proposed a novel feature fusion mechanism that replaces the current channel feature with a feature on the same channel from the other modality according to the scale factor of the batch normalization layer. CMX [15] performed a feature fusion module with a cross attention mechanism. Shapeconv [22] introduced a shape-aware convolutional layer to utilize the shape information from depth modality. SGNet [23] proposed spatial information-guided convolution to help integrate the RGB feature and 3D spatial information. The SA-Gate [24] employed a separation-and-aggregation gating operation for jointly filtering and recalibrating both modalities. Different from the previous works, we improve feature interaction and rectification with spatial and channel correlation and propose a coordinate attention fusion module to integrate features from RGB and depth modality.

Efficient semantic segmentation. The objective of efficient semantic segmentation is to achieve high-quality segmentation accuracy with low computational cost. Several efficient segmentation networks have been reported to satisfy this requirement. MiniNet [16] introduced a multi-dilated depthwise convolution with fewer parameters and better performance. WFDCNet [17] proposed a depthwise factorized convolution to separate dimension and improve computational efficiency. FANet [18] presented a fast attention module and manually added a downsampling layer in ResNet to reduce the computational cost. FRNet [25] used an asymmetric encoder–decoder architecture with factorized and regular blocks, making a trade-off between accuracy and speed. DSANet [26] employed a channel split and shuffle to reduce the computation and maintain higher segmentation accuracy. In this work, we utilize a shallow backbone and a simple architecture to ensure efficient inference speed.

3. Proposed Method

3.1. Overview

We propose an Interactive Efficient Multi-Task RGB-D Semantic Segmentation Network (IEMNet) following the EMSANet [19], as shown in Figure 1. The network extracts RGB and depth information through two encoder branches with ResNet as backbone, and replaces each $3 \times 3$ convolution of the original ResNet with a Non-Bottleneck-1D-Block.
(NBt1D) to improve performance. The NBt1D consists of a $3 \times 1$ convolution layer, a ReLU activation function, and a $1 \times 3$ convolution layer.

Moreover, the Cross-Modal Feature Rectification Module (CM-FRM) [15] is introduced and placed between two encoder branches to integrate and rectify information from both the modalities. In this way, the interacted and rectified features will be fed into the next stage to further promote feature extraction. Meanwhile, we also propose the coordinate attention fusion module (CAFM) to replace the RGB-D fusion module in EMSANet, enabling more comprehensive feature fusion through the coordinate attention mechanism, merging two feature maps into one, and sending it to the decoder via the skip connection.

Our decoder consists of two branches [19]: one for semantic segmentation and the other for instance segmentation. The semantic branch includes three decoder modules, a $3 \times 3$ convolution layer, and two learned upsampling blocks. It incorporates shallow structure with skip connections and applies $4 \times$ upsampling to reduce computational cost. Each decoder module contains a $3 \times 3$ convolution layer, three NBt1D blocks, and a learned upsampling block. Additionally, we calculate loss on a multi-scale to improve performance and training effectiveness. The instance branch is essentially the same as the semantic branch, except for the task heads. Instance segmentation results are represented by 2D gaussian heatmaps encoded from their centers and offset vectors pointing from each pixel to the corresponding instance center in the x and y directions.

### 3.2. Cross-Modal Feature Rectification Module

To address noise rectification and facilitate feature interaction, we introduce the Cross-Modality Feature Rectification Module (CM-FRM) [15], which calibrates features from both modalities at different stages in the encoder. The detailed structure of the CM-FRM is depicted in Figure 2, where it performs feature rectification in both channel and spatial dimensions by leveraging information from the other modality.

**Channel-wise feature rectification.** Given the features of two modalities $\text{RGB}_{\text{in}} \in \mathbb{R}^{H \times W \times C}$ and $\text{Depth}_{\text{in}} \in \mathbb{R}^{H \times W \times C}$, maximum and average pooling are first applied simultaneously across their channel dimensions, obtaining four output vectors. These vectors are concatenated and processed by a multi-layer perceptron (MLP) with a sigmoid activation function to obtain the channel weights, which are then split into $W^{c}_{\text{RGB}}$ and $W^{c}_{\text{Depth}}$.
and $W_{\text{Depth}}$. Then, they are multiplied with the input separately on channel dimension, obtaining the channel-wise rectificated features as follows:

$$\begin{align*}
\text{RGB}^C_{\text{rec}} &= W^C_{\text{Depth}} \otimes \text{Depth}_{\text{in}}, \\
\text{Depth}^C_{\text{rec}} &= W^C_{\text{RGB}} \otimes \text{RGB}_{\text{in}},
\end{align*}$$

where $\otimes$ denotes channel-wise multiplication.

**Figure 2. Cross-Modal Feature Rectification Module.**

**Spatial-wise feature rectification.** Given the same features of two modalities RGB\textsubscript{in} and Depth\textsubscript{in}, they are concatenated on channel dimension and processed by two convolutional layers and a ReLU function. A sigmoid activation function is then applied to obtain the feature map $F \in \mathbb{R}^{H \times W \times 2}$. Afterwards, $F$ is split into two spatial weight maps $W^S_{\text{RGB}}$ and $W^S_{\text{Depth}}$. Then, they are multiplied with the input separately on spatial dimension, obtaining the spatial-wise rectificated features as follows:

$$\begin{align*}
\text{RGB}^S_{\text{rec}} &= W^S_{\text{Depth}} \ast \text{Depth}_{\text{in}}, \\
\text{Depth}^S_{\text{rec}} &= W^S_{\text{RGB}} \ast \text{RGB}_{\text{in}},
\end{align*}$$

where $\ast$ denotes spatial-wise multiplication.

The final rectificated features of CM-FRM are obtained as follows:

$$\begin{align*}
\text{RGB}_{\text{out}} &= \text{RGB}_{\text{in}} + \lambda^C \text{RGB}^C_{\text{rec}} + \lambda^S \text{RGB}^S_{\text{rec}}, \\
\text{Depth}_{\text{out}} &= \text{Depth}_{\text{in}} + \lambda^C \text{Depth}^C_{\text{rec}} + \lambda^S \text{Depth}^S_{\text{rec}},
\end{align*}$$

where $\lambda^C$ and $\lambda^S$ are two hyperparameters, and both are set to 0.5 by default. Note that RGB\textsubscript{rec} and RGB\textsubscript{rec} are obtained by Depth\textsubscript{in} multiplying the corresponding weights.

Utilizing the information of the other modality, the CM-FRM can achieve comprehensive feature rectification and interaction.

### 3.3. Coordinate Attention Fusion Module

Inspired by [27], we propose a coordinate attention fusion module (CAFM) that receives the rectificated features and applies coordinate attention for feature fusion. As shown in Figure 3, CAFM consists of two stages, the coordinate attention stage and the fusion stage. Through the coordinate attention mechanism, our network is able to capture long-range interactions and know which features to pay more attention to and which features to suppress.

**Coordinate attention stage.** Given the rectificated features RGB\textsubscript{out} $\in \mathbb{R}^{H \times W \times C}$ and Depth\textsubscript{out} $\in \mathbb{R}^{H \times W \times C}$, spatially factorized adaptive pooling is first applied to both features to aggregate features along the two spatial directions, respectively, yielding four direction-aware feature maps. Then, we concatenate the feature maps on spatial dimension after...
the reshape operation and process the result with a $1 \times 1$ convolution and a non-linear activation function, yielding

$$Z = \delta(\text{Conv}_{1 \times 1}([\text{RGB}^b, \text{RGB}^w, \text{Depth}^b, \text{Depth}^w])), \quad (4)$$

where $[,]$ denotes the concatenation operation along the spatial dimension, $\delta$ is a non-linear activation function, and $Z$ is the intermediate feature map.

---

**Figure 3.** Coordinate Attention Fusion Module.

Then, $Z$ is split into two separate tensors $G^h$ and $G^w$. Another two $1 \times 1$ convolutions are utilized to separately transform $G^h$ and $G^w$ to tensors with the same channel number to the input. Then, we split $G^h$ and $G^w$ as attention weights for two modalities. They are multiplied with the input separately. The process is expressed as follows:

$$G^h_{\text{RGB}} G^w_{\text{RGB}} = \mathcal{F}_{\text{split}} \left( \sigma(\text{Conv}_{1 \times 1}(G^h)) \right),$$

$$G^w_{\text{RGB}} G^w_{\text{Depth}} = \mathcal{F}_{\text{split}} \left( \sigma(\text{Conv}_{1 \times 1}(G^w)) \right), \quad (5)$$

$$\text{RGB}_{\text{CA out}} = \text{RGB}_{\text{out}} \otimes G^w_{\text{RGB}} \otimes G^h_{\text{RGB}},$$

$$\text{Depth}_{\text{CA out}} = \text{Depth}_{\text{out}} \otimes G^w_{\text{Depth}} \otimes G^h_{\text{Depth}}, \quad (6)$$

where $\otimes$ denotes matrix multiplication.

Finally, $\text{RGB}_{\text{CA out}}$ and $\text{Depth}_{\text{CA out}}$ are concatenated on the channel dimension as the output of coordinate attention stage.

**Fusion stage.** In this stage, we use a simple channel embedding [15] to merge the features. Two $1 \times 1$ convolutions, a $3 \times 3$ depthwise convolution, and an intermediate ReLU activation function constitute the main path of the channel embedding. A $1 \times 1$ convolution forms the skip connection path. The concatenated features pass through these two paths separately and are then added to obtain the final output, which serves as the input for each stage of the decoder.

### 3.4. Loss Function

For semantic segmentation, we utilize the cross-entropy loss on a multi-scale, which is formulated as follows:

$$\mathcal{L}_{\text{sem}} = \frac{1}{4} \sum_{i=1}^{4} \mathcal{L}_{\text{CE}}(\mathcal{S}_{\text{out}}^i, \mathcal{S}_{\text{gt}}^i), \quad (7)$$

where $\mathcal{S}_{\text{out}}^i$ and $\mathcal{S}_{\text{gt}}^i$ denote semantic predictions and labels in a different scale.

For instance segmentation, Mean Squared Error (MSE) loss and L1 loss are used for center and offset supervision, respectively. The instance loss is computed as follows:
\[
L_{\text{ins}} = \frac{1}{4} \lambda_{\text{MSE}} \sum_{i=1}^{4} L_{\text{MSE}}(C_{\text{out}}^i, C_{\text{gt}}^i) + \\
\frac{1}{4} \lambda_{\text{L1}} \sum_{i=1}^{4} L_{\text{L1}}(O_{\text{out}}^i, O_{\text{gt}}^i),
\]

where \(C_{\text{out}}^i\) and \(C_{\text{gt}}^i\) denote center predictions and labels in different scales, \(O_{\text{out}}^i\) and \(O_{\text{gt}}^i\) denote offset predictions and labels in different scales, \(\lambda_{\text{MSE}}\) and \(\lambda_{\text{L1}}\) are the loss weight of the MSE loss and L1 loss, and we set them as 2 and 1 as default.

The total loss is organized as:

\[
L = \lambda_{\text{sem}} L_{\text{sem}} + \lambda_{\text{ins}} L_{\text{ins}},
\]

where \(\lambda_{\text{sem}}\) and \(\lambda_{\text{ins}}\) are hyperparameters, denoting task weights of semantic and instance segmentation.

4. Experiment

4.1. Datasets and Evaluation Measures

We evaluate the proposed method on two indoor datasets, NYUv2 and SUNRGB-D. NYUv2 [20] contains 1449 indoor RGB-D images with detailed annotations for semantic and instance segmentation. The standard dataset split is adopted, 795 images for training and the remaining 654 images for testing. SUNRGB-D [21] integrates multiple indoor RGB-D datasets, including NYUv2, with a total of 10,335 indoor images, annotated in 37 classes. Of these, 5285 images are used for training, and the remaining 5050 images are used for testing. The instance annotations of SUNRGB-D are extracted using 3D bounding boxes and semantic annotations [19]. Moreover, we use the Hypersim [28] dataset for pretraining. Hypersim is a realistic synthetic dataset, including 77,400 samples with masks for semantic and instance segmentation. The high-quality and large number of samples make it highly suitable for pretraining. We evaluated the proposed IEMNet and existing SOTA methods in terms of the mean intersection over union (mIoU), pixel accuracy (PA) [29], and frames per second (FPS).

4.2. Implementation Details

Our model was implemented based on Pytorch, and all training and testing processes were completed on the RTX A5000 GPU. During the experiments, the pre-trained weights from [19] on ImageNet were used to initialize the encoders of the two branches, and the model was trained for 500 epochs with a batch size of 8. We used the AdamW [30] optimizer with a weight decay of 0.01 for training, while the initial learning rate was set to \(6 \times 10^{-5}\) and \(4 \times 10^{-5}\) for NYUv2 and SUNRGB-D, and the poly learning rate schedule was applied to further adjust the learning rate. For data augmentation, random scaling, random cropping, and random flipping were employed, and for RGB images, slight color jitter augmentation was added in the HSV color space.

4.3. Quantitative Results on NYUv2 and SUNRGB-D

Table 1, Figures 4 and 5 show the quantitative results of the proposed method and other methods on NYUv2 and SUNRGB-D, including the results after pretraining on the Hypersim and fine-tuning on the corresponding datasets. Task weights \(\lambda_{\text{sem}}\) and \(\lambda_{\text{ins}}\) were set as 1:3 for NYUv2 and 1:2 for SUNRGB-D, determined through extensive experiments. The pretraining process uses only 20% of Hypersim data in each epoch to accelerate training, with an initial learning rate of \(6 \times 10^{-5}\) and task weights of 1:2 for semantic and instance segmentation. Fine-tuning was performed on NYUv2 and SUNRGB-D, with the same task weights and learning rate. FPS is calculated on NYUv2.
Table 1. Quantitative results on NYUv2 and SUNRGB-D.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>NYUv2</th>
<th>SUNRGB-D</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PA (%)</td>
<td>mIoU (%)</td>
<td>PA (%)</td>
</tr>
<tr>
<td>ShapeConv * [22]</td>
<td>ResNext101 32x8d</td>
<td>76.4</td>
<td>51.3</td>
<td>-</td>
</tr>
<tr>
<td>ShapeConv * [22]</td>
<td>Res101</td>
<td>-</td>
<td>-</td>
<td>82.2</td>
</tr>
<tr>
<td>SGNet * [23]</td>
<td>Res101</td>
<td>76.8</td>
<td>51.1</td>
<td>82.0</td>
</tr>
<tr>
<td>SA-Gate [24]</td>
<td>Res50</td>
<td>77.9</td>
<td>52.4</td>
<td>82.5</td>
</tr>
<tr>
<td>CMX * [15]</td>
<td>SegFormer-B5</td>
<td>80.1</td>
<td>56.9</td>
<td>83.8</td>
</tr>
<tr>
<td>ESANet (pre. SceneNet) [31]</td>
<td>Res34Nt1D</td>
<td>-</td>
<td>51.6</td>
<td>-</td>
</tr>
<tr>
<td>EMSANet (pre. Hypersim) [19]</td>
<td>Res34Nt1D</td>
<td>78.1</td>
<td>53.3</td>
<td>81.9</td>
</tr>
<tr>
<td>Ours (Semantic only)</td>
<td>Res34Nt1D</td>
<td>76.7</td>
<td>50.3</td>
<td>82.0</td>
</tr>
<tr>
<td>Ours</td>
<td>Res34Nt1D</td>
<td>76.8</td>
<td>51.3</td>
<td>81.9</td>
</tr>
<tr>
<td>Ours (pre. Hypersim)</td>
<td>Res34Nt1D</td>
<td>78.8</td>
<td>54.5</td>
<td>82.3</td>
</tr>
</tbody>
</table>

*: additional test-time augmentation, N/A: no implementation available, pre. SceneNet: additional pretraining on SceneNet, pre. Hypersim: additional pretraining on Hypersim.

Figure 4. Quantitative results on NYUv2.

Figure 5. Quantitative results on SUNRGB-D.

It can be observed that introducing the instance segmentation task significantly improves performance on both datasets compared to using only semantic segmentation. Combining multiple tasks does not increase the computational cost during testing, as only semantic results are predicted. Meanwhile, our method gets further improvement after pretraining on Hypersim without increasing the inference cost. Using data several times larger than the original training set for pretraining achieves a huge improvement for our method, as the impact of the data on the model performance is significant. However, our
method still achieves enhancement on this basis, which can be seen from the comparison with EMSANet and subsequent ablation experiments. In terms of mIoU comparison, our method ranks just behind CMX, but our inference speed is much faster, reaching 42.97 FPS. In terms of FPS comparison, our method is only slower than ESANet and EMSANet, but the mIoU is higher, achieving 54.5% on NYUv2 and 49.1% on SUNRGB-D, surpassing most methods. Thus, compared to other methods, our IEMNet achieves the optimal balance between performance and inference speed.

4.4. Qualitative Results on NYUv2

We visualize five semantic segmentation results of ESANet, EMSANet, and our method on the NYUv2 dataset, as shown in Figure 6. Visually, the method proposed in this paper performs better in segmentation accuracy than the ESANet and EMSANet. For example, our method IEMNet correctly identifies the TV cabinet and the washbasin while the ESANet and EMSANet wrongly classify them with other classes. It demonstrates that our proposed method enhances the network’s ability to utilize depth information through more comprehensive feature interaction and fusion, resulting in better performance.

![Figure 6. Qualitative results on NYUv2.](image)

4.5. Ablation Study on NYUv2

To investigate the impact of different components on segmentation performance and inference speed, we conducted ablation studies using only semantic decoder on NYUv2 and SUNRGB-D. The same training strategy and various model settings, including the CM-FRM, CAFM, and different backbones, are employed as shown in Table 2, where the RGB-D Fusion module is the original fusion module in the EMSANet.
Table 2. Ablation for CM-FRM, CAFM and Backbone.

<table>
<thead>
<tr>
<th>Backbone</th>
<th>CM-FRM</th>
<th>Fusion</th>
<th>NYUv2 mIoU (%)</th>
<th>SUNRGB-D mIoU (%)</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res34NBt1D</td>
<td>×</td>
<td>RGB-D Fusion</td>
<td>49.20</td>
<td>46.81</td>
<td>46.18</td>
</tr>
<tr>
<td>Res34NBt1D</td>
<td>✓</td>
<td>RGB-D Fusion</td>
<td>50.27</td>
<td>47.88</td>
<td>43.26</td>
</tr>
<tr>
<td>Res34NBt1D</td>
<td>×</td>
<td>CAFM</td>
<td>49.41</td>
<td>47.11</td>
<td>45.28</td>
</tr>
<tr>
<td>Res34NBt1D</td>
<td>✓</td>
<td>CAFM</td>
<td>50.38</td>
<td>48.10</td>
<td>42.97</td>
</tr>
<tr>
<td>Res18</td>
<td>✓</td>
<td>CAFM</td>
<td>46.04</td>
<td>45.17</td>
<td>63.37</td>
</tr>
<tr>
<td>Res34</td>
<td>✓</td>
<td>CAFM</td>
<td>47.06</td>
<td>46.36</td>
<td>54.57</td>
</tr>
<tr>
<td>Res50</td>
<td>✓</td>
<td>CAFM</td>
<td>49.86</td>
<td>47.05</td>
<td>28.03</td>
</tr>
<tr>
<td>Res18NBt1D</td>
<td>✓</td>
<td>CAFM</td>
<td>46.74</td>
<td>46.09</td>
<td>55.21</td>
</tr>
</tbody>
</table>

Firstly, it can be seen that our CAFM outperforms RGB-D Fusion in fusing cross-modal features and achieving better results on both datasets. The significant improvement after introducing the CM-FRM demonstrates the effectiveness of enhanced feature interaction and calibration. Although using the CM-FRM and CAFM increases the inference time, the FPS remains at 42.97, achieving efficient speed. Note that the FPS is very slow when using Res50. This is because the block and number of layers are different with Res18 and Res34, causing a significant increase in parameter and computational complexity. Moreover, using NBt1D, PA, and mIoU is generally better, while the inference speed decreases. This discrepancy with [19,31,32] is due to their FPS calculations being performed on a robotic platform, while ours are on the Pytorch platform. Thus, NBt1D is more suitable for robotic platforms compared to normal convolution. Considering the performance and inference speed of various backbones, using Res34NBt1D as the backbone achieves better balance between performance and speed.

5. Conclusions

In order to achieve good balance between performance and speed for RGB-D semantic segmentation, we propose an interactive efficient multitask RGB-D semantic segmentation network. Based on the EMSANet, the cross-modal feature rectification module and coordinate attention fusion module are introduced into the encoder branches to accomplish feature rectification and fusion at different scales, constructing an interactive encoder structure that enables comprehensive feature interaction. Furthermore, without compromising the inference speed of the semantic segmentation task, our method incorporates an instance segmentation task, thereby enhancing the semantic segmentation performance. Results of the NYUv2 and SUNRGB-D datasets indicate that the proposed method exhibits optimal performance in balancing performance and speed under indoor scenes. In future work, we will port our method to robotic platforms to verify its performance. In addition, integrating more tasks into the current framework is expected to further improve performance, such as depth estimation.

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

Funding: This research was funded partly by the National Natural Science Foundation of China, grant number 62073004, partly by the Science and Technology Plan of Shenzhen, grant number JCYJ2020109140140340, and partly by the Shenzhen Fundamental Research Program, grant number GXWD20201231165807007-20200807164903001.

Data Availability Statement: Not applicable.

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


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.