Real-Time Edge Attention-Based Learning for Low-Light One-Stage Object Detection

Yen-Yu Pu, Ching-Te Chiu and Shu-Yun Wu
ke123tw@gmail.com, ctchiu@cs.nthu.edu.tw, shuyunijk@gmail.com
Department of communication engineering, National Tsing Hua University, Hsinchu, Taiwan

***Abstract*—Advances in Convolutional Neural Network (CNN) has spurred extensive research on computer vision as object detection is a very important part. Object detection can either be one stage, such as Single-Shot Multibox Detector (SSD) and You Look Only Once (YOLO), and two stage, such as the faster region-based CNN. Majority of these studies use RGB images. Insufficient lighting often cause detection errors, which we aimed to solve by combining depth information with RGB images.**

We used edge attention-based learning to extract the edge features from RGB and depth images. The extracted features were fused with the feature maps from the backbone. These features were enhanced using the enhanced feature block and upsample block for small objects. From the final layer feature maps of the RGB and depth paths, we adjust the weighting of the RGB and depth paths to fuse their features, which significantly improves the object detection performance. Finally, the results obtained by the weight fusion layer were combined with the edge images to adjust the confidence score according to the edge ratio and output the final result. We evaluated our method on the SUN RGB-D dataset.

I. Introduction

Object detection is one of the most important computer vision tasks. Recently, several object detection studies use deep learning methods; however, most of them can only perform detection RGB images. Therefore, detection errors are likely to occur in insufficiently light scenes. When there is insufficient lighting, the original object detection can only frame the location and correctly classify the larger objects. The pillows on the sofa cannot be detected. Fortunately, there are many devices, such as Microsoft Kinect and Intel Realsence, that can simultaneously capture the RGB and depth information. This can greatly improve object detection in many applications. Therefore, we aimed to exploit the depth information to enhance and improve the accuracy of object detection.

Our architecture was based on the one-stage object detection single-shot multibox detector (SSD)[1] architecture to maintain the advantage of speed and overcome the shortcomings of the original SSD object detection. In this study, we defined a small object as a default box whose length or width was between 30 and 60 pixels. To enhance the detection of small objects, the SSD detects small objects using lower-level feature maps. However, this may lead to insufficient information extraction or insufficient receptive fields.

Our goal was to improve the detection for small-scale objects in insufficiently lit scenes and make the inference time consistent with real time simultaneously. Thus, we provided information other than the RGB images, such as depth and edge information, to solve the challenge posed by insufficiently lit scenes. We also proposed different modules to solve the detection challenges posed by small-scale objects. We added depth-stream, which can significantly improve detection accuracy. Subsequently, we incorporated the enhanced feature block, upsample block, and edge feature extraction to enhance the features and optimise the parameters. Finally, we used the weight fusion layer to fuse the RGB and depth results; then, we used the edge-enhanced non-maximum suppression (NMS) to select the final result.

II. Related Work

A. Two-Stream Fusion Method

There are various methods of fusion information, which can be broadly divided into two categories. One is early fusion, such as [2], [3], [4]. Early fusion entails using CNNs to fuse different types of information before obtaining the final result. Although the use of early fusion may uncover unexpected features, finding the best way to fuse them is a challenge.

The other method is the late fusion [5], [6], [7]. It entails the use of the two stream fusion method to train different types of information and fuse a two-stream result to obtain the final result. The late fusion can be more flexible and accurate than the early fusion. However, this may increase the number of parameters significantly.

III. Proposed Architecture

The proposed method was composed of five major blocks: enhanced features block, upsample block, edge feature extraction, weight fusion layer, and edge-enhanced NMS. The overall architecture is shown in Fig.1; each block is explained in the following sections.

A. Enhanced Feature Block

We used the method proposed by Shih *et al.*[8] to extract multiple features. This method requires few parameters to extract features in different layers. As shown in Fig.2, we used the three layers in the Conv4 and Conv5 blocks (for example, conv4_1, conv4_2, conv4_3 and conv5_1,

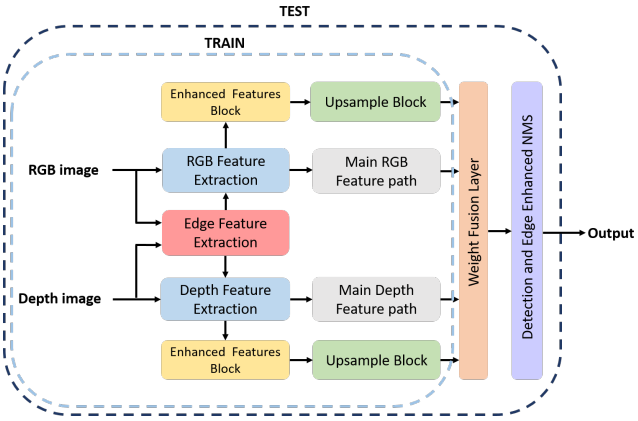

Figure 1: Overview of proposed architecture.

conv5_2, conv5_3) to extract useful information and provide more useful information to the following upsample block to improve accuracy.

In the first part, we used 1 × 1 convolution to reduce the number of channels of each input layer to avoid the problem of too many parameters caused by too many channels. To maintain the independence of information between each layer, we concatenated the information between each layer and passed it to the second part.

The main function of the second part was to extract the compressed information. Furthermore, to avoid the loss of the original information, we used shortcuts to concatenate the output of the last layer with the compressed information to obtain the final output of the enhanced feature block.

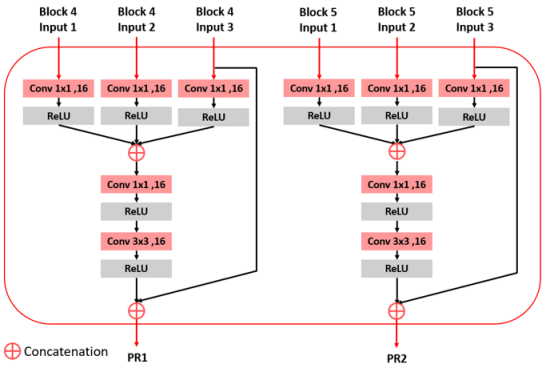

Figure 2: Architecture of enhanced feature block.

B. Upsample Block

We used the result of the previous enhanced feature block as the input of the upsample block; then, we used maxout[9] to compress the PR1(From Conv4 block) channel number map because it can fuse the more noticeable features of the feature map without increasing the number of parameters. For the PR2(from Conv5 block), we first used 1 × 1 convolution to compress the number of channels, following which we used deconv to expand the feature map. To improve the receptive field, we used dilated convolution.

Based on the multiple-path fusing results [10], we adopt element-wise point-to-point multiplications to fuse the RGB and depth paths. After the fusing, the receptive field was expanded using the dilated convolution layer. Therefore, we demonstrated the possibility of using the upsample block to solve the defects of the original SSD in detecting small objects.

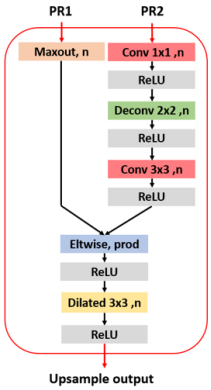

Figure 3: Architecture of upsample block.

C. Edge Feature Extraction

To obtain very abundant information, we first used the Canny algorithm to convert the RGB and depth images into edge images. Then, information was extracted through edge feature extraction, as shown in Fig.4.

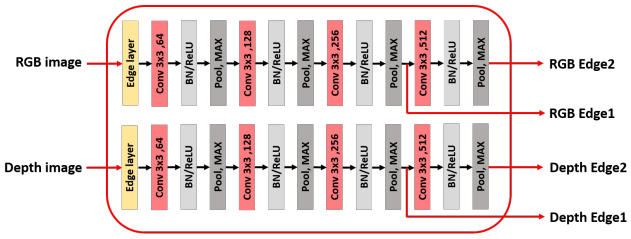

Figure 4: Architecture of edge feature extraction.

There are two paths in edge feature extraction. One is for processing the RGB boundaries, and the other is for processing the depth boundaries. First, we used the edge layer (canny algorithm) to convert the RGB and depth images to edge images. Then, each edge image was passed through four sets of convolution layers and batch normalisations to extract the boundary features. Finally, we output the feature maps as 38 × 38 (Edge1) and 19 × 19 (Edge2). Many difference features can be observed between the RGB, depth, and edge information.

Therefore, this method can be used to increase information richness.

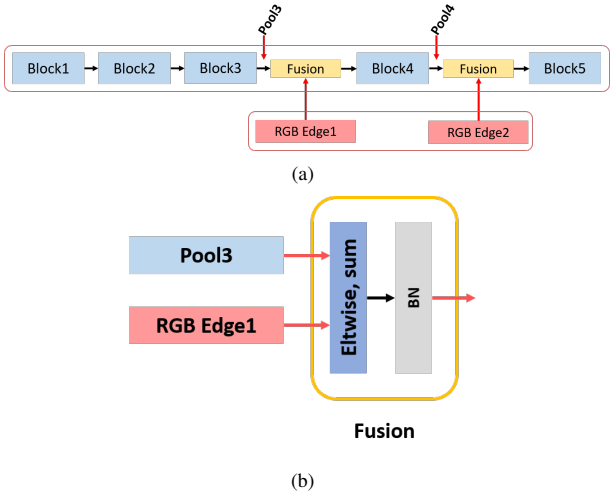

Figure 5: (a) Edge feature map and RGB feature map fusion method. (b) Details of fusion block.

Then, we fused the edge feature extraction feature, RGB feature, and depth feature maps through the element-wise (Eltwise) point-to-point addition, as shown in Fig.5. We used the Eltwise layer because it was necessary to maintain the number of channels. Thus, using the RGB, depth, and edge information made it possible to maintain the number of channels. Finally, we chose the Conv4 block and Conv5 block for fusion.

D. Weight Fusion Layer

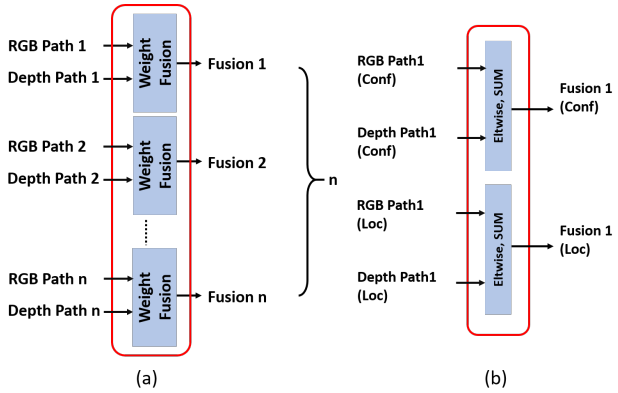

Figure 6: (a) Architecture of the weight fusion layer. (b) Weight fusion block architecture in the weight fusion layer.

In the weight fusion layer, the main goal was to fuse the RGB and depth information. We adjust the weighting of the RGB and depth paths to fuse their features. The summation of the RGB and depth weights are set to one. The initial weight of the RGB path and depth path are set to 0.5 respectively. The RGB and depth weights are adjusted to get the maximum accuracy. Fig.6(a) shows the weight fusion layer architecture. We merged the RGB and depth information based on the feature map size. RGB Path 1 comes from the output of the upsample block. RGB Path 2 to n are come from the outputs of main RGB feature block of layer 2 to n. The SSD detection method was divided into two paths: the first path, the conf path, was for classification, and the other, called the loc path, was for the regress bounding box. We multiplied them by the weight and summed up the corresponding depth conf path and loc path, as shown in Fig.6(b).

E. Edge-Enhanced NMS

When performing the final output detection, we included the edge images to enhance the confidence score of the object. We observed that the proportion of the edge of an object usually accounts for only less than 1% of the ground-truth bounding box area. Therefore, this feature could be used to adjust for the final confidence score.

Algorithm 1 Edge-enhanced NMS algorithm.

$B_{nms} \leftarrow B_n$ use NMS algorithm to delete bounding boxes with IOU greater than the threshold. Then, select 250 boxes with the highest score.
for All B_{nms} **do**
  if $Score > 0.1$ **then**
    $edge\ rate = \frac{1}{w \times h} \sum_{i=0}^{h} \sum_{j=0}^{w} edge\ point$;
    $edge\ function = 0.1 \times e^{(\frac{-edge\ rate^2}{200})}$;
    $score = score \times (1 + edge\ function)$;
  end if;
end for;
return 200 boxes with the highest score.

IV. Experimental Results

In this section, we describe several experiments conducted to demonstrate the ability of the proposed method to improve performance. We performed ablation studies to show the set-up of our architecture. At the end of this section, we compared our results with those of other studies.

A. Environment and Datasets

All experiments in this paper were implemented by Caffe. The proposed method was accelerated on an 11-GB NVIDIA RTX2080Ti GPU and 32-GB NVIDIA Tesla-V100-SXM2. The CPU was Intel Core i7-7800X 3.5GHz, and the main memory was 32GB DDR4 RAM. We used four different architectures as the backbone: VGG-16, Mobilenet, Densenet-169, and SE-ResneXt50. They were pretrained on the ImageNet dataset, which we used to initialize the backbone weights.

During training, the batch size was eight. The training iteration was set to 120k. The learning rate was 0.001, which was reduced by 0.1 at 80k and 100k iterations. With reference to the original paper[1], the input size was resized to 300 × 300 pixels. The proposed model was trained and tested on SUN RGB-D[11] which contains low-light images.

B. Results

We proposed different modules to solve the detection challenges. Table.I show the accuracy and parameter. Compared with before joining our module, our model can significantly improve the accuracy.

We used the enhanced feature block and edge feature extraction on the VGG16 backbone. However, we did not use these two modules on other backbones. We did not use the enhanced feature block on the other backbones because they already contained similar functions, such as the Densen block, residual block, or inverted residual block, which eliminated the problem of insufficient feature extraction. Thus, we could directly use the upsample block to fuse the low- and high-level features.

During edge feature extraction, we also attempted to use different backbones. However, they did not yield the best results. This is possibly attributable to their usual need to use pre-train models; thus, adding the edge information to the backbone does not significantly impact the final training effect.

However, the other proposed method had significant effects on the different backbones.

Table I: Comparison of adding different modules.

Method							mAP(%)	Para(M)
Backbone	Depth-stream	UB	EFB	EFE	WFL	EENMS		
VGG-16	×	×	×	×	×	×	34.6	26.28
VGG-16	√						43.2	51.21
VGG-16	√	√	√				43.55	54.27
VGG-16	√	√	√	√			43.86	57.37
VGG-16	√	√	√	√	√		46.56	57.37
VGG-16	√	√	√	√	√	√	46.58	57.37
Densenet-169	√	√	×	×	√	√	47.52	40.15
Mobilenet v3	√	√	×	×	√	√	43.9	18.19
SE-ResneXt-50	√	√	×	×	√	√	46.77	68.9

C. Comparison with Other Works on SUN RGB-D

In this section, we compare the results obtained using the proposed method and several state-of-the-art methods on the SUN RGB-D dataset. There were 5285 training and 5050 testing images. The results are shown in Table II. Our method with all different backbones are significantly outperformed [12], not only in terms of accuracy but also by achieving real-time detection. Our method with Densenet-169 achieves the highest accuracy and its frame rate is higher than that of [2] by 11fps.

V. Conclusion

In this study, we combined the depth-stream, enhanced feature block, upsample block, edge feature extraction, weight fusion block, and edge-enhanced NMS to improve the object detection performance for insufficiently lit scenes and small-scale objects. Compared with the original SSD, including the depth-stream improved the accuracy from 34.63% to 43.2%, an improvement of 8.57% on the SUN-RGB-D dataset. The combination of the enhanced features block and upsample block improved the accuracy by 0.35%, can be improved from 43.2% to 46.55%, on the SUN-RGB-D dataset. Further, the combination of edge feature extraction, weight fusion block, and edge-enhanced NMS improved the accuracy by 3.03%, from 43.55% to 46.58%, on the SUN-RGB-D dataset. Our method can not only improve the VGG-16 backbone but also improve other backbones, such as Densenet-169, Mobilenetv3, and SE-ResneXt50. On Densenet-169, we obtained the best accuracy, at 47.52% with 40.15M parameters. On Mobilenetv3, we obtained an accuracy of 43.92% with 18.19M parameters. In other words, the proposed method can reach real-time detection, at 33FPS. On SE-ResneXt50, we obtained an accuracy of 46.77% with 68.9M parameters.

Table II: Comparison with other works on SUN RGB-D.

Method	Backbone	mAP(%)	Para(M)	FPS
[1] SSD RGB	VGG-16	34.63	26.28	46
[1] SSD depth	VGG-16	33.71	26.28	46
[13](CVPR'14)	-	32.94	-	-
[14](Neurocomputiog'18)	-	35.2	-	-
[15](ICASSP'20)	RGB:pruning-VGG16 Depth:pruning-VGG16	35.71	45.59	8
[12](CVPR'16) Baseline	RGB:VGG-16 Depth:Alexnet	43.8	206.97	<3
[2](TIP'19)	RGB:VGG-16 Depth:Alexnet	47.5	-	<2
Our method	RGB:VGG-16 Depth:VGG-16	46.58	57.38	27
Our method	RGB:Densenet-169 Depth:Densenet-169	**47.52**	40.15	13
Our method	RGB:Mobilenet v3 Depth:Mobilenet v3	43.92	**18.19**	33
Our method	RGB:SE-ResneXt-50 Depth:SE-ResnetXt-50	46.77	68.9	17

References

[1] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[2] G Li, Y Gan, H Wu, N Xiao, and L Lin, "Cross-modal attentional context learning for rgb-d object detection.," *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, vol. 28, no. 4, pp. 1591, 2019.

[3] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Asian conference on computer vision*. Springer, 2016, pp. 213–228.

[4] Seong-Jin Park, Ki-Sang Hong, and Seungyong Lee, "Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4980–4989.

[5] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[6] Enqing Chen, Xue Bai, Lei Gao, Haron Chweya Tinega, and Yingqiang Ding, "A spatiotemporal heterogeneous two-stream network for action recognition," *IEEE Access*, vol. 7, pp. 57267–57275, 2019.

[7] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[8] Kuan-Hung Shih, Ching-Te Chiu, Jiou-Ai Lin, and Yen-Yu Bu, "Real-time object detection with reduced region proposal network via multi-feature concatenation," *IEEE transactions on neural networks and learning systems*, 2019.

[9] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, "Maxout networks," in *International conference on machine learning*, 2013, pp. 1319–1327.

[10] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg, "Dssd : Deconvolutional single shot detector.," *CoRR*, 2017.

[11] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[12] Saurabh Gupta, Judy Hoffman, and Jitendra Malik, "Cross modal distillation for supervision transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2827–2836.

[13] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *European conference on computer vision*. Springer, 2014, pp. 345–360.

[14] Saihui Hou, Zilei Wang, and Feng Wu, "Object detection via deeply exploiting depth information," *Neurocomputing*, vol. 286, pp. 58–66, 2018.

[15] Jiou-Ai Lin, Ching-Te Chiu, and Yen-Yu Cheng, "Object detection with color and depth images with multi-reduced region proposal network and multi-pooling," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1618–1622.