Article

Confidence-Aware Object Detection Based on MobileNetv2 for Autonomous Driving

Wei Li and Kai Liu *

School of Computer Science and Technology, Xidian University, Xi'an 710071, China; wli_6@stu.xidian.edu.cn

* Correspondence: kailiu@mail.xidian.edu.cn

Abstract: Object detection is an indispensable part of autonomous driving. It is the basis of other high-level applications. For example, autonomous vehicles need to use the object detection results to navigate and avoid obstacles. In this paper, we propose a multi-scale MobileNeck module and an algorithm to improve the performance of an object detection model by outputting a series of Gaussian parameters. These Gaussian parameters can be used to predict both the locations of detected objects and the localization confidences. Based on the above two methods, a new confidence-aware Mobile Detection (MobileDet) model is proposed. The MobileNeck module and loss function are easy to conduct and integrate with Generalized-IOU (GIOU) metrics with slight changes in the code. We test the proposed model on the KITTI and VOC datasets. The mean Average Precision (mAP) is improved by 3.8 on the KITTI dataset and 2.9 on the VOC dataset with less resource consumption.

Keywords: artificial neural networks; deep learning; object detection

Citation: Li, W.; Liu, K. Confidence-Aware Object Detection Based on MobileNetv2 for Autonomous Driving. *Sensors* **2021**, *21*, 2380. <https://doi.org/10.3390/s21072380>

Academic Editor: Felipe Jiménez

Received: 20 February 2021

Accepted: 25 March 2021

Published: 30 March 2021

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

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

1. Introduction

Object detection is a basic task in computer vision. Many object detection methods have been proposed in the past few decades. Early methods mainly rely on handcrafted features and shallow machine learning algorithms, which are easy to overfit. As a result, the performance is not satisfactory. In recent years, deep artificial neural networks have been emerged and become the dominant methods in object detection problems.

Deep artificial neural networks for object detection can be roughly divided into two categories: one-stage models and two-stage models. Two-stage models have better performance, but they consume more computational resources. The one-stage models consume less resources, but their performance is not as good as that of the two-stage models. Both one-stage models and two-stage models can be used for autonomous driving and for different purposes. For two-stage models, they can be deployed on higher performing hardware. Some one-stage models can be deployed on mobile devices. In this paper, we propose Conf-MobileDet, a Confidence-aware one-stage object Detection model based on the MobileNetv2 [1] backbone for autonomous driving.

Object detection for autonomous driving involves accurate detection ability in traffic scenes. Objects to be detected mainly consist of vehicles, pedestrians, and cyclists. In addition, objects in traffic scenes are often very dense and of various sizes. Many objects are occluded by others. In some cases, objects used for training are not well labeled. These phenomena inevitably make the model learn ambiguous features and decrease the localization confidence. Traditional object detection models output definite detected object positions, the objectness score, and their classes. These outputs cannot quantify the positioning confidence.

To address these problems, we propose a MobileNeck module and use a cross-entropy loss function for learning bounding box regression and localization confidence based on Generalized-IOU (GIOU) [2] loss. Specifically, to obtain the confidence of bounding box prediction, we model the bounding box prediction and bounding box labels as Gaussian distributions and Dirac delta functions, respectively. The new loss can be defined by a

KL divergence with respect to the distribution of the bounding box prediction and the bounding box labels. The loss function is equal to a cross-entropy loss function. We use this cross-entropy loss function to model the confidence of the bounding box prediction. Moreover, we combine the proposed loss function with the truncated MobileNetv2 backbone and MobileNeck module to test the proposed methods.

The contributions of our work are as follows:

1. We designed a multi-scale MobileNeck module that was used to further extract higher dimensional features and enlarge the effective receptive field [3].
2. We used a cross-entropy loss function for learning bounding box regression and localization confidence based on the GIoU loss function.
3. We applied the proposed methods to the MobileNetv2 backbone. Experiments were performed on the KITTI and Pascal VOC datasets, which showed that the proposed model achieved better performance compared to the original methods.

2. Related Work

Many object detection models have been proposed for autonomous driving. Traditional methods rely on hand-crafted features and shallow machine learning algorithms, which easily overfit. As a result, the performance is not promising. Convolutional neural networks (CNNs) extract object features autonomously and simultaneously use a linear classifier to predict the object locations and classes. The result outperforms traditional methods by a large margin.

CNNs for object detection can be approximately divided into two classes: one-stage models and two-stage models. Two-stage models generate a sparse object candidate set in the first stage and predict objects on the sparse set in the second stage. The representative works of two-stage models include R-CNN [4] and their descendants, such as Fast R-CNN [5], Faster R-CNN [6], Mask R-CNN [7], and Cascade R-CNN [8].

Different from two-stage models, one-stage models directly predict object locations and classes based on predefined dense anchor boxes and treat the object detection task as a regression problem. The representative works of one-stage models include MobileNet [1,9,10], SqueezeDet [11], Single Shot Multibox Detector (SSD) [12], YOLO [3,13–15], EfficientDet [16], GhostNet [17], and their descendants.

Our purpose is to design an object detection model that can be deployed on mobile devices. MobileNet and its descendants are designed for these tasks. MobileNetv2 is a recent representative object detection model, so our work is based on MobileNetv2.

In addition to carefully designed models for mobile devices, some researchers are working toward the design of various loss functions for their specific use. One-stage object detection models predict thousands of bounding boxes and corresponding classes in which many of them belong to the negative samples. The contribution from positive samples to the loss function is easily overwhelmed by that of negative samples. SSD resolves this problem by heuristic hard negative mining. RetinaNet [18] resolves this problem by modifying the traditional cross-entropy loss function. They propose focal loss to tackle the imbalance of positive samples and negative samples. In RetinaNet, focal loss can filter out easy examples, leaving hard samples to train a discriminative network. As a result, the negative effects introduced by the imbalance of positive samples and negative samples are resolved. In this work, we use focal loss for classification. The proposed methods can work well together with focal loss.

The traditional loss function for bounding box regression is the smooth L_1 loss. It independently regresses the central point coordinates, width, and height with respect to the predefined anchor boxes. In addition, the actual evaluation target is the Intersection over Union (IoU). They are not absolutely equivalent. To compensate for the gap between the smooth L_1 loss and IoU, Zhou et al. proposed IoU loss [19], which regresses the offset between the location of the object's center and the center of an anchor box and then uses the width and height of the anchor box to predict the relative scale of the predicted object boxes. Apart from IoU loss, Generalized-IoU (GIoU) loss [2], Distance-IoU (DIoU) loss [20],

and Complete-IoU (CIoU) loss [20] were proposed, which can be seen as extending IoU loss. In this work, we used the proposed methods to improve the performance of GIoU loss. It is easy to conduct, and the modification can work well together with the original GIoU loss. Since the objectiveness score is used to reflect the possibility of whether an object exists, it cannot reflect the quality of the predicted bounding boxes well. We need better means to mend the deficiency. This problem is resolved by bounding box confidence-based Non-Maximum Suppression (NMS).

Our work is related most to [21,22], which output Gaussian parameters for bounding box regression. However, there were three differences.

1. During the design of the proposed model, we considered that the model should be applicable to mobile devices. Based on this prerequisite, we designed the light-weight multiscale MobileNeck network. It was used to further extract higher dimensional features and enlarge the effective receptive field.
2. The proposed model contains fewer multiply-add operations compared to [21,22].
3. We illustrate the generated unconfidence degree in the object detection results. In addition, explanations are added to describe the rationality of proposed methods.

3. Materials and Methods

In this section, first, we describe the Conf-MobileDet architecture. Next, we propose the bounding box parameterization and the cross-entropy loss-based bounding box confidence prediction. Finally, we introduce a confidence-based NMS approach.

3.1. Model Architecture

The proposed model is composed of four parts: MobileNetv2 backbone, Spatial Pyramid Pooling (SPP) layer, MobileNeck layers, and object detection layers. The model is shown in Figure 1.

Figure 1. The proposed model. SPP, Spatial Pyramid Pooling; Conf-aware, Confidence-aware; Det, detection.

We used the truncated MobileNetv2 model as our backbone network. The basic MobileNetv2 backbone was built on a depthwise separable convolution layers and pointwise convolution layers, which made the backbone contain fewer multiply-add operations during each forward propagation. The layer parameters are in Table 1. It was pretrained on ImageNet and used to extract high-dimensional features for each input image.

The SPP layer takes the output of the truncated MobileNetv2 model. It pools feature maps of various sizes and outputs feature maps that have the same sizes in spatial dimensions. Based on SPP, the proposed model can take images of different sizes during both the training and evaluation stages.

Since MobileNeck is built with depthwise separable convolutions and pointwise convolutions and is located between the backbone and the detection head, we call it MobileNeck. Compared to SSDNeck layers built with conventional convolution layers (Figure 2), MobileNeck imposes a lower computational burden. We used upsampling and skip connections to strengthen the semantic information in the first detection branch.

Table 1. Parameters of truncated MobileNetv2 used as the backbone of the proposed model.

Input	Operator	t	c	n	s
$224^2 \times 3$	Conv2d	-	32	1	2
$112^2 \times 32$	Bottleneck	1	16	1	1
$112^2 \times 16$	Bottleneck	6	24	2	2
$56^2 \times 24$	Bottleneck	6	32	3	2
$28^2 \times 32$	Bottleneck	6	64	4	2
$28^2 \times 64$	Bottleneck	6	96	3	1
$7^2 \times 160$	Bottleneck	6	160	3	2

Figure 2. MobileNetv2 with the SSDdetector.

The detection layer of the proposed model is confidence-aware. It is responsible for predicting the confidence of each output with bounding box coordinates, objectness scores, and classification scores. As a result, the dimension of each output is $(c + 8 + 1) * n$ compared to conventional $(c + 4 + 1) * n$, where c is the number of classes in the dataset and n is the number of anchors at each pixel on the output feature maps. It is $4 * n$ longer than conventional outputs. These additional outputs are used to predict the confidence scores of each bounding box.

3.2. Model Parameters

Figure 1 shows the proposed model. It regresses the two-dimensional top-left point (x_c, y_c) and width, height (w, h) of a bounding box independently.

$$\begin{aligned}
 b_x &= 2\sigma(t_x) - 0.5 + c_x, \\
 b_y &= 2\sigma(t_y) - 0.5 + c_y, \\
 b_w &= p_w(2\sigma(t_w))^2, \\
 b_h &= p_h(2\sigma(t_h))^2.
 \end{aligned} \tag{1}$$

where b_* denotes the predicted bounding boxes, t_* is the output of the proposed network, c_* is the offset of a cell from the top-left corner of the feature map, and p_* is the width or height of the predefined anchor. Different from [21], in which the offsets of c_x are confined within the boundaries of their grids and may lead to the grid sensitivity phenomenon [3], we used the metrics described in Equation 1 to prevent the problem [3]. In the equation, the output of the network t_* is multiplied by a factor that is greater than one. This allows the offset of predicted bounding boxes to be larger than its original grid. Figure 3 shows the bounding box regression process.

Figure 3. Bounding box regression in the proposed model.

The objective is to estimate the confidence score of the predicted location. Similar to classification score prediction, the network needs to predict a probability distribution along with a definite object location. Because the predicted central point coordinates, box width, and height are independent, we can use a single-variate Gaussian model for simplicity:

$$f_{\Theta}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2)$$

where Θ is the trainable network parameter set, μ is the predicted bounding box location, and σ stands for the degree of unconfidence for the predicted bounding box location. The bounding box confidence can be formulated as $conf = 1 - \sigma$, ($0 < \sigma < 1$). For $conf \rightarrow 1$, the network becomes extremely confident about the predicted bounding box location. μ and σ are the outputs of the last layer of the network. They are produced by a convolutional layer. Figure 1 shows the proposed architecture.

The bounding box labels can also be formulated as a Gaussian distribution with $\sigma \rightarrow 0$, which is a delta function:

$$\delta(x - x_g) = \begin{cases} 1 & x = x_g, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where x_g denotes the bounding box labels. The relationship between Equations (2) and (3) and the model parameters is shown in Figure 4.

Figure 4. Gaussian model and its parameters used for bounding box regression.

3.3. Bounding Box Regression with Cross-Entropy Loss

We used the cross-entropy loss function to model the localization confidences of predicted bounding boxes over a single sample:

$$\begin{aligned}
L_{reg} &= H(p_D(x), p_{\Theta}(x)) \\
&= - \int p_D(x) p_{\Theta}(x) dx \\
&= - \int \delta(x - x_g) f_{\Theta}(x) dx \\
&= \frac{(x_e - x_g)^2}{2\sigma^2} + \frac{\log(\sigma^2)}{2} + \frac{\log(2\pi)}{2}
\end{aligned} \tag{4}$$

where $H(*, *)$ is cross-entropy loss function; $p_D(x)$ is equal to $\delta(x - x_g)$ and denotes the ground truth of the object's coordinates; $p_{\Theta}(x)$ is equal to $f_{\Theta}(x)$, and it is a Gaussian distribution based on the output of the Gaussian parameters. By deriving the formula, we get $L_{reg} = \frac{(x_e - x_g)^2}{2\sigma^2} + \frac{\log(\sigma^2)}{2} + \frac{\log(2\pi)}{2}$. Because $\frac{\log(2\pi)}{2}$ is a constant, the loss function can be rewritten in a simple form:

$$L_{reg} = \frac{(x_e - x_g)^2}{2\sigma^2} + \frac{\log(\sigma^2)}{2} \tag{5}$$

If $|(x_e - x_g)|$ is large, the network is unconfident about the prediction. In this condition, we expect the network outputs to be large σ , so that the loss will become small (Figure 4). For $\sigma^2 = 1$, $\frac{\log(\sigma^2)}{2}$ is equal to 0. The cross-entropy loss degenerates to the normal Euclidean bounding box regression loss:

$$L_{reg} = (x_e - x_g)^2 \tag{6}$$

As a result, the proposed model is an extension of the normal Euclidean bounding box regression model. For the batch training method, assuming there are M samples in a batch, the loss function for a batch of data is:

$$\hat{\Theta} = \frac{1}{M} \arg \min_{\Theta} \sum H(p_D(x), p_{\Theta}(x)). \tag{7}$$

The goal of the localization loss function is to estimate the trainable weight set $\hat{\Theta}$ that minimizes the cross-entropy loss L_{reg} over M samples.

3.4. NMS Based on Bounding Box Confidence

Conventional NMS eliminates overlapping bounding boxes by comparing the objectness score in a local region. It reserves only one bounding box that has the maximum objectness score. There is a gap between the objectness score and bounding box positioning because the objectness score mainly reflects the classification quality, and it is not equivalent to how precise the position of the predicted bounding boxes is. We used a confidence score to remedy it. We revised the objectness score by integrating confidence scores by the following equation, which considers both the objectness score and confidence scores:

$$Object_Score = \sqrt{Object_Score * Conf_Score}. \tag{8}$$

where $0 \leq Object_Score \leq 1$, $0 \leq Conf_Score \leq 1$. The radical sign in the equation is used to prevent the multiply results from becoming too small. We found using the radical sign in the equation led to better performance.

4. Results

The proposed model was evaluated on the KITTI [23] and Pascal VOC [24] datasets. We analyzed the performance by mAP (mean Average Precision) and computational resource consumption (Gflops, inference time). At the end of the experiment, the object detection result was used to analyze how the predicted degree of unconfidence affected the loss function to prove the rationality of the proposed methods. We carried out experiments

with a single NVIDIA 1080 Ti GPU, 8 GB of main memory, a Core i7 processor (3.2 GHz), and an Ubuntu 18.04 operating system.

4.1. Experiments on the KITTI Dataset

KITTI is an object detection dataset for autonomous driving. It contains 7381 training images with ground truth labels. There are three object categories in KITTI: vehicle, person, and cyclist. The objects to be detected are of various sizes and contain many small samples, which make it a challenging task. KITTI allows users to use it under the terms and conditions of the Creative Commons Attribution (CC BY) license. Users can copy, redistribute the material in any medium or format, and remix, transform, and build upon the material.

The training was composed of two stages. In the first stage, the learning rate increased from 10^{-6} slowly after it reached 0.001. In this stage, the truncated MobileNetv2 network was frozen, and only the MobileNeck layers were updated. In the second stage, the whole network was updated. The Stochastic Gradient Descent (SGD) optimizer was used. The learning rate was divided by 10 after the 80,000th and 100,000th iterations. The training was stopped at the 120,000th iteration. We scaled the input image size into various sizes such as 320×320 and 352×352 in the training stage. It took about 20 h to converge. In the evaluation stage, the input image size was scaled to 520×520 to achieve better performance. Because the labels of the test set were not publicly available, similar to SqueezeDet, we randomly split the 7381 training images in half into a training set and a validation set. All results were on the validation set. The batch size was eight. The backbone layer weights used for initializing the model were pretrained on ImageNet. Data augmentation techniques such as random cropping and flipping were adopted to avoid overfitting.

The performance of the models was evaluated in terms of mAP and Gflops (Giga floating-point operations per second). We re-evaluated MobileNetv2 with the SSD detection head, MobileNetv2 with MobileNeck, and its detection confidence-aware version for a fair comparison. In addition, other studies performed training using the default settings in the official code of each algorithm. To test the effect of MobileNeck and the confidence-aware loss function, we conducted a control experiment. First, we tested MobileNetv2 with the SSD detection head model. Second, we tested MobileNetv2 with MobileNeck and conventional loss. In this experiment, the loss was the same as that used in the first experiment. Third, we modified the loss function in the second experiment. We used the confidence-aware version in this experiment. The performance for each configuration is shown in Table 2.

Table 2. Control experiment on the KITTI dataset.

Method	Car	Pedestrian	Cyclist	mAP	Gflops
MobileNetv2 + SSD [1]	89.3	65.7	73.0	76.0	12.5
MobileNetv2 + MobileNeck (ours)	90.2	68.1	78.2	78.8	4.6
Conf-MobileDet (ours)	90.4	69.5	79.4	79.8	4.6

In the table, both the MobileNeck structure and confidence-aware loss have positive effects on the MobileNetv2-based object detection model. Compared to MobileNetv2 with the SSD detector, the proposed model outperformed it by 3.8 percentage points in terms of mAP with fewer Gflops.

We compare the proposed model with several classical lightweight deep artificial neural networks (SSD, YOLOv3, Gaussian YOLOv3, SqueezeDet, SqueezeDet+) in Table 3.

Table 3. Evaluation results on the KITTI dataset.

Method	Car	Pedestrian	Cyclist	mAP	Gflops
SSD [12]	85.1	48.1	50.7	61.3	88.2
YOLOv3 [15]	79.5	83.1	79.0	80.5	34.5
Gaussian YOLOv3 [21]	87.3	79.9	83.6	83.6	34.5
SqueezeDet [11]	82.9	70.4	76.8	76.7	9.7
SqueezeDet+ [11]	85.5	73.7	82.0	80.4	77.2
Conf-MobileDet (ours)	90.4	69.5	79.4	79.8	4.6

In the table, the proposed model has better performance than most other models in the table. Compared to SqueezeDet, which is the representative object detector designed for mobile devices, Conf-MobileDet outperformed it by 3.1 percentage in terms of mAP evaluation metrics and consumed fewer Gflops. Although SqueezeDet+ achieved better performance than Conf-MobileDet by 0.6 percentage points, its Gflops were 16 times those of Conf-MobileDet. For clarity, a scatter plot is drawn to compare different methods, whose horizontal and vertical ordinates are the Gflops and detection accuracy, respectively (Figure 5). As a result, Conf-MobileDet is more suitable for mobile devices that have only limited computational resources.

Figure 5. Scatter plot comparing Gflops and detection accuracy of different methods. mAP, mean Average Precision.

The detection result is shown in Figure 6. The predicted bounding boxes are represented by dashed rectangles. The predicted degree of unconfidence for the box size is represented by solid rectangles around the dashed rectangles, and the degree of unconfidence for box center coordinates is visualized by crossed lines inside the dashed rectangles. In order to facilitate the observation of the predicted degree of unconfidence, they are mapped to the exponential space:

$$\begin{aligned}
 \hat{uc}_x &= \min(\exp(uc_x * b), w/2.0), \\
 \hat{uc}_y &= \min(\exp(uc_y * b), h/2.0), \\
 \hat{uc}_w &= \min(\exp(uc_w * b), w/2.0), \\
 \hat{uc}_h &= \min(\exp(uc_h * b), h/2.0),
 \end{aligned} \tag{9}$$

where uc_* is the degree of unconfidence. b is a weighting factor, and we set $b = 15$ in this experiment. $\min(*, *)$ is used to prevent the boundary of the solid rectangle and cross lines from deviating too much from their common center point. It can be seen that for

blurred objects or objects that are occluded by others, the offsets of their corresponding solid rectangles and cross lines from their common center point are larger than for objects that are easy to distinguish. Therefore, the loss in Equation (7) for difficult samples will be larger than that of easy samples accordingly. This will cause the network to converge in a direction that minimizes the losses provided by these difficult samples. It is better to zoom in the figures before viewing.

Figure 6. Object detection result on the KITTI dataset. It is better to zoom in the figures before viewing.

We extended the experiment under different circumstances. We introduced changes to the images (first line in Figure 7) such as illumination, rotation, translation, etc. We found that changes in illumination and rotation had a great impact on the original results

(The second line in Figure 7). We gradually reduced the illumination (the third line to the fifth line in Figure 7) and increased the rotation angle (the sixth line to the eighth line in Figure 7) to the original images. We found that the degree of unconfidence increased gradually. This experiment indicated that adequate illumination and correct angles were important for the safety of autonomous driving as it output a low degree of unconfidence.

Figure 7. Cont.

Figure 7. Object detection result on the KITTI dataset under different circumstances. It is better to zoom in the figures before viewing.

4.2. Experiment on the Pascal VOC Dataset

We also tested the proposed methods on the Pascal VOC dataset. Because inference time is essential for autonomous driving, in this experiment, we compared MobileNetv2 and Conf-MobileDet by mAP and inference time. The Pascal VOC dataset contains three computer vision tasks: classification, detection and segmentation. The object detection task has 20 different classes to be detected, such as cats, houses, and persons. We used the VOC 07+12 train and valsets for training and the VOC 07 test set for evaluation. In this experiment, the backbone was pretrained on the ImageNet dataset and fine-tuned on the Pascal VOC dataset.

The training was composed of two stages. In the first stage, the learning rate slowly increased from 10^{-6} after it reached 0.0001. In this stage, the truncated MobileNetv2 network was frozen, and only the MobileNeck layers were updated. In the second stage, the whole network was updated. The SGD optimizer was used. The learning rate was divided by 10 after 80,000 and 100,000 iterations. The training was stopped at 120,000 iterations. We scaled the input image size into various sizes, such as 320×320 and 352×352 in the training stage. It took about 12 h to converge. In the evaluation stage, the input image size was scaled to 320×320 for fast reference speed. We list the mAP and inference time of MobileNetv2 model and the proposed model in Table 4.

Table 4. Control experiment on the Pascal VOC dataset.

Method	mAP	Inference Time (ms)
MobileNetv2 + SSD [1]	68.4	79.4
MobileNetv2 + MobileNeck (ours)	70.1	58.8
Conf-MobileDet (ours)	71.3	58.8

In the table, both the MobileNeck structure and confidence-aware loss have positive effects on the MobileNetv2-based object detection model. Compared to MobileNetv2, Conf-MobileDet outperformed it by 2.9 percentage points in terms of mAP with less inference time.

The detection result on VOC dataset is shown in Figure 8. The predicted bounding boxes are represented by dashed rectangles. The predicted degree of unconfidence for box size is represented by solid rectangles around the dashed rectangles, and the degree of unconfidence for box center coordinates is visualized by crossed lines inside the dashed rectangles. In order to facilitate the observation of the predicted degree of unconfidence, they are mapped to the exponential space before drawing. We set $b = 12$ in this experiment (Equation (9)). It can be seen that as more occlusions are added, the predicted degree of unconfidence will be higher. This is the way the predicted degree of unconfidence impacts

the confidence-aware loss function. The higher the degree of unconfidence, the larger the loss is. The computational resources will tilt to these “hard” samples. It is better to zoom in the figures before viewing.

Figure 8. Object detection result on the Pascal VOC dataset. It is better to zoom in the figures before viewing.

5. Discussion

In this paper, an object detection architecture based on MobileNeck and confidence-aware loss was proposed. We evaluated MobileNeck and confidence-aware loss on the KITTI and Pascal VOC datasets. Experimental results showed that it had better perfor-

mance compared to previous mobile device-oriented object detection architectures such as MobileNetV2 and SqueezeDet. In addition, we analyzed how the degree of unconfidence impacted the training of the model. As a result, the issues mentioned in the Introduction Section were fixed.

In the future, our challenge is to apply the proposed methods to other object detection tasks, such as infrared object detection (Figure 9). Infrared object detection are often applied to early warning systems. The objects to be detected in this task often lack rich texture features, and the background is complex. The left figure of Figure 9 is a failure example: it misjudges the background as the object. We hope the proposed method can predict the degree of unconfidence so that we can trust the result with a low degree of unconfidence and pay more attention to improving the detection result with a high degree of unconfidence, which in turn would reduce the false alarm rate of early warning systems.

Figure 9. Infrared object detection result.

Author Contributions: W.L. wrote the main manuscript text and prepared Figures 1–9. K.L. contributed to the interpretation of the fundamental theories. K.L. and W.L. conceived of and designed the experiments, discussed the issues, and exchanged views on the manuscript. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant Nos. 61571345 and 91538101.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in KITTI, Pascal VOC at 10.1109/CVPR.2012.6248074, 10.1007/s11263-009-0275-4, reference number [23,24].

Acknowledgments: The authors would like to thank the reviewers for their helpful comments and criticism.

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

References

1. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–23 June 2018; pp. 4510–4520.
2. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
3. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
5. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [[CrossRef](#)]
7. 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.
8. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.

9. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
10. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Adam, H. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019.
11. Wu, B.; Iandola, F.; Jin, P.H.; Keutzer, K. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
14. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
15. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
17. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
18. 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, Istanbul, Turkey, 30–31 January 2018.
19. Zhou, D.; Fang, J.; Song, X.; Guan, C.; Yin, J.; Dai, Y.; Yang, R. IoU Loss for 2D/3D Object Detection. In Proceedings of the 2019 International Conference on 3D Vision (3DV), Québec City, QC, Canada, 16–19 September 2019.
20. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
21. Choi, J.; Chun, D.; Kim, H.; Lee, H.J. Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019.
22. He, Y.; Zhu, C.; Wang, J.; Savvides, M.; Zhang, X. Bounding Box Regression with Uncertainty for Accurate Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
23. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
24. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Development Kit. 2006. Available online: <http://host.robots.ox.ac.uk/pascal/VOC/> (accessed on 28 March 2021).