

Article

Object Detection and Depth Estimation Approach Based on Deep Convolutional Neural Networks [†]

Huai-Mu Wang ¹, Huei-Yung Lin ^{1,2}  and Chin-Chen Chang ^{3,*}

¹ Department of Electrical Engineering, National Chung Cheng University, Chiayi 621, Taiwan; huaimu@godel.ee.ccu.edu.tw (H.-M.W.); lin@ee.ccu.edu.tw (H.-Y.L.)

² Advanced Institute of Manufacturing with High-Tech Innovations, National Chung Cheng University, Chiayi 621, Taiwan

³ Department of Computer Science and Information Engineering, National United University, Miaoli 360, Taiwan

* Correspondence: ccchang@nuu.edu.tw

[†] This paper is an extended version of the paper entitled “A Real-Time Forward Collision Warning Technique Incorporating Detection and Depth Estimation Networks”. Presented at 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020.

Abstract: In this paper, we present a real-time object detection and depth estimation approach based on deep convolutional neural networks (CNNs). We improve object detection through the incorporation of transfer connection blocks (TCBs), in particular, to detect small objects in real time. For depth estimation, we introduce binocular vision to the monocular-based disparity estimation network, and the epipolar constraint is used to improve prediction accuracy. Finally, we integrate the two-dimensional (2D) location of the detected object with the depth information to achieve real-time detection and depth estimation. The results demonstrate that the proposed approach achieves better results compared to conventional methods.



Citation: Wang, H.-M.; Lin, H.-Y.; Chang, C.-C. Object Detection and Depth Estimation Approach Based on Deep Convolutional Neural Networks. *Sensors* **2021**, *21*, 4755. <https://doi.org/10.3390/s21144755>

Academic Editor: Hossam A. Gabbar

Received: 2 June 2021

Accepted: 9 July 2021

Published: 12 July 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/>).

Keywords: object detection; depth estimation; stereo vision; deep learning

1. Introduction

Autonomous driving techniques [1–3] have been studied intensively for several decades. Because of the advances in sensor technology and the demands of commuters, manufacturers have expended considerable resources on developing autonomous vehicles. The Society of Automotive Engineers classifies five levels of automated driving, the third level of which is conditional automation, or self-driving under ideal conditions with limitations. This level has drawn much attention as developers attempt to implement effective detection and recognition of the surrounding environment (e.g., the road, traffic signs, other vehicles, and pedestrians) so that the vehicle can detect and recognize objects ahead and estimate their depth from a visual sensor.

Conventional approaches to object detection use multiple windows of varying sizes to slide repeatedly over images at fixed distances to detect objects of interest. Felzenszwalb et al. [4] presented a pedestrian detection approach that used a deformable part model with a histogram of oriented gradients and a support vector machine. Recently, the advent of convolutional neural networks (CNNs) [5–11] rapidly superseded traditional object detection. These deep neural networks hypothesize bounding boxes, extract features from them, and use high-quality object classifiers.

In this paper, we propose a real-time object detection and depth estimation approach using learning-based techniques for images acquired from a vehicle's onboard camera. First, we present an improved object detection approach—in particular for small objects—and then use deep neural networks and epipolar geometry to create stereo images and generate depth maps. Our approach modifies the monocular depth estimation network [12] for

binocular images and introduces a new correlation function to generate disparity maps through model training with some image reconstruction loss. Finally, the two-dimensional (2D) location is integrated with depth information to achieve effective object detection and depth estimation. This paper is organized as follows: Section 2 reviews related works. Section 3 is the proposed approach. Section 4 describes the implementation and results. Section 5 presents the conclusions.

2. Related Works

CNN-based object detection approaches [7–11] were primarily divided into two categories, namely, one-stage and two-stage detection. In two-stage detection, features were extracted from candidate regions and targets were classified. Such methods included the faster region-based CNN (Faster R-CNN) [7], region-based fully convolutional network (R-FCN) [8], and feature pyramid network (FPN) [9]. A network structure with heuristic sampling was used to target the class imbalance problem, and cascading can regress the parameters of the bounding box. Generally, CNN representation played a key role in these methods. The learned features were designed to encode highly discriminative and robust object characteristics with a moderate position bias. Several approaches were proposed to address these problems. For instance, ResNet and Inception both extracted features from deeper CNN backbones [13,14]. The FPN introduced a top-down architecture to construct feature pyramids and integrated low- and high-level information [9]. However, extracting such features from deeper neural networks led to high computational costs and networks with a low inference rate.

In a one-stage detection method, designed to be efficient and computationally low cost, candidate region extraction and target classification are performed in an end-to-end network, such as a Single Shot MultiBox Detector (SSD) [10] or a You Only Look Once (YOLO) [11]. To accelerate the detection phase, a single-stage framework was proposed and an object proposal generation was removed. YOLO [11] and SSD [10] have demonstrated the possibility of real-time processing with a clear drop of 10 to 40% of current two-stage solutions. RetinaNet [15] substantially improved the precision scores so that they become comparable to the highest scores reported for two-stage detectors. Unfortunately, these performance gains were credited to the deep ResNet-101 model [13], which greatly limited efficiency.

The SSD used multiple bounding boxes of different sizes to detect dense objects quickly and accurately. However, its small-vehicle detection performance was low because it ignored the smaller features between layers. The average precision (AP) and average recall for small objects in the Microsoft Common Objects in Context (COCO) dataset [16] were only 5.3 and 9.6%, respectively [10]. Other approaches for detecting small objects must be developed for particular applications. Because shallow convolutional networks produce feature and texture loss, previous approaches could not detect small objects effectively. To address this problem, RefineDet [17] was adopted as the main detection framework. It used the advantages of the Faster R-CNN and SSD and incorporated the FPN for shared features. It exhibited an AP of 25.6% on the COCO dataset over the original SSD and a high frame rate. In our approach, we used the global information in the convolutional layers to improve the transfer connection blocks (TCBs). The detection benchmarks were then evaluated based on the PASCAL Visual Object Classes (VOC) and COCO datasets [16,18].

For traditional stereopsis, environmental parameters were limited and certain parts of the left and right images were inconsistent because the angles at which the user viewed the images prevented the calculation of disparity values. To solve the two problems, several depth estimation approaches were proposed. The conventional stereo vision approaches [19–22] included CNN-based methods and video-based processing. Deep learning techniques markedly improved depth estimation performance on the KITTI dataset [23]. For different input sources, the existing networks were modified for single-view [12,24] and stereo-view depth estimation [19,25,26] with a multi-scale CNN and probabilistic graphical models.

Several approaches based on fully convolutional networks (FCNs) for view synthesis and depth estimation have been proposed, wherein ground-truth depth maps were not necessary in the training stage. In Deep3D [27], left images were input to binocular vision images to create the corresponding right images. The core idea was to use a single image to reconstruct stereopsis, with the disparity values of the single image predicted on the basis of the probabilities of such values occurring for each image pixel. Next, the disparity images were used to synthesize the right images. To predict more accurate stereo images in unsupervised monocular image depth estimation networks (e.g., Monodepth [12]), the consistency and gray-scale smoothness of the left–right stereo images and right–left stereo images were carefully considered before the loss function was modified. Unsupervised training-based depth estimation networks were subsequently introduced to solve these problems.

3. Proposed Approach

In the proposed approach, we presented a real-time object detection and depth estimation approach based on a light-network structure. Our approach consisted of two parallel modules: object detection and depth estimation. In the flowchart in Figure 1, input images were acquired from a vehicle’s onboard camera from which we presented an improved object detection approach. Then, we use deep neural networks and epipolar geometry to create stereo images and generate depth maps. We modified the stereo image network for disparity prediction and used the epipolar constraint to derive depth images from these disparities. Finally, we integrated the 2D location with the depth information and output the results for applications.

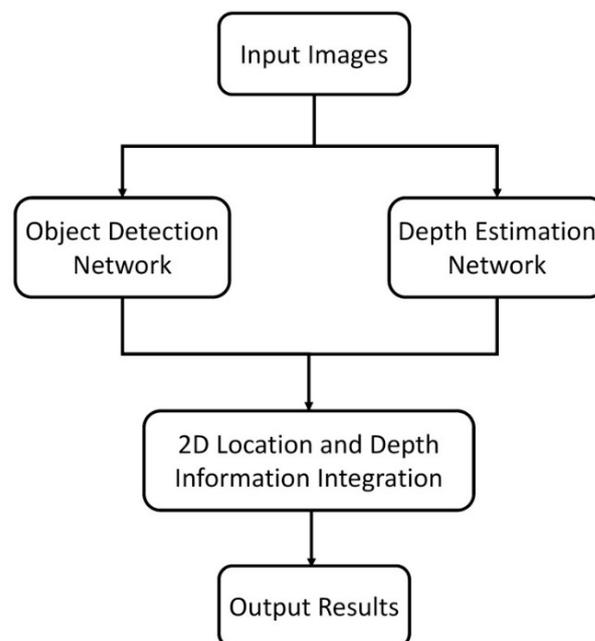


Figure 1. Flowchart of the proposed approach.

3.1. Object Detection

Object detection networks are plagued by problems such as high computational cost and inaccurate identification of small and faraway objects. Because small objects occupy a small space in the images, their detailed features are filtered out in the first few convolutional layers, and they are consequently ignored. To address this problem, we consulted the FPN, in which the feature information of all feature layers is shared and detailed patterns or context features are retained. Additionally, to reduce computational cost, we employ VGG16 as the backbone network.

RefineDet [17], which focuses on small-object detection and real-time computing capabilities, contains a network model that combines a two-stage and a one-stage detection network. It consists of two inter-connected modules: the anchor refinement module (ARM) and the object detection module (ODM). The ARM is used to remove negative anchors to reduce search space and roughly adjust the locations and sizes of anchors for better initialization. The ODM is used to regress correct locations of objects and predict multi-class labels based on the refined anchors. The TCBs are designed to fuse the information of the upper and lower convolutional layers between the ARM and the ODM. In our approach, we modified RefineDet [17] to improve small-object detection based on the following two improvements.

(1) Enhanced fine-feature extraction:

Inspired by single-shot face detection with feature fusion and segmentation supervision [28], we replaced element-wise addition with element-wise multiplication to prevent overflow. To suppress noise, we use the parametric rectified linear unit (PReLU) [29] as the activation function. The differences between correct classification and misclassification were strengthened and the lower computational cost improved object detection. This solved the problem of poor object detection for small and distant objects and high extraction rates in incorrect candidate regions.

(2) Shared global information with features of each pixel:

We imported global features, all feature maps could share the global information of other feature maps. Hence, we strengthened crucial features and suppress noise.

In the following, we introduced improvements for object detection in more detail. The proposed approach combined the concept of a two-stage detector into a one-stage network. We first extracted the object-agnostic region from the ARM and then used the ODM to classify the multi-scale object within the selected region. Finally, we added the TCBs to connect the feature maps to share information between low- and high-level layers. Figure 2 shows a flowchart of the improved TCB used for the modified RefineDet. We first up-sampled Layer (L) to match the dimensions between Layer (L) and Layer ($L-1$). Then, we replaced element-wise addition with element-wise multiplication. Third, transferred features were obtained by multiplying up-sampled Layer (L) and Layer ($L-1$) in the element-wise way. After that, we concatenated Layer ($L-1$) to the obtained transferred features. Finally, we applied the PReLU activation function before the convolutional layer.

Moreover, we used the SENet [30] to enable the received global information in the feature maps to be shared with each cell. All feature maps shared the global information of other feature maps, strengthening crucial context features and suppressing noise. Figure 3 illustrates the TCB model with incorporated squeeze-and-excitation flow. For a convolutional layer, SE-Block proposes to share the global features by using global pooling. Then, SE-Block uses a fully connected (FC) layer, the Sigmoid function, and the ReLU function to limit model complexity. The final output of the SE-Block is obtained by scaling.

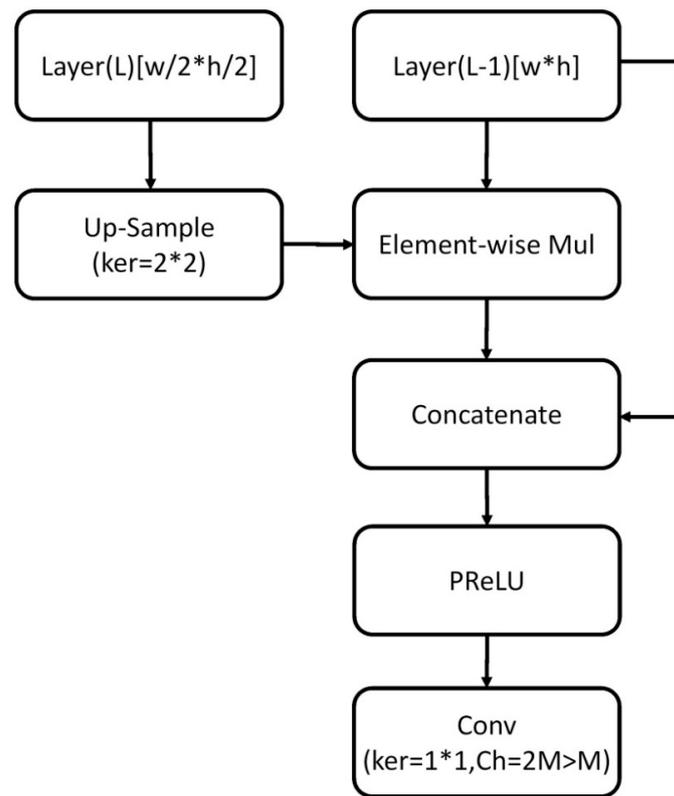


Figure 2. Flowchart of the improved TCB used for the modified RefineDet.

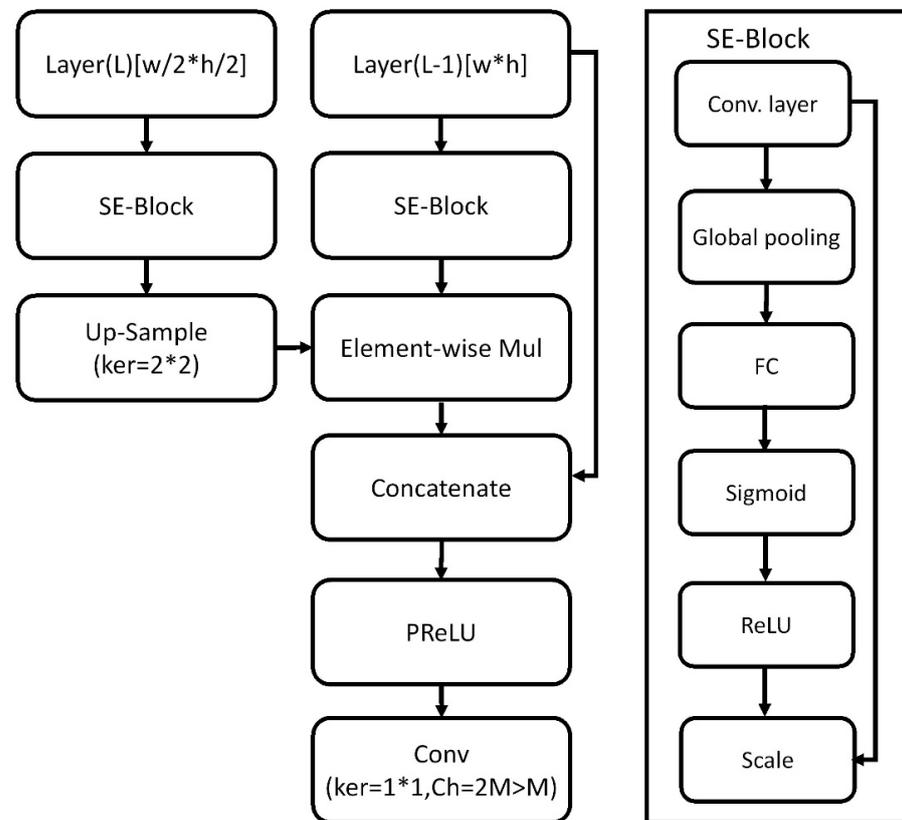


Figure 3. TCB block model with incorporated squeeze-and-excitation flow.

3.2. Depth Estimation

For depth estimation, the proposed approach was based on Monodepth [12] and used stereo vision to predict the disparity maps [31,32]. The original Monodepth was designed for disparity estimation from monocular images but not binocular images. Hence, we modified the network structure for binocular images, as depicted in Figure 4. The improvements made to the depth estimation network architecture are as follows:

- (1) Input layer: We input left and right images;
- (2) Shared convolutional layer: With shared weights, we used the same convolution kernel to extract features of left and right images;
- (3) Correlation layer: We use mathematical inner product operations to match the common regions between left and right feature maps;
- (4) Disparity map prediction: We predicted all the possible disparity values for all matching points using a normal distribution method for six different scales;
- (5) Grayscale image reconstruction: We reconstructed the left and right images on the basis of the predicted disparity maps and the internal camera parameters for the six scales.

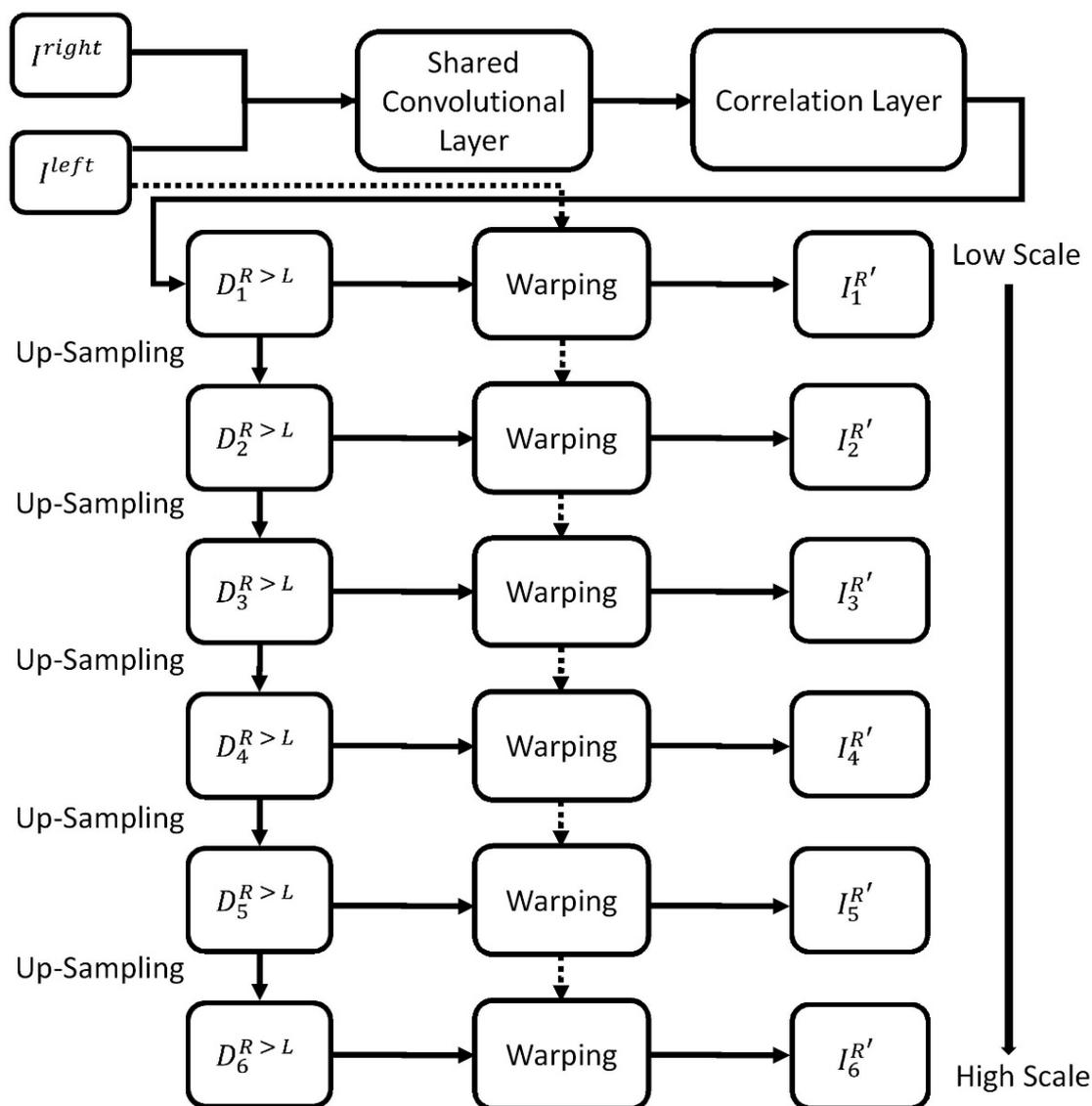


Figure 4. Modified network structure with binocular images. Stereo image pairs are used to generate the disparity maps.

Moreover, we proposed a learning method to perform single-image depth estimation with a deep neural network despite the lack of ground-truth depth information. Compared with other approaches, the accuracy of the output disparities was lower for the single-image input. We therefore introduced a training loss to improve the robustness and consistency of the generated left and right images. We defined an image reconstruction loss function L_{total} with the epipolar constraint for the disparity map generation as follows:

$$L_{total} = \alpha_{ap} (L_{ap}^l + L_{ap}^r) + \alpha_{ds} (L_{ds}^l + L_{ds}^r) + \alpha_{lr} (L_{lr}^l + L_{lr}^r), \quad (1)$$

$$L_{ap}^l = \frac{1}{N} \sum_{i,j} \alpha \frac{1 - SSIM(I_{i,j}^l, \hat{I}_{i,j}^l)}{2} + (1 - \alpha) \| (I_{i,j}^l, \hat{I}_{i,j}^l) \|, \quad (2)$$

$$L_{ds}^l = \frac{1}{N} \sum_{i,j} \left| \delta_x d_{i,j}^l \right| e^{-\|\delta_x I_{i,j}^l\|} + \frac{1}{N} \sum_{i,j} \left| \delta_y d_{i,j}^l \right| e^{-\|\delta_y I_{i,j}^l\|}, \quad (3)$$

$$L_{lr}^l = \frac{1}{N} \sum_{i,j} \left| d_{i,j}^l - d_{i,j+d_{i,j}^l}^r \right|. \quad (4)$$

In Equation (1), L_{total} consists of $L_{ap}^l, L_{ap}^r, L_{ds}^l, L_{ds}^r$ and L_{lr}^l, L_{lr}^r with weights α_{ap}, α_{ds} and α_{lr} . In (2), L_{ap}^l indicates the structural similarity (SSIM) index consistency between the two left images (the original ground-truth image and the generated image); N is the number of pixels; and $I_{i,j}^l, \hat{I}_{i,j}^l$, and $SSIM(\cdot)$ represent the real image, generated image, and SSIM, respectively. In (3), L_{ds}^l indicates the smoothness of the generated image for the surrounding pixels, where $\delta_x, \delta_y, d_{i,j}$, and $I_{i,j}$ represent the Gaussian standard deviation on the x -axis direction, the Gaussian standard deviation on the y -axis direction, the depth of the pixel, and the original image, respectively. In (4), L_{lr}^l indicates the consistency between the two predicted disparity maps, where $d_{i,j}^l$ and $d_{i,j+d_{i,j}^l}^r$ are the left and right disparity values, respectively. Also, L_{ap}^r, L_{ds}^r , and L_{lr}^r can be defined similarly;

Another component of our approach was the input of the image pair to the low-level layers of the convolutional network to determine common features between the left and right images. The features are then sent to DispNetC [19] for correlation prediction. In the network training stage, DispNetC extracts the inner product of the conjugate epipolar lines for feature matching. The correlation of two feature maps centered at x_l in the left feature map and x_r in the right feature map is defined by

$$c(x_l, x_r) = \sum_{o \in [-k, k] \times [-k, k]} \langle f_l(x_l + o), f_r(x_r + o) \rangle, \quad (5)$$

where f_l and f_r are the left and right feature maps, respectively; k is a constant.

All possible disparities $D_{i,j}$ in the image are predicted by the normal distribution

$$\sum_d D_{i,j}^d = 1, \quad 0 < d < 1, \quad (6)$$

where d is the probability of the matching point corresponding to each disparity value and

$$D = \frac{f \cdot B}{Z}, \quad (7)$$

where f, B , and Z are the focal length, stereo baseline, and depth, respectively. The disparity map is then converted to a gray-level image by

$$\hat{I}_{i,j}^r = \sum_d I_{i,j}^d D_{i,j}^d, \quad (8)$$

and

$$I_{i,j}^r = I_{i,j+D}^l. \quad (9)$$

Finally, the L_1 norm is used and the loss function for network prediction is calculated by

$$L_1 = \| \hat{I}^r - I^r \| . \quad (10)$$

4. Implementation and Results

This section reported the implementation and results of the proposed approach for object detection and depth estimation.

4.1. Implementation

Unlike the conventional approaches, this proposed approach did not require the parameters to be adjusted for different environments. Considering the computational constraint on mobile systems for vehicular applications, we used the same hardware platform (NVIDIA GTX 1080) to evaluate the performance of different algorithms. The stereo camera system used for data acquisition cost much less than light detection and ranging (LiDAR) or other time of flight (ToF) sensors. We performed quantitative analysis on several datasets and compared the proposed approach with previous methods. Datasets used for evaluation included PASCAL VOC [18], KITTI [23], BDD100K [33] and our own database. The software environment contained Ubuntu-16.04, Python 3.5, a Machine Learning API PyTorch graphics processing unit 0.40 [34], and a Tensorflow graphics processing unit 1.40 v. It was not easy to determine the parameters appropriately. In the experiments, these were set heuristically for the best performance. The training parameters were as follows: the learning rate, iteration, and batch size for the detection network were 0.01, 120,000, and 16, respectively; those for the depth estimation network were 0.01, 50, and 8, respectively.

4.2. Evaluation on Object Detection

For the detection network, the improved TCB structure over RefineDet increased the accuracy but not the network complexity as the processing frame rate was maintained at a stable value. Small object detection was also improved for faraway vehicles. As shown in Figure 5, the detection range was increased approximately from 25 to 50 m and the frame rates of the two methods were maintained in real time.

The comparison of the various detection algorithms tested on the PASCAL VOC dataset is shown in Table 1. Our approach provided the best mAP compared with that of RefineDet [17], SSD [10], YOLOv2 [11], Faster R-CNN [7], and R-FCN [8]. Moreover, it maintained a rate of 25 frames per second (FPS). Hence, we detected objects in real time. For the KITTI dataset, we simplified the number of classes from 16 to 3 (car, person, and bicycle) for the evaluation and sped up the frame rate to 50 FPS on the NVIDIA GTX 1080 platform, as shown in Figure 6. Comparative test results for RefineDet are presented in Figure 7, where the mAP of the proposed approach was better than that of RefineDet. In addition, our approach detected the person class more effectively. For the BDD100K dataset, seven classes were used for evaluation. This dataset contained more challenging scenes, such as those in low illumination or containing occlusions. The object detection evaluation is illustrated in Figure 8. The results showed that the mAP of the proposed approach was larger than that of RefineDet. Moreover, our approach performed better than RefineDet for detecting objects. Although the accuracy was lower compared with the KITTI dataset results, our approach exhibited the desired improvements. Finally, we collected our own dataset from Taiwan road scenes for evaluation. The image sequences were captured from a car recorder at 30 FPS with a resolution of 1280×720 . The object detection evaluation in our dataset is illustrated in Figure 9. From the results, the mAP of the proposed approach was much better than that of RefineDet. In addition, our approach outperformed RefineDet for detection in each class.



(a)



(b)

Figure 5. Comparison of the object detection results. (a) The resulting image is obtained from the original RefineDet. (b) The resulting image is the output of our approach.

Table 1. Comparison of detection algorithms tested on the PASCAL VOC dataset.

Approach	Zhang et al. [17]	Liu et al. [10]	Redmon et al. [11]	Ren et al. [7]	Dai et al. [8]	Our Approach
Backbone	VGG-16	VGG-16	VGG-16	ResNet-50	ResNet-50	VGG-16
Training Data	PASCAL VOC	PASCAL VOC	PASCAL VOC	PASCAL VOC	PASCAL VOC	PASCAL VOC
Input Size	320×320	300×300	416×416	320×320	320×320	320×320
Boxes	6375	6200	Unknown	Unknown	Unknown	6500
FPS	25	35	67	2.4	5.9	25
mAP	79.49	75.3	76.8	73.8	77.6	79.75



Figure 6. Reducing the number of classes to speed up object detection.

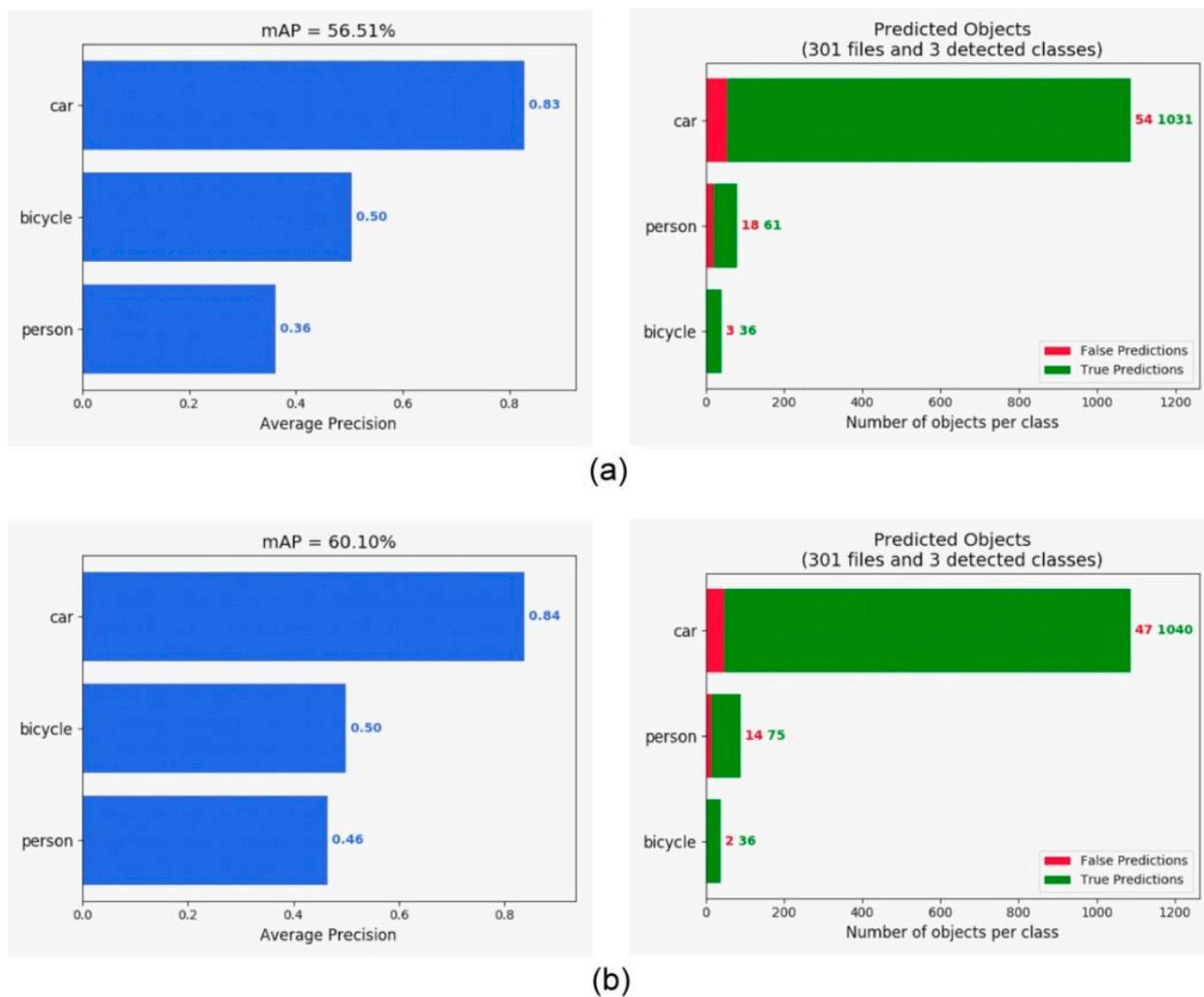


Figure 7. mAP comparison of the original RefineDet and our approach tested on the KITTI dataset. (a) Evaluation of RefineDet. (b) Evaluation of our approach.

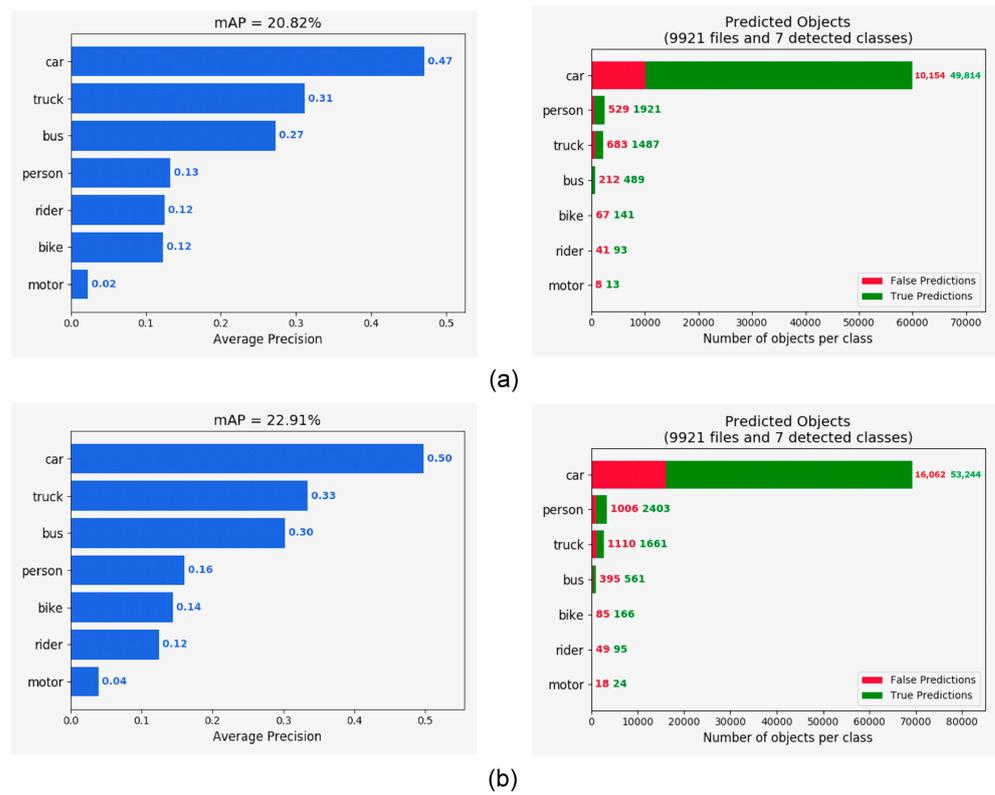


Figure 8. mAP comparison of the original RefineDet and our approach tested on the BDD100K dataset. (a) Evaluation of RefineDet. (b) Evaluation of our approach.

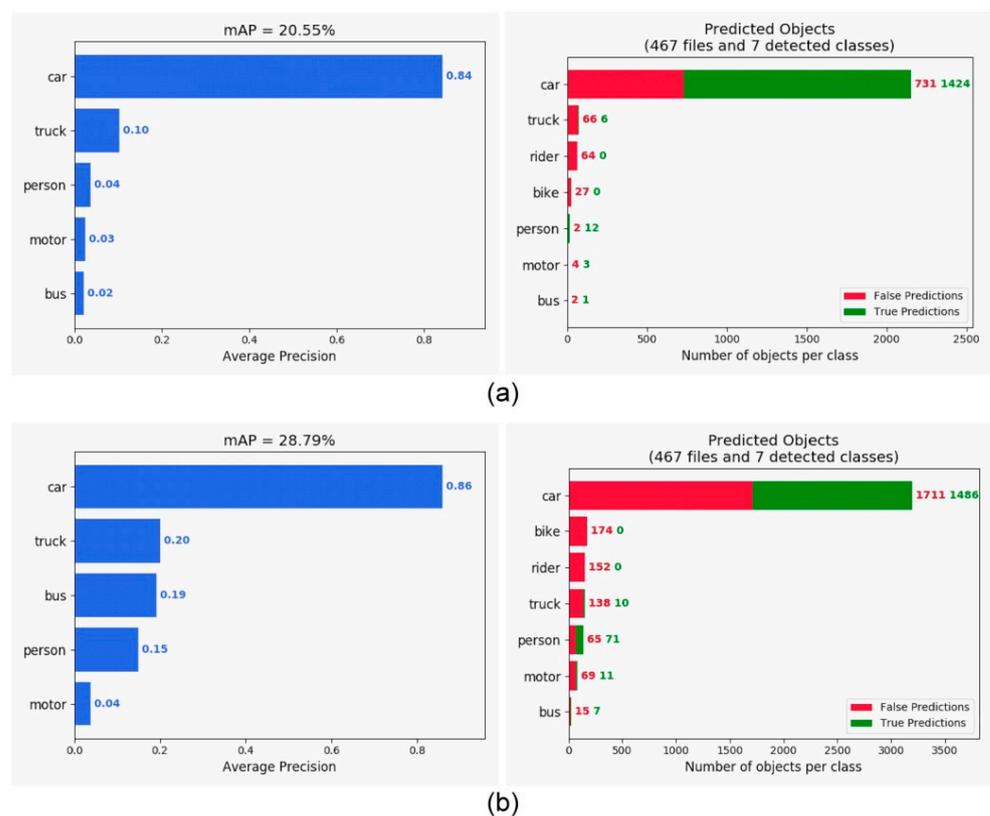


Figure 9. mAP comparison of the original RefineDet and our approach tested on our own dataset. (a) Evaluation of RefineDet. (b) Evaluation of our approach.

4.3. Evaluation of Depth Estimation

The depth estimation network was trained with unsupervised learning and had a lower computational cost, which for the proposed fully convolutional neural network depended on the size of the input images. Using low-resolution images as inputs, we reduced the cost with rough depth estimation. Using PyD-Net [32] as a reference for the depth estimation network, we designed a light fully convolutional neural network with only six FPN layers for the image reconstruction loss, which reduced the complexity of Monodepth as well as the computational cost.

The evaluation of the depth estimation network was performed with the KITTI dataset. A stereo image pair and the estimated disparity map are shown in Figure 10, which shows that the proposed approach can accurately estimate a dense depth map. Figure 11 shows several results of the depth map prediction with the input image (upper left), ground-truth disparity map (upper right), estimated disparity map (bottom left), and disparity difference between the ground-truth and prediction (bottom right). For traffic scene 1, there was an approaching vehicle and some objects. The depths of the vehicle and the objects were greatly estimated. For traffic scene 2, there were faraway small vehicles and some objects in the scene. The proposed approach estimated the depths well for the faraway small vehicles and the objects. For traffic scene 3, there was a vehicle for roadside parking and some objects in the scene, and the depths of the vehicle and the objects were reasonably estimated. These results showed that the proposed approach produces visually reasonable depth maps.

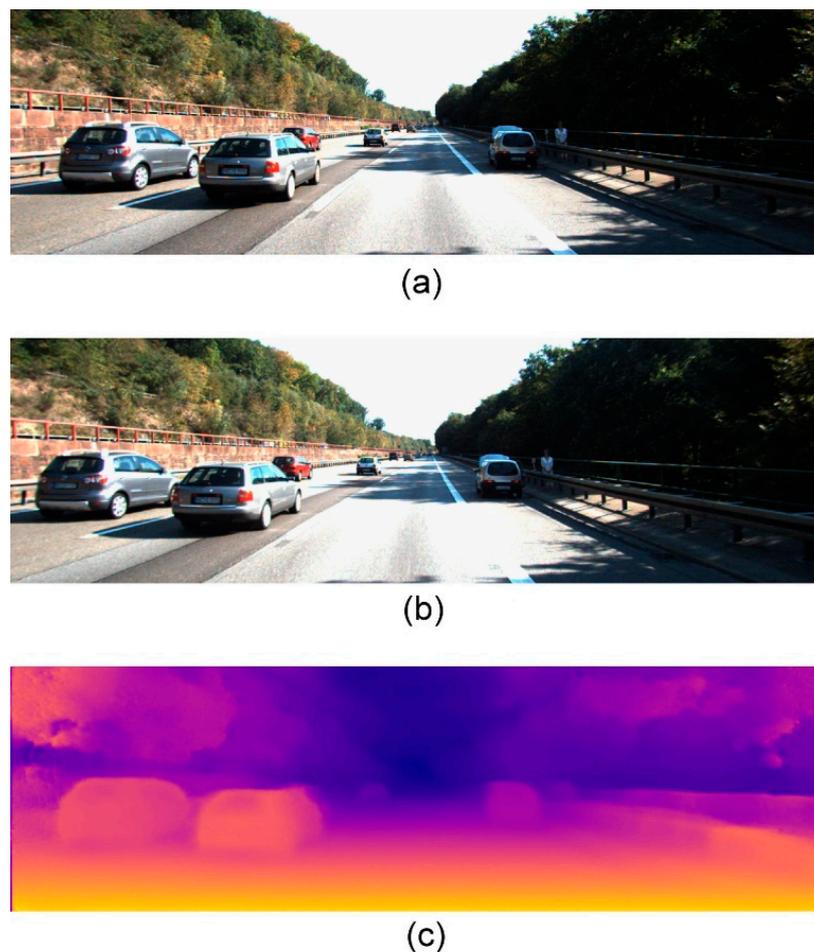


Figure 10. A stereo image pair and the estimated disparity map. (a) left image; (b) right image; (c) estimated disparity map.

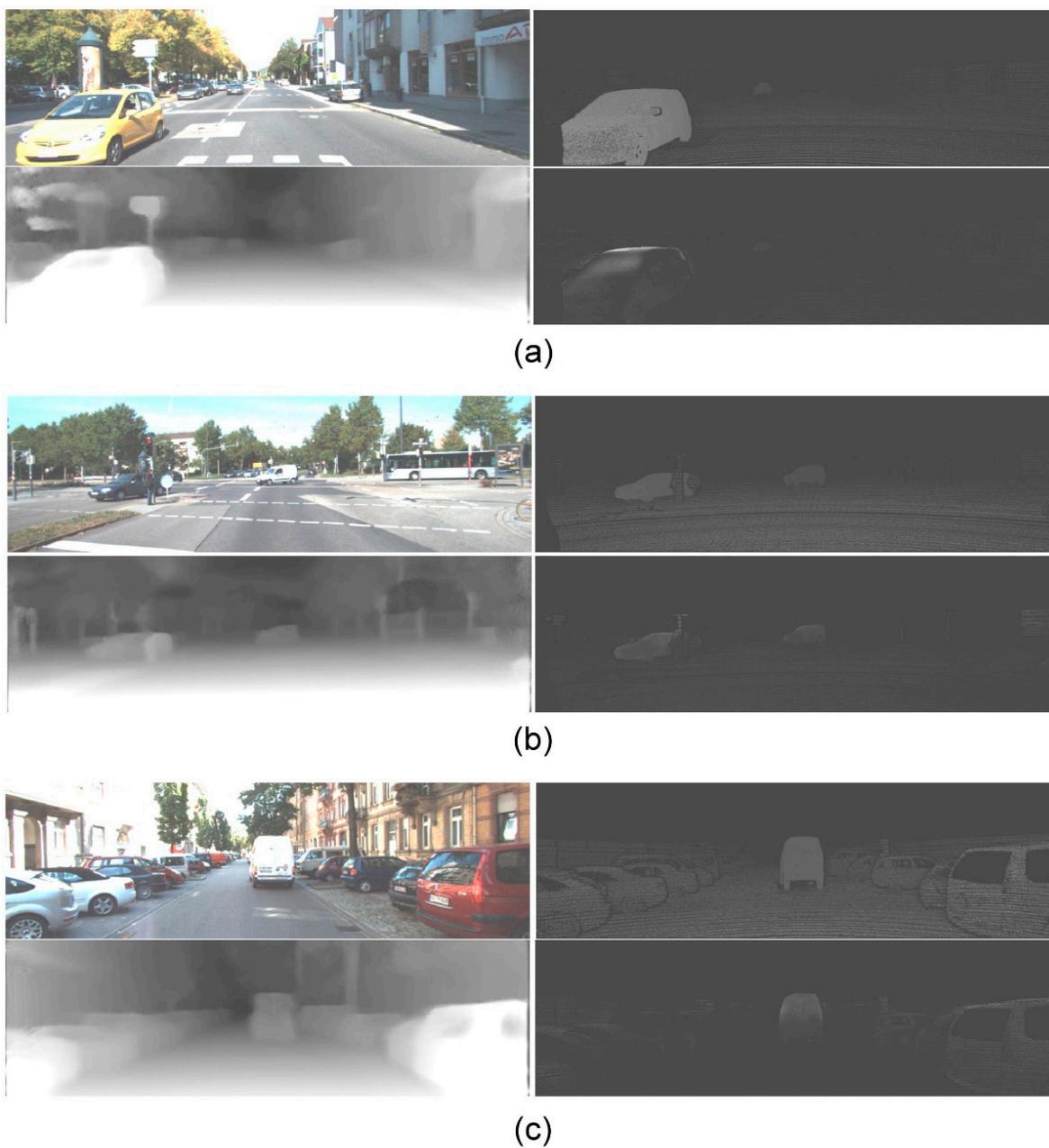


Figure 11. Three results of the depth map prediction using the KITTI dataset. (a) Traffic scene 1 with an approaching vehicle. (b) Traffic scene 2 with faraway small vehicles. (c) Traffic scene 3 with a vehicle for roadside parking.

Moreover, we adopted common evaluation metrics. Let P be the number of pixels. The notations d_i and \hat{d}_i are the ground-truth disparity value and estimated disparity value, respectively. Each metric was defined in the followings. The root mean square error RMS was defined by

$$RMS = \sqrt{\frac{1}{P} \sum_{i=1}^P (\hat{d}_i - d_i)^2}. \quad (11)$$

The absolute relative difference $Abs-rel$ was defined by

$$Abs-rel = \frac{1}{P} \sum_{i=1}^P \frac{|\hat{d}_i - d_i|}{d_i}. \quad (12)$$

The square relative difference $Sq-rel$ was defined by

$$Sq-rel = \frac{1}{P} \sum_{i=1}^P \frac{\|\hat{d}_i - d_i\|^2}{d_i}. \quad (13)$$

The root mean square logarithmic error $Log-rms$ was defined by

$$Log-rms = \sqrt{\frac{1}{P} \sum_{i=1}^P (\text{Log}\hat{d}_i - \text{Log}d_i)^2}. \quad (14)$$

The depth error ratio of one pixel Er was defined by

$$Er = \max\left(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}\right) < t, \text{ where } t \in [1.25, 1.25^2, 1.25^3]. \quad (15)$$

Finally, the evaluation metric $D1-all$ was defined as the percentage of misclassified pixels (error > 3 pixels) in the whole image.

The comparison of different algorithms is shown in Table 2, where the proposed approach outperformed all previous methods on $Log-rms$ and $D1-all$ metrics. Our approach had similar accuracy as the methods of Lai et al. (stereo only) [35] and Godard et al. [12] + Stereo and was better than the other methods on $Abs-rel$, $Sq-rel$, and RMS metrics; however, it was less accurate on $Er < 1.25$ and $Er < 1.25^3$ metrics. To summarize, the results indicated that the proposed approach had the same levels of error as the previous methods, and compared with the previous networks demonstrated its feasibility.

Table 2. Comparison of algorithms with various evaluation metrics.

Approach	<i>Abs-rel</i>	<i>Sq-rel</i>	<i>RMS</i>	<i>Log-rms</i>	<i>D1-all</i>	<i>Er < 1.25</i>	<i>Er < 1.25³</i>
Godard et al. [12]	0.124	1.40	6.137	0.217	30.350	0.841	0.975
Pilzer et al. [24] (half-cycle stereo)	0.228	4.277	7.646	0.318	Null	0.748	0.945
Pilzer et al. [24] (full-cycle+D+SE)	0.190	2.556	6.927	0.353	Null	0.751	0.951
Lai et al. (stereo only) [35]	0.078	0.811	4.700	Null	Null	0.983	Null
Poggi et al. [32]	0.153	1.363	6.030	0.252	Null	0.789	0.630
Godard et al. [12] + Stereo (no correlation)	0.083	0.944	4.765	0.163	13.087	0.927	0.986
The Proposed Approach	0.08	0.925	4.846	0.160	12.480	0.929	0.987

To accelerate the image processing, the resolution of each image was reduced to 512×256 pixels. Figure 12 shows the real-time processing results of two stereo image pairs. For each scene, the upper two images were stereo images. The disparity maps derived using our approach (bottom left) showed clear improvements compared with the disparity maps obtained from the original lightweight network (bottom right).

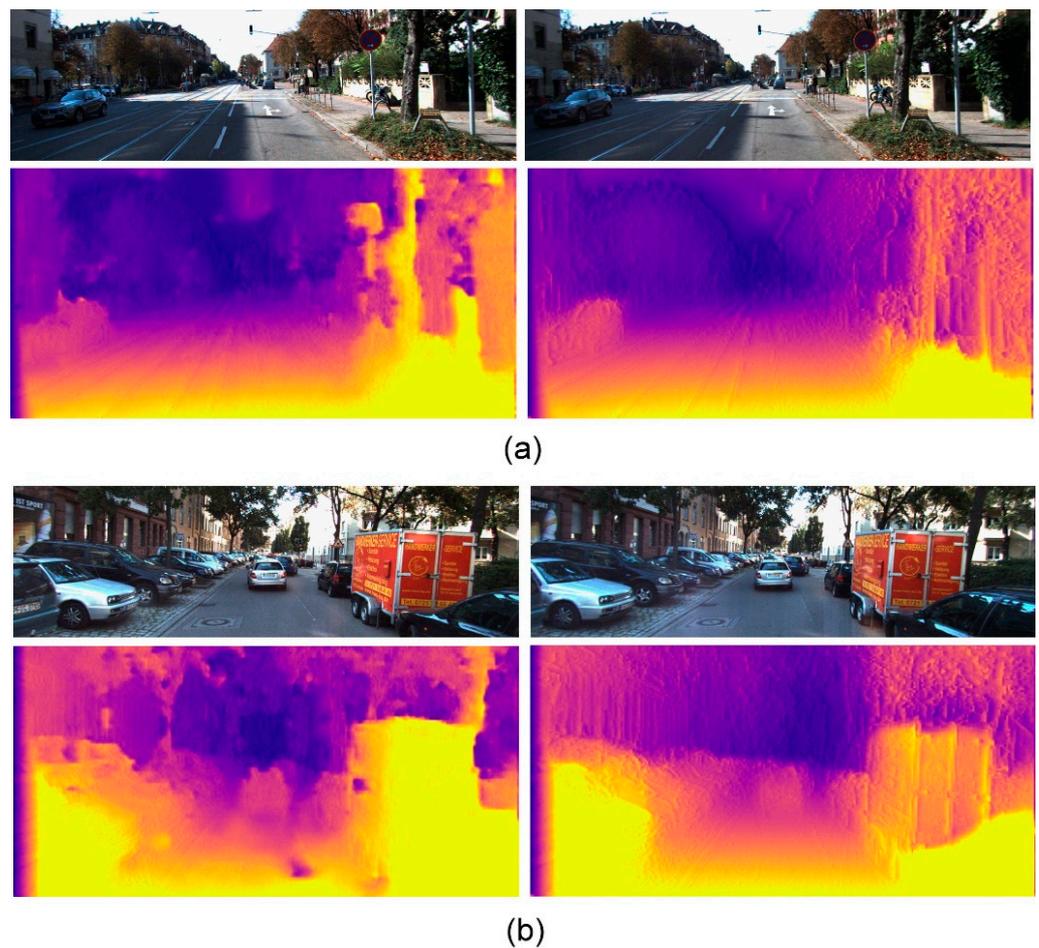


Figure 12. Disparity maps of two stereo image pairs generated using our approach (bottom left) display clear improvements over the original lightweight network (bottom right). (a) Scene 1. (b) Scene 2.

5. Conclusions

We presented an object detection and depth estimation approach based on deep learning techniques. Object detection was improved through the incorporation of the TCBS with the CNN as small objects were detected in real-time. Moreover, we applied binocular vision to the monocular-based disparity estimation network. The comparison with previous networks demonstrated the feasibility of the proposed approach. In future studies, object detection and depth estimation networks can be integrated into the feature extraction process within a convolutional network to reduce network and computational resources. Additionally, transfer learning will be used to enable the networks to train in object detection and depth estimation independently.

Author Contributions: Methodology, H.-M.W. and H.-Y.L.; Supervision, H.-Y.L. and C.-C.C.; Writing—original draft, H.-M.W.; Writing—review & editing, H.-Y.L. and C.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Ministry of Science and Technology of Taiwan for financially supporting this research under Contract No. MOST 106-2221-E-194-004.

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. Zhao, Q.; Sheng, T.; Wang, Y.; Ni, F.; Cai, L. Cfenet: An accurate and efficient single-shot object detector for autonomous driving. *arXiv* **2018**, arXiv:1806.09790.
2. Li, Y.F.; Tsai, C.C.; Lai, Y.T.; Guo, J.I. A multiple-lane vehicle tracking method for forward collision warning system applications. In Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPAASC), Kuala Lumpur, Malaysia, 12–15 December 2017; pp. 1061–1064.
3. Naghavi, S.H.; Avaznia, C.; Talebi, H. Integrated real-time object detection for self-driving vehicles. In Proceedings of the 2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP), Isfahan, Iran, 22–23 November 2017; pp. 154–158.
4. Felzenszwalb, P.F.; McAllester, D.A.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 24–26 June 2008.
5. Lai, C.; Lin, H.; Tai, W. Vision based ADAS for forward vehicle detection using convolutional neural networks and motion tracking. In Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019), Heraklion, Crete, Greece, 3–5 May 2019; pp. 297–304.
6. Wang, H.M.; Lin, H.Y. A real-time forward collision warning technique incorporating detection and depth estimation networks. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2020), Toronto, ON, Canada, 11–14 October 2020.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; pp. 91–99.
8. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2016; pp. 379–387.
9. 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.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV2016), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
11. 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 (CVPR2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
12. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 270–279.
13. 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 (CVPR2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
14. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
15. 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 (ICCV 2017), Venice, Italy, 22–29 October 2017; Volume 1, pp. 2999–3007.
16. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the 12th European Conference on Computer Vision (ECCV2014), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
17. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICPR 2018), Beijing, China, 20–24 August 2018; pp. 4203–4212.
18. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]
19. Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICPR2016), Cancún, Mexico, 4–8 December 2016; pp. 4040–4048.
20. Teed, Z.; Deng, J. Deepv2d: Video to depth with differentiable structure from motion. *arXiv* **2018**, arXiv:1812.04605.
21. Perez-Cham, O.E.; Puente, C.; Soubervielle-Montalvo, C.; Olague, G.; Castillo-Barrera, F.E.; Nunez-Varela, J.; Limon-Romero, J. Automata design for honeybee search algorithm and its applications to 3D scene reconstruction and video tracking. *Swarm Evol. Comput.* **2021**, *61*, 100817. [[CrossRef](#)]
22. Liu, Y.; Wang, H.; Dong, C.; Chen, Q. A car-following data collecting method based on binocular stereo vision. *IEEE Access* **2020**, *8*, 25350–25363. [[CrossRef](#)]
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 (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 3354–3361.
24. Pilzer, A.; Xu, D.; Puscas, M.; Ricci, E.; Sebe, N. Unsupervised adversarial depth estimation using cycled generative networks. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 587–595.
25. Tonioni, A.; Tosi, F.; Poggi, M.; Mattocchia, S.; Stefano, L.D. Real-time self-adaptive deep stereo. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2019), Long Beach, CA, USA, 16–20 June 2019; pp. 195–204.

26. Yang, G.; Zhao, H.; Shi, J.; Deng, Z.; Jia, J. Segstereo: Exploiting semantic information for disparity estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 636–651.
27. Xie, J.; Girshick, R.; Farhadi, A. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In Proceedings of the European Conference on Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 11–14 October 2016; pp. 842–857.
28. Tian, W.; Wang, Z.; Shen, H.; Deng, W.; Chen, B.; Zhang, X. Learning better features for face detection with feature fusion and segmentation supervision. *arXiv* **2018**, arXiv:1811.08557.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV2015), Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
30. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICPR2018), Beijing, China, 20–24 August 2018; pp. 7132–7141.
31. Huang, P.; Lin, H. Rear obstacle warning for reverse driving using stereo vision techniques. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC 2019), Bari, Italy, 6–9 October 2019; pp. 921–926.
32. Poggi, M.; Aleotti, F.; Tosi, F.; Mattoccia, S. Towards real-time unsupervised monocular depth estimation on cpu. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018), Madrid, Spain, 1–5 October 2018; pp. 5848–5854.
33. Yu, F.; Xian, W.; Chen, Y.; Liu, F.; Liao, M.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv* **2018**, arXiv:1805.04687.
34. Ketkar, N. Introduction to pytorch. In *Deep Learning with Python*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 195–208.
35. Lai, H.Y.; Tsai, Y.H.; Chiu, W.C. Bridging stereo matching and optical flow via spatiotemporal correspondence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2019), Long Beach, CA, USA, 16–20 June 2019.