Hindawi
Journal of Robotics
Volume 2021, Article ID 6652828, 7 pages
https://doi.org/10.1155/2021/6652828

Research Article

Building a Real-Time 2D Lidar Using Deep Learning

Nadim Arubai, Omar Hamdoun, and Assef Jafar

Higher Institute for Applied Sciences and Technology, Damascus, Syria

Correspondence should be addressed to Nadim Arubai; nadim.arubai@hiast.edu.sy

Received 10 November 2020; Accepted 24 January 2021; Published 5 February 2021

Academic Editor: L. Fortuna

Applying deep learning methods, this paper addresses depth prediction problem resulting from single monocular images. A vector of distances is predicted instead of a whole image matrix. A vector-only prediction decreases training overhead and prediction periods and requires less resources (memory, CPU). We propose a module which is more time efficient than the state-of-the-art modules ResNet, VGG, FCRN, and DORN. We enhanced the network results by training it on depth vectors from other levels (we get a new level by changing the Lidar tilt angle). The predicted results give a vector of distances around the robot, which is sufficient for the obstacle avoidance problem and many other applications.

1. Introduction

Depth prediction is an ill posed problem, but it is useful in many applications because it is cheap and easy to modify the hardware and software. It is used in many applications such as autonomous driving, obstacle avoidance, and object detection. Usually, a depth image is predicted, i.e., the grey color in every pixel of the output is equal to a real-world depth. Alternatively, a cloud of points is predicted using a Lidar system. In some applications, it is enough to use a 2D Lidar to collect depths. Therefore, we suggest shaping the output as a depth vector, i.e., a vector that contains distances around the robot in a fixed step.

We suggest training on depth vectors. These vectors are chosen as targets to obtain less complex models. We suggest training on multiple levels to increase the accuracy by accumulating the experience. We suggest a small CNN (4 layers) to solve the problem in real time (40 FPS). Depth vectors are not commonly used in image-related problems, and to our knowledge, there is no CNN (convolutional neural network) that has been evaluated for depth vector. Therefore, we tested many networks such as VGG, ResNet, FCRN, and DORN.

2. Related Works

Depth prediction using single image witness a success with Saxena et al. [1] in 2005 using MRF (Markov random field). They continued to develop the idea where they built a 3D scene from single image [2], while others used semantic segmentation [3] and CRF (conditional random field) [4].

After the unexpected results of AlexNet [5] in 2012 for image classification, CNNs become popular in image-related problems, including obstacle detection, super pixels, super resolution, semantic segmentation, normal predictions, and depth prediction.

Depth prediction with CNNs was started by Eigen et al. [6] in 2014 by predicting a rough image matrix of distances and then refining it by the original image to get a better prediction. Later in 2014 [7], they used another network to handle many image-related problems. In 2016, Laina et al. [8] used encoder-decoder structure and fully convolutional network for this task. They first decoded the image by ResNet and then used up-projection to get the depth predictions. In 2016, Cao et al. [9] dealt with the depth prediction as a classification problem. This method gives confidence for each class but suffers from the output discretization. In 2018, Fu et al. [10] used ordinal classification to predict the depth image and got results outperforming other methods. In 2019, Ren et al. [11] fused the regression and ordinal classifications to get better results.

Depth prediction was also solved as an unsupervised problem: from stereo images in 2016 [12, 13] and from sequential images in 2017 [14]. In 2019, Wang et al. [15, 16] used unsupervised learning to predict a depth image and

then projected it to a cloud in a 3D space and used the cloud for obstacle detections and got results similar to the real Lidars. In 2017, Kuznietsov et al. [17] used semisupervised learning by using two loss functions, one predicts depths from two images as unsupervised and the other predicts depth from real distances as supervised to get a better result than using just one of them.

In 2014, Liu et al. [18] used the CRF as preprocessing and then they used a CNN to get a good depth prediction. CRF with CNNs continues to be developed by [19, 20].In 2018 [21], the affinity learned was used to enhance the predictions by using the neighbouring pixels and recursive training to shift the pixels toward better values. This method gives a high detail depth image.

In 2015, Wang et al. [22] solved the semantic segmentation with the depth prediction using CNN as a joint problem to reach better results than by solving one of them alone. In 2018, Ramirez et al. [23] used semisupervised learning to solve jointly depth prediction with semantic segmentation. In 2019, Zhan et al. [24] solved it jointly with surface normal prediction. Joint problems for depth predictions appeared first by Ladický et al. [25] in 2014 without using CNNs by building a pyramid of the image and then predicting the correct scale factor to a predefined canonical depth using traditional features. Although this method gives better results to depth prediction and obstacle labelling than solving every problem alone, it needs a lot of resources.

Fusing sensors to get better depth predictions was used lately. In 2017, Ma et al. [26] benefited from knowing sparse depths and the image to get a dense depth image. In 2019, Xia et al. [27] gave a general model which could benefit from any known depth in the image to enhance the overall predictions. This method also gives a confidence image for the predictions.

Working at the stage of sensor is also used for depth prediction: OPA (offset pixel aperture) also called DP (duel-pixel) cameras [28]; these cameras have spilt green pixels to two halves that could be used for depth estimation combining with machine learning for enhanced estimations (with a single camera [29] and with a stereo [30]). Analog pixel-to-pixel conversion is used for object and motion detection [31]. It works on pixels before they get digitalized. For analog videos, a different CNN (cellular nonlinear (or neural) network) [32] is used for image segmentation [33], motion detection [34], and other applications. Cellular nonlinear networks use analog and logic circuits to process the video in real time.

It is hard to categorize the algorithms used for this depth prediction. However, we have identified the following categories as in Figure 1:

(i) Learning-based problems: supervised, unsupervised, and semisupervised.

(ii) Based on target continuity: regression, classification, and ordinal classification.

(iii) Based on preprocessing: with CRF, end-to-end, and other.

(iv) Based on input type: single image, stereo, and video.

(v) Based on target shaping: depth image and depth vector.

(vi) Fused with other sensors: OPA (offset pixel aperture), sparse Lidar cloud points, and disparity image from RGBD cameras.

The method we proposed is based on target shaping.

These techniques and others predict the depth image in good accuracy, but they need an RGBD camera, 3D Lidar, kinetic, or stereo (two calibrated cameras) to obtain the targets for training. There are many datasets that are free and ready to download on the Internet for depth prediction. However, in real environments, one of the mentioned expensive sensors is needed for training and fine tuning.

We suggest using a 2D Lidar to obtain vectors of distances and consider them as targets to predict from a single grey image. To solve the problem, we suggest a light weight CNN which is sufficiently accurate and fast in execution. First, we discuss the needs, and then we will show the test results on the CNNs using a derived dataset from KITTI to accomplish a vector depth prediction on multilayers.

The related work to our research is limited to pseudo-Lidar. Wang et al. [15, 16] used unsupervised learning and CNNs to predict depth image. The pixels are then projected on a 3D cloud related to the Lidar system. Jeff et al. [35] predicted the nearest point in vertical rectangles for obstacle avoidance using reinforcement learning with usual features extraction (no CNN). Their results give good relative depths while we need to predict the absolute depths. Our technique differs from theirs because we use supervised learning to predict the exact location of recorded points, which is related to calibration and geometrical posing of the sensors. The main benefit of our project is to predict depth in fast manner around a robot in 2D and create a multilayer to be near 3D. After training, cameras are only needed to be installed on several sides of a robot more easily than Lidars and they are cheaper.

We aim to predict a vector of depths from a single grey image using CNNs in an end-to-end manner. We need a robot with a camera for the input and 2D Lidar for the output. We need to record or derive a suitable dataset. Further, we need a suitable CPU for training and testing. We also need to calibrate the sensors.

3. Proposed Model

We use the ASPP [36] module to get a pyramid of features maps, which increases the filter's field of view on the previous feature map by using dilated convolutions.

The network is simple: it consists of 2 convolutional layers for low level features, an ASPP layer for local and global understanding, followed by a dropout layer, and a fully connected layer for depth predictions. We use two pooling layers before and after the ASPP, batch normalization and ReLU after convolutions, and sigmoid at the end of the network. Figure 2 shows the proposed network.

4. Calibration

The main purpose here is to prove that we need a full image or most of it and not just a small region. The calibration also helps specify the length of the output vector.

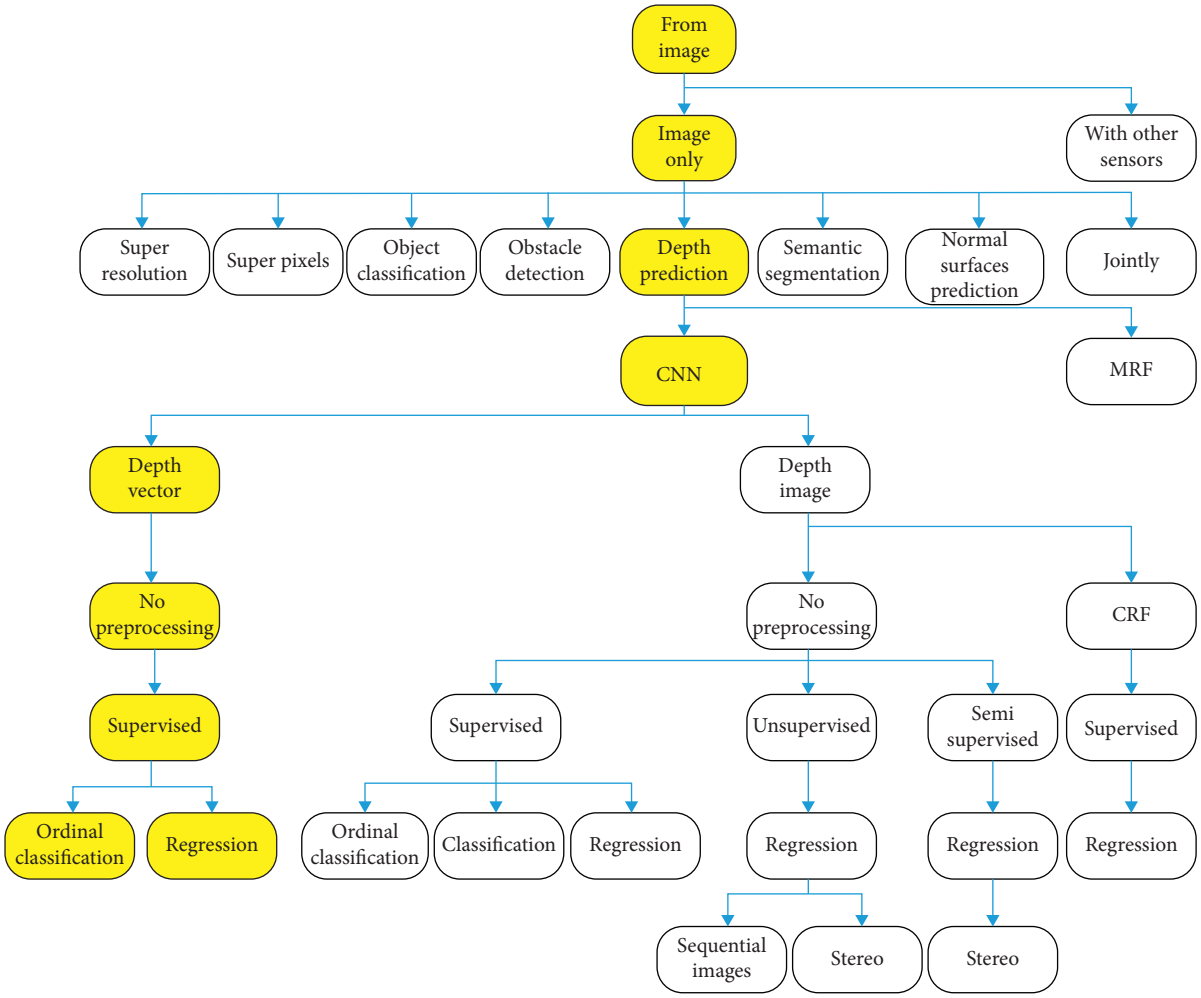


Figure 1: Tree of major problems related to depth prediction.

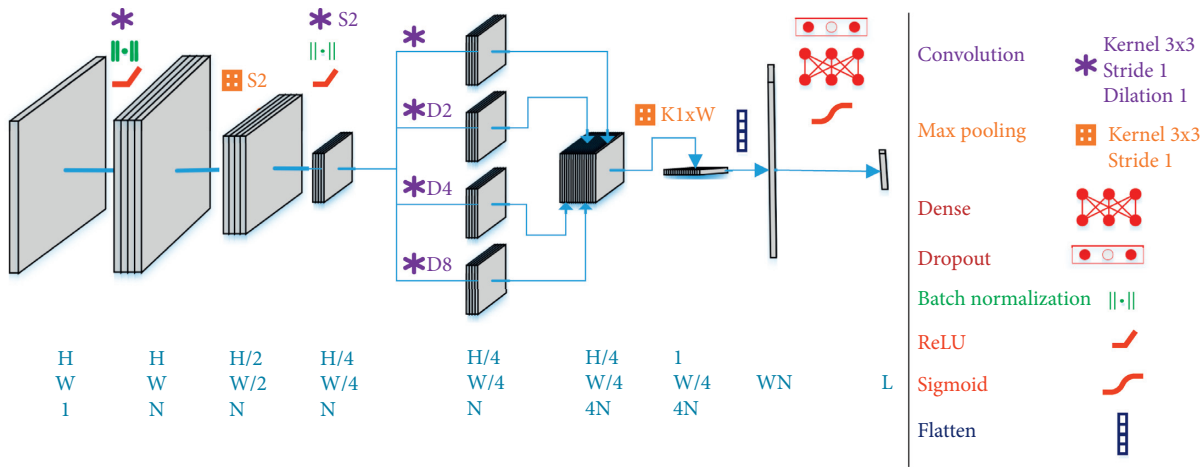


Figure 2: The proposed network. $N = 64$, $H = 64$, $W = 160$, and $L = 80$.

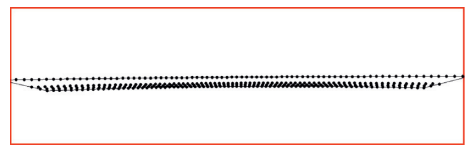
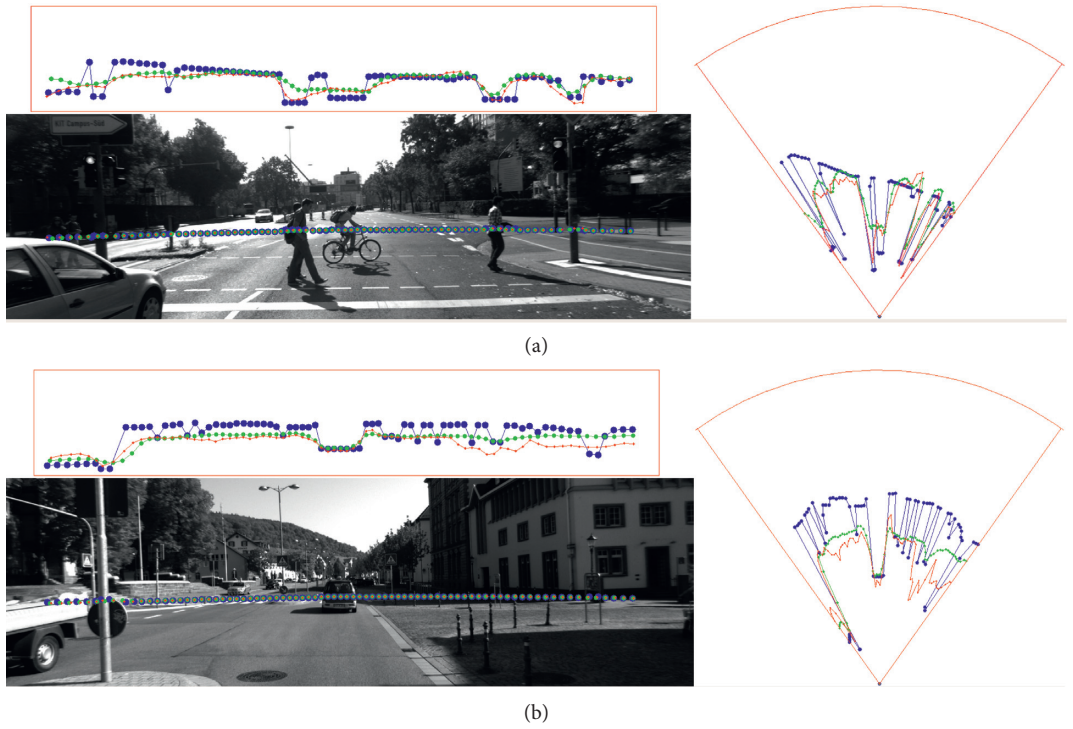


Figure 3: The region of targets inside the image in the KITTI dataset. The higher arc is at 4 meters, while the lower one is at 60 meters. There are 80 laser points per line.

Table 1: Results on KITTI test dataset.

		Small is better					Large is better				
Network	Params	Seq Rel	Abs Rel	Scale invar.	RMSL	RMS	EVAR	δ_3	δ_2	δ_1	FPS
	—	0.076	0.465	0.105	0.468	0.141	0.000	0.872	0.692	0.290	—
VGG19	58.1 M	**0.016**	**0.152**	**0.029**	**0.240**	**0.079**	**0.693**	0.974	**0.926**	**0.777**	12
Proposed	**3.2 M**	0.018	0.176	0.031	0.251	0.082	0.672	**0.976**	0.916	0.724	**40**
DORN	96.8 M	0.020	0.172	0.033	0.256	0.085	0.617	0.969	0.920	0.750	6
ResNet50	23.7 M	0.021	0.180	0.035	0.266	0.085	0.636	0.966	0.907	0.733	21
FCRN	33.7 M	0.025	0.201	0.053	0.324	0.092	0.577	0.957	0.888	0.702	10



(a)

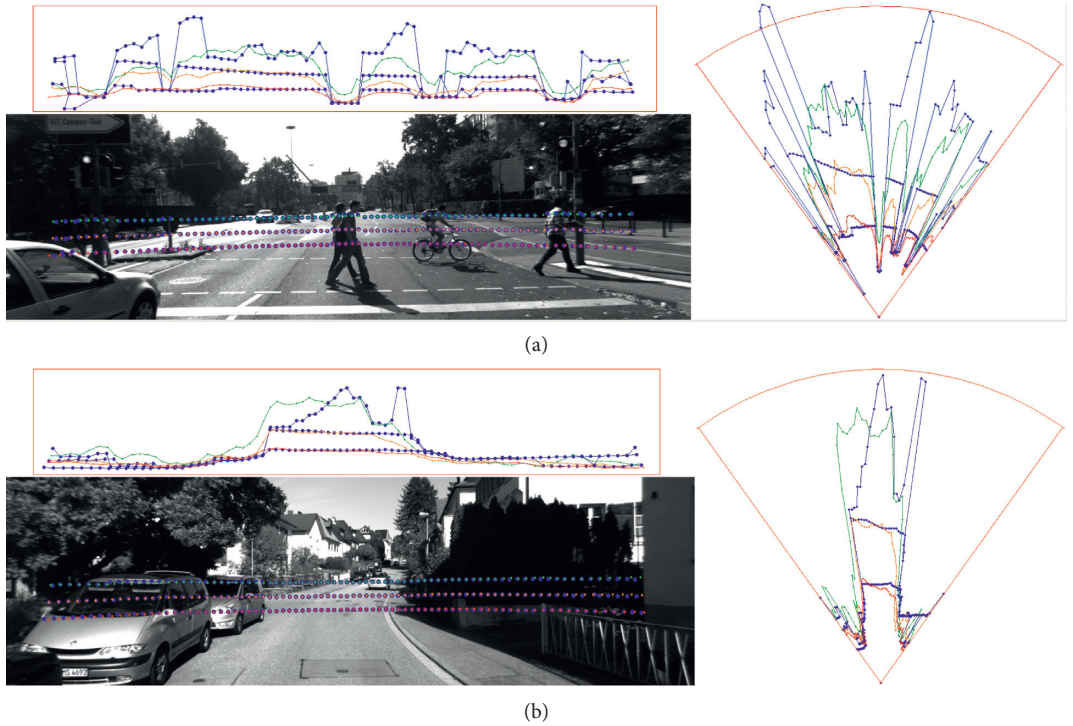
(b)

Figure 4: Comparison between VGG19 (green) and the proposed model (red) on the KITTI testing dataset. Ground truth is in blue.

From equations in [37], we project Lidar points to the image system. If we fix the distance and change the angle, the points will form a horizontal arc. In Figure 3, we could see that a vector of depths is projected to a large region in the image. This means we need a region that is equal to or larger than this region as an input.

5. Experimental Tests

For the outdoor environment, we used the KITTI [37, 38] dataset. KITTI provides four images for each target. The target is usually a depth image built by the projection from the Lidar cloud to a single image. We only need vector depths around the car. Therefore, we use the raw targets and derive a vector depth. The cloud is stored in raw format as a spiral beginning from forward high and looping until it reaches the earth. The cloud map has 64 levels, but there are many problems in the storage process. The files have different sizes, because the infinites and unavailable readings were not stored. Some sensors read data from neighbouring sensors. We have two ways to cut a desired level. We convert the points to spherical coordinates (r, ϕ, θ). The first method is by observing the difference $\Delta\phi$. When $\Delta\phi$ turns negative with a huge discontinuity $\pi/2$, we jump to another level. The



(a)

(b)

Figure 5: Displaying multilayer prediction for the proposed model (green, orange, and red) on the KITTI testing dataset. Ground truths are in blue. Level 10 is the higher one.

second method is by choosing a desired θ_d with a small region around it, $\theta_d \pm \varepsilon$, and then increasing the ϕ in fixed step and taking the nearset θ to θ_d. In this paper, the second method is used with $\theta_d = 92.8^\circ$ which is near level 15. The nearest point at the fixed ϕ is only taken.

As opposed to depth image, when most of the image is masked out, depth vector masks out less than 0.5%. We use all the provided datasets including city, residential, person, campus, and road. We distribute them amongst training and testing sets randomly according to the drive index. The samples number is 38616 for training and 9273 for testing.

We tested the following networks: VGG19, ResNet50, FCRN, DORN, and our proposed model. The trained ResNet50 are used as initial weights for FCRN. We use berHu loss function [8] for all networks, except DORN, which uses ordinal classification [10]. Adam optimizer is used with a learning rate $1e-4$ and decay coefficient $1e-5$.

The images are resized to 160×96 and then cropped to 160×64, while the output labels per image are 80 laser points and the batch size is 64.

Table 1 shows the results. We note that VGG19 gives the best accuracy in most terms, but it is slow and consumes a lot of memory. The proposed network has the highest frame rate (40 FPS) and needs less memory. DORN gives good results, especially at low ranges, but it is very slow and consumes a lot of memory.

To show the results, we projected the targets and predicted distances (as points) to the input images. In spherical coordinate, target distances $r \in [4\,\text{m}, 80\,\text{m}]$, $\phi \in [-40^\circ, 40^\circ]$, and $\theta = 92.8^\circ$. We also plotted the polar and forward views of these points.

In Figure 4, the 1st sample, the proposed model, predicts people's depths better than the VGG19. However, it still fails to predict the small road sign on the left. The car, two persons, and a road lamp appear as 4 local minima. In the 2nd sample, the high discontinuity in the labels is filtered by the networks.

We used the trained proposed networks to train two further networks to predict level 10 (90.8°) and level 20 (94.8°). We obtained better results than expected. Then, we retrained on level 15 by taking level 10 network as initial weights. The network with these 3 stages of training yielded better results than using a single stage of training. Usually, more targets lead to better results; but this way, we could accumulate the experience of multiple levels on one level and get results near the depth image results. In fact, the whole network is used to predict this small depth vector instead of the whole image matrix. Figure 5 shows some results on samples from the testing dataset. As a result, we could build a 3D Lidar from a 2D Lidar by tilting its angle, collecting a new dataset, and training a small network with it.

6. Conclusion

We conclude that we can build a 2D Lidar from a camera. We could generalize a 2D Lidar to a 3D Lidar. We benefit from training on multiple levels to boost a depth vector prediction on a certain level. In general, cameras are cheaper than Lidars and easier to handle with software. They can be installed more comfortably. The transformation matrix from the Lidar system to the camera system is stored inside the model. Only a camera calibration is needed when using a

new camera. The results are affected slightly by changing the camera's height or angles slightly. We have better synchronization because we predict the depths for each image, instead of synchronizing each image with the corresponding Lidar points. In general, CNNs have 2 main parts: the encoder and the decoder. The encoder does not change when predicting depth vector, but the decoder becomes much smaller and faster to learn and test. We cannot predict points at the extreme left and extreme right of the image, because the labels with very small distances could be outside the image. Finally, we were able to perform depth predictions with a small network (4 layers) and achieve good performance in terms of accuracy and execution time using a CPU. VGG gives remarkable accuracy compared to other networks as well as for DORN, but with lower execution time and more memory consummation.

Future work includes using the trained models on one level and a camera to obtain better predictions on all levels as unsupervised learning.

Data Availability

The data used in this study are available at the following website, source code: https://github.com/NadimArubai/BuildingARealtime2DLidarUsingDeepLearning. The KITTI dataset is available at http://www.cvlibs.net/datasets/kitti/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. C. Platt, Eds., pp. 1161–1168, MIT Press, London, UK, 2006.

[2] A. Saxena, M. Sun, and A. Ng, "Make3D: Depth perception from a single still image," *AAAI*, vol. 2008, 2008.

[3] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1253–1260, New York, NY, USA, 2010.

[4] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin, "Fast automatic single-view 3-d reconstruction of urban scenes," *ECCV*, vol. 2008, 2008.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012*, Lake Tahoe, NV, USA, 2012.

[6] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *NIPS*, vol. 2014, 2014.

[7] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2650–2658, London, UK, 2015.

[8] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV)*, pp. 239–248, London, UK, 2016.

[9] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3174–3182, 2018.

[10] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2002–2011, New York, NY, USA, 2018.

[11] H. Ren, M. El-Khamy, and J. Lee, "Deep robust single image depth estimation neural network using scene understanding," *CVPR Workshops*, vol. 2019, 2019.

[12] R. Garg, B. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: geometry to the rescue," *ECCV*, vol. 2016, 2016.

[13] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6602–6611, London, UK, 2017.

[14] T. Zhou, M. Brown, N. Snavely, and D. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6612–6619, Berlin, Germany, 2017.

[15] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: bridging the gap in 3D object detection for autonomous driving," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8437–8445, London, UK, 2019.

[16] Y. You, Y. Wang, W.-L. Chao et al., "Accurate depth for 3D object detection in autonomous driving," 2020.

[17] Y. Kuznietsov, J. Stückler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2215–2223, New York, NY, USA, 2017.

[18] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5162–5170, New York, NY, USA, 2015.

[19] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016.

[20] B. Li, C. Shen, Y. Dai, A. Hengel, and M. He, "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1119–1127, Taiwan, China, 2015.

[21] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," *ECCV*, vol. 23, 2018.

[22] P. Wang, X. Shen, Z. L. Lin, S. Cohen, B. L. Price, and A. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2800–2809, London, UK, 2015.

[23] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Stefano, "Geometry meets semantics for semi-supervised monocular depth estimation," 2018.

[24] H. Zhan, C. S. Weerasekera, R. Garg, and I. Reid, "Self-supervised learning for single view depth and surface normal estimation," in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, pp. 4811–4817, London, UK, 2019.

[25] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 89–96, New York, NY, USA, 2014.

[26] F. Ma and S. Karaman, "Sparse-to-Dense: depth prediction from sparse depth samples and a single image," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, New York, NY, USA, 2018.

[27] Z. Xia, P. Sullivan, and A. Chakrabarti, "Generating and exploiting probabilistic monocular depth estimates," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 62–71, New York, NY, USA, 2020.

[28] N. Wadhwa, R. Garg, D. E. Jacobs et al., "Synthetic depth-of-field with a single-camera mobile phone," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–13, 2018.

[29] R. Garg, N. Wadhwa, S. Ansari, and J. Barron, "Learning single camera depth estimation using dual-pixels," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7627–7636, Berlin, Germany, 2019.

[30] Y. Zhang, N. Wadhwa, S. Orts-Escolano, C. HÃ€ne, S. Fanello, and R. Garg, "Du2net: learning depth estimation from dual-cameras and dual-pixels," 2020.

[31] O. Krestinskaya and A. P. James, "Real-time analog pixel-to-pixel dynamic frame differencing with memristive sensing circuits," 2018.

[32] L. Chua and L. Yang, "Cellular neural networks: theory," *Circuits and Systems*, vol. 35, pp. 1257–1272, 1988.

[33] P. Arena, A. Basile, M. Bucolo, and L. Fortuna, "An object oriented segmentation on analog CNN chip," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 7, pp. 837–846, 2003.

[34] P. Foldesy, G. Linan, A. Rodriguez-Vazquez, S. Espejo, and R. Dominguez-Castro, "Object oriented image segmentation on the CNNUC3 chip," 2000.

[35] J. Michels, A. Saxena, and A. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," 2005.

[36] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, London, UK, 2017.

[37] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the KITTI dataset," *IEEE Access*, vol. 32, pp. 1231–1237, 2013.

[38] A. Geiger, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012.