Machine Learning and Artificial Intelligence
J.-L. Kim (Ed.)

doi:10.3233/FAIA220423

SD-Depth: Light-Weight Monocular Depth Estimation Using Space and Depth CNN for Real-Time Applications

Hatem IBRAHEM[a], Ahmed SALEM[a], and Hyun-Soo KANG[a,1]
[a] *School of Information and Communication, Chungbuk National University, Korea*

Abstract. With the help of the space-to-depth and depth-to-space modules, we provide a convolutional neural network design for depth estimation. We show designs that down sample the spatial information of the picture utilizing space-to-depth (SD) as opposed to the widely used pooling methods (Max-pooling and Average-pooling). The space-to-depth module may shrink the image while maintaining the spatial information of the image in the form of additional depth information. This technique is far superior to Max-pooling, which diminishes the image's information and features. We also suggest a lightweight decoder step that builds a high-resolution depth map out of many low-resolution feature maps using the depth-to-space (DS) module. The suggested architecture effectively learns depth estimation with high processing speed and accuracy. We trained and evaluated our suggested model on NYU-depthV2 dataset and attained low error values (RMSE= 0.342) and high delta accuracies (δ_3=0.996) at a fast-processing speed (25Fps).

Keywords. Depth estimation, Convolutional neural networks, Real-time processing.

1. Introduction

The majority of convolutional neural network (CNN) architectures use Max-pooling (MP) to compress the feature space to a more condensed representation, but MP introduces information loss as they do so because the important information only exists in the window's maximum value, which is used to slide over the input data. Due to the information lost during the pooling process, certain pooling algorithms provide a lossy compressed representation, which has a severe impact on the entire learning process utilizing the neural network design. The suggested down-sampling methodology lowers the input features' spatial size, but it adds the spatial reduction as additional depth channels using a convolutional learnable technique that keeps the same amount of information. The depth-to-space (DS) module [1] and the space-to-depth (SD) [2] module [2] were first suggested for the image super-resolution task. Recent developments in CNN designs have demonstrated their improved performance on a variety of computer vision applications, particularly the depth estimation task. In robotics, 3D image interpretation, medical diagnosis, virtual/augmented reality, and self-driving cars, depth estimate is a crucial problem. For those applications, performing depth estimate with high speed and accuracy is therefore quite helpful.

[1] Corresponding Author, Hyun-Soo KANG, School of Information and Communication, Chungbuk National University, Korea; E-mail: hskang@cbnu.ac.kr.

Following is a summary of our contribution in this paper:

- We provide a new convolutional neural network architecture that is built on the efficient SD and DS modules that can effectively learn depth estimation tasks.
- We compare the performance of the proposed encoder architecture with convolutional architecture with MP.

An overview of the proposed CNN architecture using SD and DS modules is shown in Figure 1. The SD-CNN compresses the input image features to many low-resolution feature maps, the DS module up-samples the representation to form the depth map.

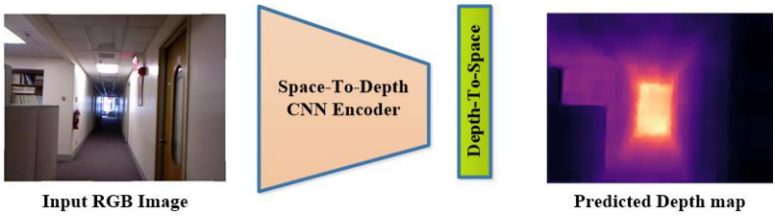

Figure 1. Overview of the proposed CNN architecture using SD and DS modules.

2. Related work

The depth estimation is mostly performed by predicting a per-pixel label or continuous value. To estimate depth from stereo pair pictures or from a single image, several depth estimation techniques using various CNN architectures were recently introduced. A coarse prediction CNN network was suggested by Eigen et al. [3] to predict an initial coarse depth map and improve it using a different CNN network. A cascade of several continuous random fields (CRF) was proposed by Xu et al. [4] to integrate the output of multiple CNNs. A monocular depth estimation approach employing relative depth maps of certain pairwise comparison matrices was suggested by Lee et al. [5]. A shift- and scale-invariant depth map is predicted for each semantic portion of the image using the divide-and-conquer network which was proposed by Wang et al. [6]. A depth estimation technique applying an attention module (SharpNet) to the outlines of the objects in the image that are occluded, was proposed by Ramamonjisoa et al. [7]. To maximize depth estimation utilizing U-Net architecture, Wu et al. [8] presented PhaseCam3D, a depth estimation technique based on optimizing a mask on the camera aperture plane. From Big to Small (BTS), a CNN encoder-decoder that employs unique local planar guiding layers in the decoding step for precise depth estimates, was suggested by Lee et al [9]. To achieve accurate depth estimation, Yin et al. [10] suggested an encoder-decoder CNN for depth estimation based on imposing a geometric restriction of the virtual normal direction on the 3D structure. An encoder-decoder transformer-based architectural block that separates the depth range into bins with estimated centers depending on the image was proposed by Bhat et al [11]. Patil et al. [12] recently proposed a two head CNN network for depth estimation, the first head produces pixel-wise plane values while the second head produces a dense vector field for each pixel position. The output from the second head is adaptively fused by the output of the first head.

The DS module was employed in recent CNN-based depth methods [13-14] but the SD module was not presented as a down-sampling technique like we propose in this research. The suggested architecture outperforms state-of-the-art (SOTA) approaches for depth estimation, despite being simpler and less sophisticated than the SOTA methods.

3. Proposed method

The proposed method is built around two main blocks: the SD layer, which acts as a down-sampling module like the pooling layers, and the DS layer, which acts as a decoder stage to merge the feature depth in order to up-sample the feature maps to form the dense map at the same size as the input. Wang et al. [2] introduced the Space-to-Depth (SD) module as a method of producing a dense representation of the optical flow for use in video super-resolution. We use it as a learnable spatial down-sampling layer in our suggested technique, comparable to the pooling method. The SD module differs from pooling in that no feature compression occurs on the input feature maps, but the decrease in spatial size is transformed into depth data via pixel aggregation. This pixel aggregation is accomplished by turning input feature maps of shape rW × rH × C into feature maps of shape W × H ×C × r^2 using an aggregation process that can be described mathematically as shown in equation (1).

$$Y^{W \times H \times C \times r^2} = W_L \times F^{L-1}(X^{rW \times rH \times C}) + b_L \quad (1)$$

where Y and X are the DS layer's low-resolution output with extended depth channel and high-resolution input, respectively. W_L and b_L are the DS layer's weights and biases, W is the image width, H is the image height, C is the image channels, r is the image depth, and F is the layer's activation function. This layer is used five times in our suggested designs, each time reducing the spatial dimension by $r=2$ in width and $r=2$ in height and increasing the depth by four times $r^2 = 4$. Depending on the architecture, convolutional layers, Relu, and batch normalization are applied in a different sequence each time the input image is down sampled. Shi et al. [1] introduced the depth-to-space (DS) module as a method of aggregating the pixels of the input features to generate a higher resolution image for the image super-resolution task. We use it as a one-stage up-sampling decoder in our suggested technique. Pixel aggregation is accomplished by turning the encoder's input feature maps of shape $W \times H \times r^2$ into a dense map of form $rW \times rH$ via a learnable process that can be described mathematically as the opposite of equation (1) as in equation (2). A graphical illustration of the SD and DS is shown in Figure 2.

$$Y^{rW \times rH} = W_L \times F^{L-1}\left(X^{W \times H \times r^2}\right) + b_L \quad (2)$$

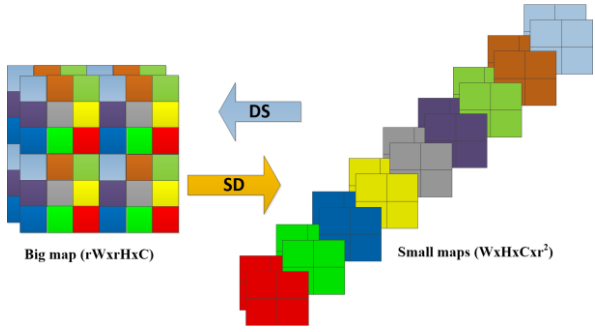

Figure 2. The two main modules in our proposed method. a) Space-to-Depth (SD) which is used to down sample the input feature map of size rW×rH×C to lower resolution map of size W×H×C×r². b) Depth-to-Space (DS) which is used as the decoder stage in our method to up-sample the input low-resolution feature map of size W × H × r2 to a higher resolution map of size rW × rH.

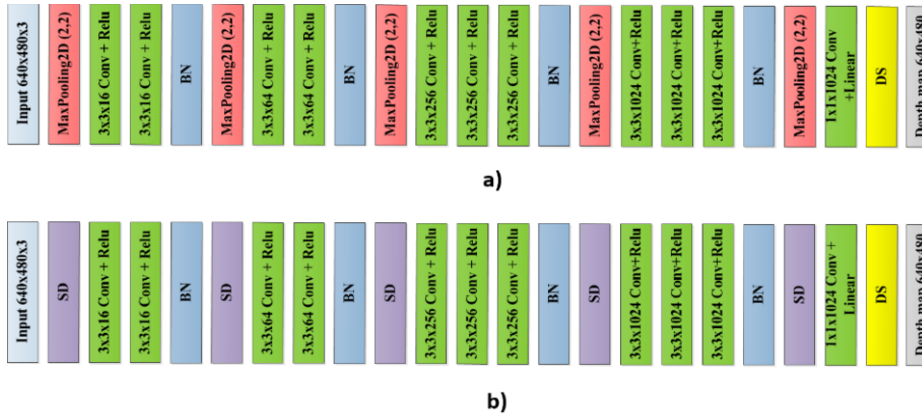

Figure 3. The proposed architectures for comparison. (a) MP-CNN: Max-pooling-based CNN architecture for down-sampling. (b)SD-CNN: Space-to-Depth (SD)–based CNN architecture for down-sampling. In both architecture DS module is used to construct the high-resolution depth map from the down-sampled features.

We present two designs using distinct down-sampling strategies, evaluate their performance, and emphasize the benefits of each. We propose a basic CNN that uses max-pooling (MP-CNN) to minimize the spatial size of the input features, using two or three convolutional layers that are repeated with "Relu" activation and batch normalization (BN). The feature depths through the down-sampling stages are 3, 16, 64, 256, and 1024, and the final dense map is created from 1024 low-resolution features produced by 1 × 1 convolutional layer with a size of 32×W and 32×H using the DS decoder, as shown in Figure 3-a. The second CNN design is SD-Net, which has the same architecture as the first with MP but uses the SD layer instead of MP to down sample the spatial size of the input and expand the depth of the output features, as illustrated in figure 3-b. In section 4 (Experimental Results), we compare the performance of the two suggested architectures, demonstrating that SD-Net has much higher accuracy in the depth estimation task than the MP-Net. The Huber loss (a function that selectively acts either like L_1 loss or L_2 loss depending on a threshold value "t") is the loss function used for learning the depth estimation. It is mathematically stated as in equation (3).

$$L = \begin{cases} \frac{1}{2r^2HW} \sum_{x=1}^{W} \sum_{y=1}^{H} (I_{x,y} - \tilde{I}_{x,y})^2 , if \ |I - \tilde{I}| < t \\ \frac{t}{2r^2HW} \sum_{x=1}^{W} \sum_{y=1}^{H} (|I_{x,y} - \tilde{I}_{x,y}| - \frac{1}{2}t) , otherwise \end{cases} \quad (3)$$

where I is the ground truth pixel value and $\tilde{I}$ is the predicted pixel value, and "t" is set to 1 since it experimentally speeds up the training process. L_1 and L_2 are also evaluated individually in two distinct tests for the suggested technique training, however, each one experienced a slow loss improvement problem at some point throughout the training.

4. Experimental results

The proposed architectures are trained and tested on NYU-depthV2, which consists of 1449 labeled photos of indoor scenes (bedrooms, living rooms, kitchens, bathrooms, and workplaces) and their corresponding depth maps taken by the Microsoft Kinect sensor. The dataset has 795 training photos and 654 test images. We train and test our model on

a desktop computer with an Nvidia RTX3090 GPU with Ampere 24 GB memory, an Intel Core i7-8700 CPU with a clock speed of 3.20 GHz and 64 GB RAM, and a Tensorflow Keras environment for 1500 epochs with Adam's optimizer.

The training and test image sizes are 640×480. We evaluate the depth estimation performance using the absolute relative error (REL $= \frac{1}{N}\sum_{P}^{N}\frac{|y_p - y_p|}{y_p}$), the root mean squared error (RMSE $= \sqrt{\frac{1}{N}\sum_{P}^{N}(y_p - y_p)^2}$), and the delta accuracy $\delta = \max\left(\frac{y}{\tilde{y}}, \frac{\tilde{y}}{y}\right) < t$ for t threshold values 1.25, 1.25^2, and 1.25^3, where y and $\tilde{y}$ are the ground truth and predicted pixel values, respectively, and N is the number of pixels in the depth map.

Table 1. Evaluation of the results obtained by the proposed method in terms of REL, RMSE, δ accuracy.

Model	Param. count	REL	RMSE	δ_1	δ_2	δ_3	FPS
MP-CNN	23,668,624	0.0947	0.3852	0.9287	0.9815	0.9946	26.3
DS-CNN	34,363,552	0.0924	0.3421	0.9301	0.9824	0.9955	25.0

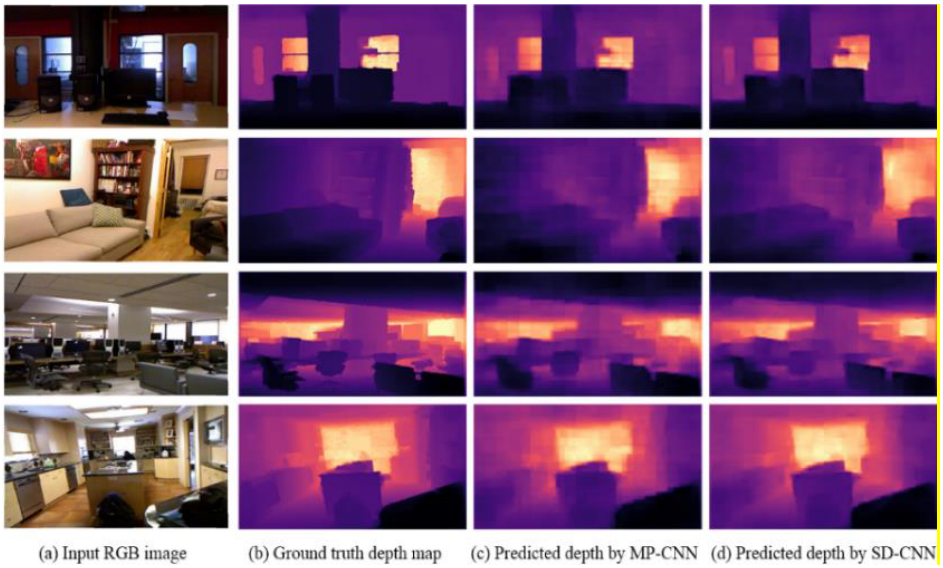

Figure 4. Results obtained by the proposed architectures. (a) Input RGB image. (b) Ground truth depth map. The predicted depth maps by MP-CNN and SD-CNN are shown in (c) and (d), respectively.

As shown in Table 1, SD-Net attained better errors (REL of 0.0924 for SD-Net versus 0.0947 MP-Net) and accuracy values but at a lower speed than MP-CNN (25.0 versus 26.3 fps). By comparing the visual quality in Figure 4, the predicted depth by SD-CNN is better than that for MP-CNN. MP-CNN introduces a severe blocking effect due to the nature of the Max-pooling process, which introduces data loss, while it is less in the case of SD-CNN as no data loss happens. Table 2 shows a comparison between the proposed models and the recent methods on the NYU-depth V2. SD-CNN is the best model in terms of REL, RMSE, and δ_1 although its architecture is simple.

Table 2. Comparison between the proposed architectures and the recent methods of depth estimation in terms of REL, RMSE, δ accuracy.

Model	REL	RMSE	δ_1	δ_2	δ_3
Eigen et al. [3]	0.158	0.641	0.769	0.950	0.988
Xu et al. [4]	0.121	0.586	0.811	0.954	0.987
Lee et al. [5]	0.131	0.538	0.837	0.971	0.994
SharpNet [7]	0.139	0.502	0.836	0.966	0.993
Yin et al. [10]	0.108	0.416	0.875	0.976	0.994
BTS [9]	0.110	0.392	0.885	0.978	0.994
SDC-Depth [6]	0.128	0.497	0.845	0.966	0.990
PhaseCam3D [8]	0.093	0.382	0.932	**0.989**	**0.997**
Adabins [11]	0.103	0.364	0.903	0.984	**0.997**
P3Depth [12]	0.104	0.356	0.898	0.981	0.996
MP-CNN (ours)	0.094	0.385	0.928	0.981	0.994
SD-CNN (ours)	**0.092**	**0.342**	**0.930**	0.982	0.996

5. Conclusion

The suggested technique uses the powerful and fast SD module for lossless image down-sampling in the encoder stage and the fast DS module for up-sampling in the decoder stage, it can efficiently learn the depth estimation problem (RMSE=0.342 and δ 3=0.996). The effectiveness of the proposed technique was demonstrated in the results section showing that it can work at a fast speed (25 fps) suitable for real-time applications.

Acknowledgement

This work was supported by the Ministry of Science and ICT, Korea, under the Grand Information Technology Research Center support program (IITP-2022-2020-0- 01462) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation) and in part by the Research Projects of “Development of automatic screening and hybrid detection system for hazardous material detecting in port container” funded by the Ministry of Oceans and Fisheries.

References

[1] Shi W, Caballero J, Huszár F, Totz, J, Aitken AP, Bishop R, Rueckert D, Wang Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.

[2] Wang L, Guo Y, Lin Z, Deng X, An W. Learning for video super-resolution through HR optical flow estimation. In Proceedings of the Asian Conference on Computer Vision (ACCV), Perth Western, Australia, 2-6 December 2018; pages 514–529.

[3] Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network. Advances In Neural Information Processing Systems, 2014, 27, 2366–2374.

[4] Xu D, Ricci E, Ouyang E, Wang X, Sebe N. Monocular depth estimation using multi-scale continuous CRFS as sequential deep networks. IEEE Trans. Pattern Anal. Mach. Intell., 41(6):1426–1440.

[5] Lee J, Kim C. Monocular depth estimation using relative depth maps. In Proceedings of the IEEE Conference on Computer Vision Pattern Recognition (CVPR), Long Beach, CA, USA, 16-20 June 2019 pages 9721–9730.

[6] Wang L, Zhang J, Wang O, Lin Z, Lu H. SDC-depth: Semantic divide-and-conquer network for monocular depth estimation. In proceedings of the IEEE Conference on Computer Vision Pattern Recognition (CVPR), WA, USA, 14–19 June 2020. pages 538–547.

[7] Ramamonjisoa M, Lepetit V. SharpNet: Fast and accurate recovery of occluding contours in monocular depth estimation. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–3 November 2019, pages 2109–2118, 2019.

[8] Wu Y, Boominathan V, Chen H, Sankaranarayanan A, Veeraraghavan AS. Phasecam3D: Learning phase masks for passive single view depth estimation. In 2019 IEEE International Conference on Computational Photography (ICCP), Tokyo, Japan, 15-17 May 2019, pages 1–12.

[9] Lee J, Han M, Ko DW, Suh IH. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326, 2019.

[10] Yin W, Liu Y, Shen C, Yan Y. Enforcing geometric constraints of virtual normal for depth prediction. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–3 November 2019, pages 5683–5692, 2019.

[11] Bhat SF, Alhashim I, Wonka P. Adabins: Depth estimation using adaptive bins. In proceedings of the IEEE Conference on Computer Vision Pattern Recognition (CVPR), Virtual, 19–25 June 2021, pages 4009–4018.

[12] Patil V, Sakaridis C, Liniger A, Van G. P3Depth: Monocular depth estimation with a piecewise planarity prior. In proceedings of the IEEE Conference on Computer Vision Pattern Recognition (CVPR), New Orleans, Louisiana, USA, 19–24 June 2022. pages. 1600-1611.

[13] Ibrahem H, Salem A, Kang HS. DTS-depth: real-time single-image depth estimation using depth-to-space image construction. Sensors. 2022; 22(5):1914.

[14] Ibrahem H, Salem A, Kang HS. RT-ViT: Real-time monocular depth estimation using lightweight vision transformers. Sensors. 2022; 22(10):3849.