Sensors and Materials, Vol. 36, No. 6 (2024) 2569–2583
MYU Tokyo

S & M 3687

Monocular Depth Estimation of 2D Images Based on Optimized U-net with Transfer Learning

Ming-Tsung Yeh,[1] Tsung-Chi Chen,[2] Neng-Sheng Pai,[1*] and Chi-Huan Cheng[2]

[1]Department of Electrical Engineering, National Chin-Yi University of Technology,
57, Sec. 2, Zhongshan Rd., Taiping Dist, Taichung 411030, Taiwan
[2]Department of Electrical Engineering, National Changhua University of Education,
No. 1, Jinde Rd., Changhua City, Changhua County 50007, Taiwan

(Received December 13, 2023; accepted May 22, 2024)

Keywords: depth estimation, transfer-learning-based U-net, convolutional autoencoder, depth classification

Estimating depth from 2D images is vital in various applications, such as object recognition, scene reconstruction, and navigation. It offers significant advantages in augmented reality, image refocusing, and segmentation. In this paper, we propose an optimized U-net network based on a transfer learning encoder and advanced decoder structures to estimate depth on a single 2D image. The encoder–decoder architecture is built from ResNet152v2 as the encoder and an improved U-Net-based decoder to achieve accurate depth predictions. The introduced ResNet152v2 network had been pretrained on the extensive ImageNet dataset, which possesses weights to extract rich and generalizable features for large-scale image classification. This proposed encoder can have prior knowledge to reduce training time and improve object position recognition. The proposed composite up-sampling block (CUB) designed in the decoder applied the 2x and 4x bilinear interpolation combined with the one-stride transpose convolution to expand the low-resolution feature maps obtained from the encoder, enabling the network to recover finer details. The skip connections are used to enhance the representation power of the decoder. The output of each up-sampling block is concatenated with the corresponding pooling layer. This fusion of features from different scales helps capture local and global context information, contributing to more accurate depth predictions. This method utilizes RGB images and depth maps as training inputs from the NYU Depth Dataset V2. The experimental results demonstrate that the transfer learning-based encoder, coupled with our proposed decoder and data augmentation techniques, enables the transformation of complex RGB images into accurate depth maps. The system accurately classifies different depth ranges based on depth data ranging from 0.4 to 10 m. By mapping different depths to corresponding colors using gradational color scales, precise depth classification can be performed on the 2D images.

*Corresponding author: e-mail: pai@ncut.edu.tw
https://doi.org/10.18494/SAM4822

ISSN 0914-4935
https://myukk.org/

1. Introduction

Many applications such as scene recognition, 3D image reconstruction, robotics, and navigation require depth images, also known as depth maps, to perform object depth classification and estimation from captured images. The depth image provides the distance information of the objects in the scene, which is the z-axis distance from the camera viewpoint. Depth maps can be obtained using a special 3D depth camera that includes two lenses and is combined with triangulation algorithms such as proactive or passive stereo vision, structured light, and time of flight. To realize the three-dimensional structure in the captured image, depth maps can provide the spatial layout information of the objects presented to achieve a visual solid geometry. However, 3D cameras are expensive, and monocular cameras take most pictures, which are more commonly used in daily activities. These types of 2D images are implemented to produce depth images that are more useful in various applications.

The monocular depth estimation applies only a single 2D image and various image vision algorithms to estimate depth and reconstruct depth maps. The deep-learning-based methods are popular approaches to performing monocular depth estimation and have obtained notable outcomes lately. Eigen *et al.* used two convolutional neural networks (CNNs) to obtain coarse global prediction by the first one and another refined that locally.[1] Li *et al.* and Liu *et al.* applied a deep CNN combined with conditional random fields-based regularization to estimate depths from single monocular images.[2–4] They employed CNNs to learn the relationship knowledge between image pixels or super-pixels and depth. Iro *et al.* proposed end-to-end fully convolutional networks incorporating residual up-sampling blocks to tackle high-dimensional regression problems and efficiently train their model to obtain better depth maps.[5] There are some problems raised by this type of deep CNN, which needs many datasets to train the network, has slow convergence, and has intolerable low-resolution feature maps. Fu *et al.* proposed a deep ordinal regression network to redefine learning as an ordinal regression problem.[6] They used the regression loss to train the network for faster convergence and higher accuracy.[6] Su *et al.* used a piled residual CNN to generate class discriminative features of an input image and regress depth by a convolutional network.[7] However, these network models have complex hierarchical structures that increase training complications and costs.

Recently, the generative adversarial network (GAN) has shown outstanding performance in data generation in reconstructing synthetic images realistically. This model is employed in image-to-image translation fields such as image semantic segmentation and painting generation or implemented for monocular image depth estimation. Jung *et al.* proposed a GAN-based supervised approach; they used the generator network to predict depth maps and applied a discriminator to estimate the loss value between the ground truth and prediction during the training stage.[8] Zheng *et al.* designed T^2Net with two networks, including translation and task prediction parts with GAN loss to synthetic depth maps.[9] Bhatia applied a stacked conditional GAN with a multi-patch discriminator network for depth estimation and reduced the mean square error.[10] Kwak and Lee applied the unsupervised deep learning mode based on the cycle GAN to generate depth maps that use the paired left and right photos captured by the stereo camera as inputs to estimate a disparity map and predict objects distance.[11] Hendra and

Kanazawa proposed the conditional GAN architecture with three submodels that combine the global scene structure with local image information.[12] They also used a structured similarity as the loss function of the generator and refiner during the training stage to improve the prediction.[12] The GAN-based depth estimation demonstrated better outcomes in reconstructing the synthetic depth map image; nevertheless, their proposed network either needs a more complex network to train the generator and refine it or requires additional paired images to distinguish disparity. Some visual artifacts, such as blurring object contours or spurious blocking, should be solved in this model.

To tackle some challenges presented in the above-mentioned methods, we propose a simple monocular image depth estimation architecture based on an optimized U-net with a composite decoder structure. Our approach applies an encoder–decoder architecture built from pretrained ResNet152v2 as the encoder and an improved U-Net-based decoder to be capable of generating a more accurate and visually appealing depth map. Figure 1 shows some samples predicted by our proposed method, which produces depth maps from a single RGB image.

2. Proposed Method

In this section, we introduce the optimized U-net with a composite fusion decoder structure proposed in this paper. The corresponding transfer-learning-based encoder and fusion decoder, loss function, training dataset, and augmentation strategies are also explained. The NYU dataset is utilized to train the network and evaluate the performance of tackling the challenging task of estimating accurate depth maps from RGB images. The NYU dataset provides a diverse range of indoor scenes with corresponding depth ground truth, all evaluated and validated using the NYU dataset.

2.1 System framework

The overall system framework is shown in Fig. 2. We propose an encoder–decoder U-net structure incorporating a transfer-learning mode to perform precise depth classification on the

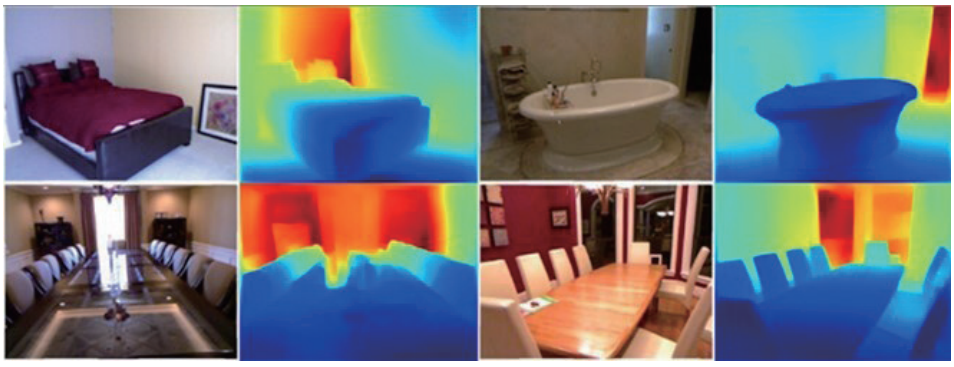

Fig. 1. (Color online) Depth maps estimated by our method.

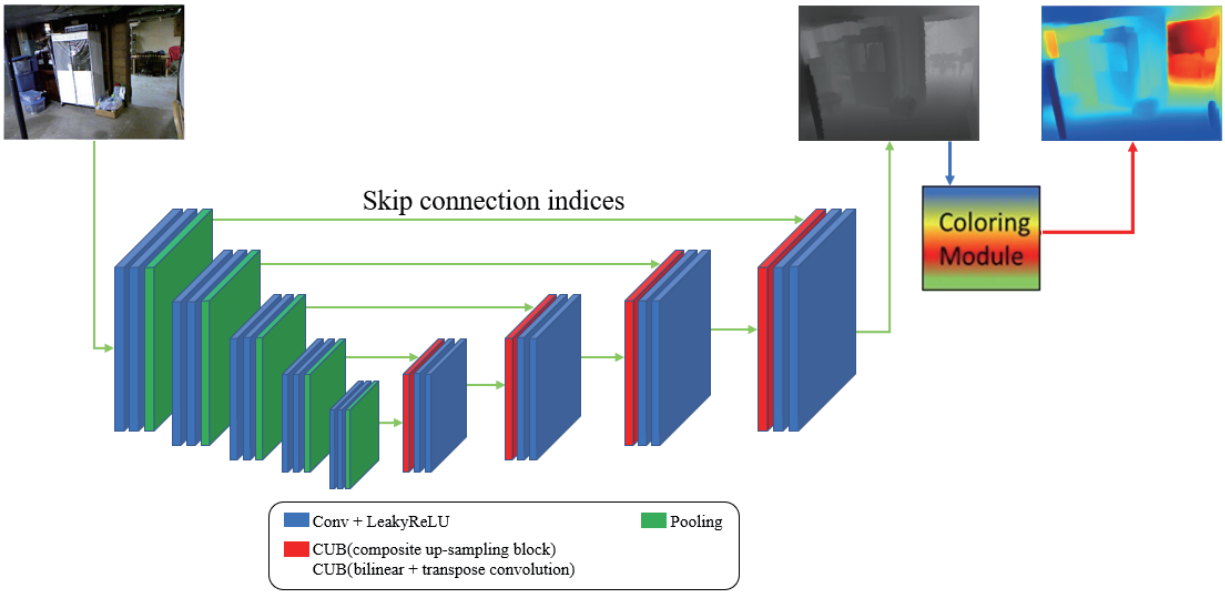

Fig. 2. (Color online) System framework of the proposed U-Net network.

2D images. This U-net includes a ResNet152V2-based encoder pretrained on the ImageNet database, which increases object recognition and feature extraction capabilities and follows an advanced decoder combined with the composite up-sampling block (CUB) structure and fusion convolution. The grayscale depth map of the prediction result applies the coloring module to map gradational color scales to different depths, allowing for an intuitive understanding of relative distance from the central viewpoint and enriching the visual representation.

2.2 Dataset

The NYU Depth v2 dataset is an invaluable resource, offering a vast collection of images and accompanying depth maps meticulously captured using cutting-edge depth cameras. Specifically designed for indoor scenes, this dataset presents a rich and diverse array of visual information, opening doors to many applications and research possibilities. This dataset holds significant potential for applications such as indoor obstacle avoidance for blind people.

This dataset has 120000 training samples and 654 testing samples; it offers essential data to train and evaluate models. For our proposed method, we specifically train on a subset of 50000 samples that have been inpainted to address missing depth values from the NYU Depth v2 dataset to help reconstruct the depth maps and ensure a complete representation of the environment captured by the depth camera. The depth maps with an upper bound of 10 m provide crucial geometric information about the scene. By utilizing this depth information, we can infer the distances of objects relative to the camera.

Throughout the training process, we employ the native resolution of the input images obtained from datasets, enabling us to preserve the crucial fine details because of keeping the original data structure without resizing. However, we down-sample the ground truth depths to

320 × 240 size to match the output resolution, ensuring compatibility between the depth maps and the predictions made by our system. We calculate the depth map estimate for the entire test image set during the testing phase. Subsequently, we upscale the estimated depth map by a factor of 2 to align it with the resolution of the ground truth and evaluate the accuracy.

2.3 Proposed network

The network architecture of the proposed optimized U-net in this paper is illustrated in Fig. 3. The proposed approach commences by encoding the input RGB image into a comprehensive set of features using the ResNet152V2 network, which has undergone pretraining on the extensive ImageNet dataset. This strategic choice empowers our network to leverage the rich and generalizable features learned from a large-scale image classification task. The encoded representations are then up-sampled through a series of layers to generate the final depth map at half the input resolution. This up-sampling process, accompanied by skip connections, forms the decoder component. Notably, our decoder does not include advanced layers, such as the batch normalization layer, because these configurations cannot improve learning convergence but increase network complication in our experiments, despite their recommendation in recent cutting-edge technology methods.

The exceptional performance achieved by our relatively simple architecture raises questions regarding the individual contributions of different components in producing high-quality depth maps. To investigate this, we conducted experiments with various cutting-edge technology

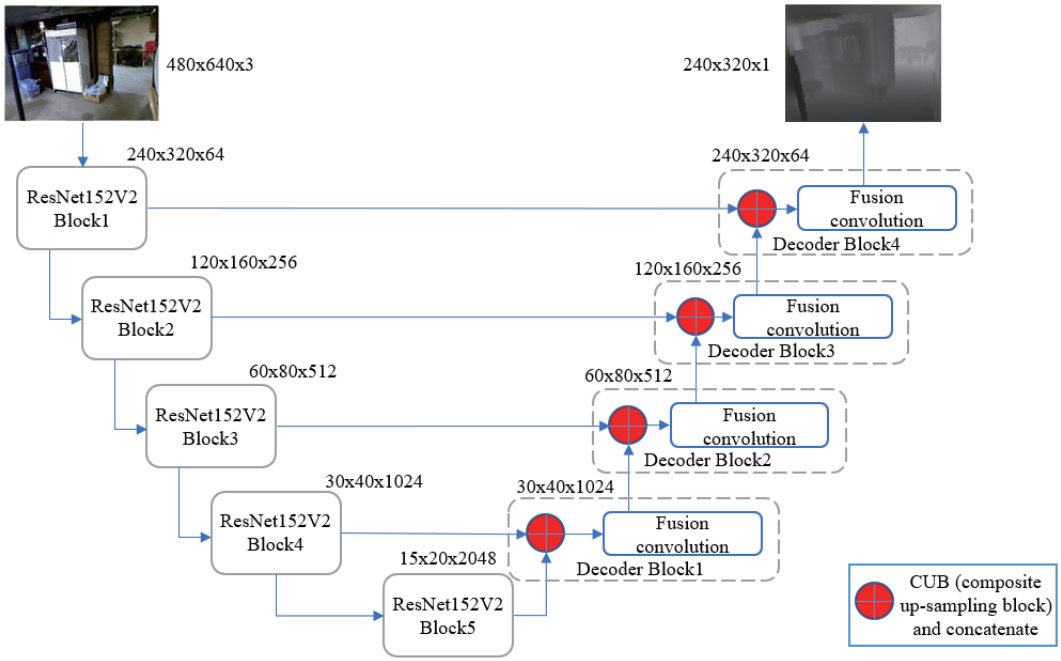

Fig. 3. (Color online) Architecture of the proposed optimized U-Net network.

encoders, including ResNet152V2 and DenseNet-169, as well as different decoder types. Our experiments demonstrate that a simpler decoder configuration can achieve outstanding results. In particular, we employ a straightforward decoder with the proposed CUB structure that performs a 2× and 4× bilinear up-sampling operation combined with transpose convolution, and the subsequent process involves applying two conventional convolutional layers. This minimalist design proves to be highly effective in generating accurate depth estimations.

In the encoding phase, the ResNet152V2 network is utilized as the backbone. The top layers initially designed for the ImageNet classification task are removed to adapt them for our depth prediction task. This modification allows us to focus solely on depth estimation. To ensure compatibility with the truncated encoder output in the decoder part, a 1 × 1 convolutional layer is applied to change the input shape that aligns with the corresponding output channels. This layer serves as the bridge between the encoded features and the subsequent up-sampling blocks. Each block consists of a 2× and 4× bilinear up-sampling operation with a transpose convolution in the CUB part and an additional design of two 3 × 3 convolutional layers in every decoder block. The output filters in these convolutional layers are set to half the number of input filters. This design choice helps to manage the complexity of the network while retaining the necessary information for accurate depth prediction.

Notably, the CUB structure is a pivotal element in the proposed U-net architecture, as shown in Fig. 4. This specialized block comprises bilinear up-sampling and one-step transpose convolution, serving as a critical innovation in this depth estimation network. In the proposed CUB structure, the initial 2× and 4× bilinear up-sampling step is pivotal in expanding the low-resolution feature maps obtained from the encoder, enabling the network to recover finer details during the depth estimation process. This up-sampling technique leverages interpolation to fill in the gaps and increase the spatial resolution of the feature maps, laying the foundation for subsequent refinement. However, the bilinear interpolation up-sampling lacks adaptability to the environment due to a fixed formula that limits performance. The additional transpose convolution has a learning capacity to follow behind the bilinear operation. It is used to optimize up-sampling actions, which allows intelligent adjustments based on varying environmental

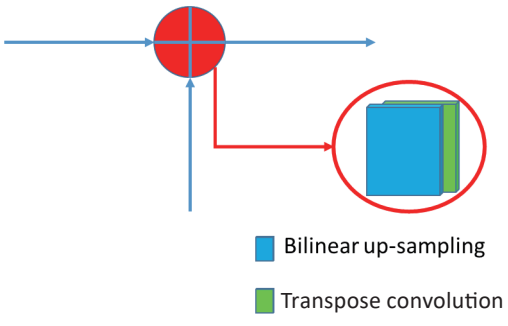

Fig. 4. (Color online) CUB structure.

conditions. The transpose convolutional layer with only one stride setting in the CUB can also help to promote network training convergence and smooth the up-scaled feature maps.

Skip connections are employed to enhance the representation power of the decoder. Each CUB and decoder block output is concatenated with the corresponding pooling layer output from the encoder, both involving the same image dimensions. This fusion of features from different scales helps capture local and global context information, contributing to more accurate depth predictions. Throughout the decoder, except for the last up-sampling block, we apply a Leaky ReLU activation function with $\alpha = 0.2$. This setting can allow a small, nonzero gradient for negative inputs, encouraging the flow of information even for negative activations. This activation function introduces nonlinearity and helps to alleviate the vanishing gradient problem during training. This preserves the visual information and allows the proposed model to leverage color cues for depth estimation. As for the target depth maps, we clip them to the range of [0.4, 10] m.

2.4 Loss functions

Selecting an appropriate loss function holds significant importance in depth regression problems, as it directly affects the training speed and the overall effectiveness of the depth estimation network. In our approach, we focus on designing a loss function that effectively captures two essential aspects: accurately reconstructing depth images and preserving intricate patterns and fine-grained information within the image representation of the depth map.

To achieve accurate depth reconstruction, our loss function minimizes the difference between the predicted depth map (denoted as $\hat{y}$) and the ground truth depth map (denoted as y). This ensures that our model learns to generate depth predictions that closely resemble the true depths of the scene. However, solely focusing on minimizing the difference between $\hat{y}$ and y may lead to overly smooth depth predictions that lack high-frequency details. To address this, a penalty term is introduced in the loss function to discourage distortions of these fine-grained intricate nuances and subtle intricacies present in the depth map. By penalizing such distortions, we encourage our model to preserve the detailed features and edges in the depth map, resulting in more visually accurate representations.

By carefully balancing the reconstruction accuracy and preservation of high-frequency details, our loss function guides the training process to find an optimal solution that produces depth predictions with accuracy and fine-grained detail preservation. This balanced approach ensures that our model learns to estimate depths accurately while retaining the intricate visual characteristics of the scene, ultimately improving the overall quality of the depth predictions.

A critical consideration in our loss function design is preserving high-frequency details, particularly the boundaries of objects in the scene. These details contribute to the perceptual quality and visual fidelity of the depth map, and keeping them is essential for generating accurate depth estimations. By penalizing distortions of these high-frequency details, we ensure that the network focuses on capturing fine-grained structural information and avoids the over-smoothing or blurring of the depth map. Additionally, a regularization term is incorporated into the loss function to encourage the smoothness of the depth map. This regularization term helps

to alleviate potential noise or inconsistencies in the predicted depth values, promoting a more coherent and visually pleasing depth estimation.

Considering these factors in our loss function design and network training, we construct a loss function comprising a weighted combination of three distinct loss components. The first loss component focuses on depth regression and quantifies the disparity between the predicted depth map $\hat{y}$ and the ground truth depth map y. Equation (1), which represents L_{depth}, is formulated as the point-wise $L1$ loss, quantifying the discrepancy between the predicted and ground truth depth values.[13] This loss term focuses on capturing the differences in depth measurements at each pixel location.

$$L_{depth}(y,\hat{y}) = \frac{1}{n}\sum_{p=1}^{n}\left|y_P - y_P\right| \tag{1}$$

Here, L_{depth} represents the pointwise $L1$ loss, quantifying pixel-level differences between the predicted and ground truth depth values. n denotes the total number of pixels, and p is an individual pixel in the depth map.

The second loss component emphasizes preserving structural details and boundaries within the depth map. A perceptual loss function is incorporated to penalize distortions in high-frequency information. Equation (2) represents L_{grad}, computed as the $L1$ loss applied to the image gradient of the depth image.[13] This loss term emphasizes preserving gradient information in the depth estimation.

$$L_{grad}(y,\hat{y}) = \frac{1}{n}\sum_{p=1}^{n}\left|g_x(y_P,y_P)\right| + \left|g_y(y_P,y_P)\right| \tag{2}$$

In Eq. (2), g_x and g_y represent the gradients of each pixel in the image, where x and y denote the horizontal and vertical spatial coordinates, respectively. g_x signifies the horizontal intensity change rate, while g_y represents each pixel vertical intensity change rate. These gradients are essential for capturing intensity variations, aiding tasks such as edge detection and feature extraction.

The structural similarity (SSIM) is incorporated into the loss function represented in Eq. (3). The SSIM is a well-established metric for image reconstruction tasks. It effectively evaluates depth map differences, including results of brightness, contrast, and structure comparisons, with values ranging from −1 to 1.[13] The third loss component introduces a smoothness regularization term, which promotes coherence and consistency in the predicted depth map. It measures the dissimilarity between the predicted and ground truth images based on their structure, with higher SSIM values indicating more accurate predictions and a closer match to the actual scene. The loss function combined with SSIM can enhance the model ability to generate depth maps resembling the true scene structure. As SSIM has a maximum value of one, we define L_{SSIM} as a loss term that measures the dissimilarity between the predicted depth image $\hat{y}$ and the ground truth depth image y.

$$L_{SSIM}\left(y,\hat{y}\right)=\frac{1-SSIM\left(y,\hat{y}\right)}{2}$$
$$SSIM(y, \hat{y}) = [l(y, \hat{y})]^{\alpha} * [c(y, \hat{y})]^{\beta} * [s(y, \hat{y})]^{\gamma} \quad (3)$$

Here, l, c, and s are the comparison measurements between the ground truth (y) and predicted ($\hat{y}$) depth map for luminance, contrast, and structure elements. α, β, and γ are the weight coefficients used to set each comparative item weightiness and all adjust to 1 to reduce this formula.

Appropriate weights are assigned to each loss component to achieve a balanced optimization process. By combining these three loss functions in a weighted manner, Eq. (4) is used in network training and tuning to estimate depth effectively while preserving critical visual details.

$$L(y, \hat{y}) = \lambda L_{depth}(y, \hat{y}) + L_{grad}(y, \hat{y}) + L_{SSIM}(y, \hat{y}) \quad (4)$$

In the depth estimation task, we choose to limit the weighting factor λ to a range of 0 to 1. To maintain the stability and performance of the model, this range is selected based on generally established standards to guarantee that the weights of each loss component are neither too large nor too small. Our comprehensive analysis and empirical findings led us to limit the depth loss L_{depth} weighting factor λ to a range of 0.1 to 0.2. The values of λ within this range are generally considered appropriate and frequent selections, resulting in faster convergence and better performance during training.

2.5 Augmentation strategies

Data augmentation is an essential technique in deep learning to combat model overfitting and enhance network generalization capabilities. In depth estimation, where our network aims to predict depth maps for complete images, careful consideration is necessary when choosing suitable geometric and photometric transformations. While image rotation is a commonly used augmentation strategy, we exclude it from our approach owing to the introduction of invalid depth data that does not correspond to the ground truth depth. Therefore, we limit our geometric augmentation to horizontal flipping with a probability of 0.5 to ensure a balanced distribution of the original and flipped images during training. This promotes the unbiased learning of object orientations and maintains semantic consistency in the scene.

Photometric augmentations such as brightness, contrast, and sharpness adjustments can further enrich the training dataset and enhance the network performance to handle variations in lighting conditions and image quality. Two image processing techniques, sharpening and brightening, were employed to improve the training process for our images. Sharpening highlights high-frequency details in the pictures by applying sharpening filters, which accentuate the differences between pixels and their surrounding pixels, thereby increasing the contrast and clarity of the images. On the other hand, brightening involves increasing the brightness of images to improve their quality, aiding in handling images under various lighting

conditions while enhancing their details and contrast. By applying these techniques, our model is better equipped to handle images with different contrasts and lighting conditions during training, improving its performance and generalization capability for real-world applications. Figure 5 shows examples of the image data augmentation used in our training stage.

3. Experimental Results and Discussion

The experimental results and analysis are demonstrated in this section, highlighting the exceptional performance of our proposed method compared with other methods. To further investigate and explore the effectiveness of our network model, we also conduct ablation studies that dissect the individual components and evaluate their effects, along with evaluation metric comparisons.

3.1 Evaluation metrics

We employ the standard set of six evaluation metrics widely utilized in previous studies to compare our proposed method and other approaches quantitatively.[1] These error metrics are well-established and provide a comprehensive evaluation framework. Equation (8) has three evaluation metrics by different thresholds.

1) Average relative error (REL) in Eq. (5):

$$\frac{1}{n}\sum_{p=1}^{n}\left|y_p - \hat{y}_p\right| / y_p. \tag{5}$$

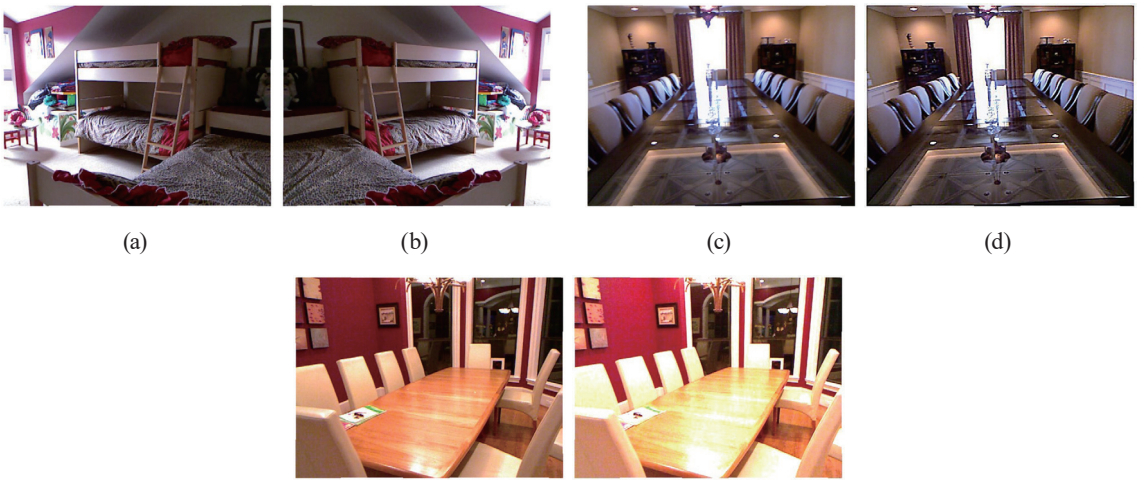

Fig. 5. (Color online) Image data augmentation. (a) Original and (b) horizontal flipped images, (c) original and (d) sharpened images, and (e) original and (f) brightened images.

2) Root mean squared error (RMS) in Eq. (6):

$$\sqrt{\frac{1}{n}\sum_{p=1}^{n}\left(y_p-\hat{y}_p\right)^2}. \tag{6}$$

3) Average (Log_{10}) error in Eq. (7):

$$\frac{1}{n}\sum_{p=1}^{n}\left|\log y-\log \hat{y}_p\right|. \tag{7}$$

4) Threshold accuracy (δ) in Eq. (8):

$$\begin{aligned}&\text{percentage (\%) of } y_p \text{ s.t. Max}\left(\frac{y_p}{\hat{y}_p},\frac{\hat{y}_p}{y_p}\right)=\delta_i<thr\\&i=1,2,3,\ thr \text{ is correspondingly set to } 1.25,1,25^2,1.25^3.\end{aligned} \tag{8}$$

When evaluating depth estimation results, the notation is as follows: y_p is the ground truth depth image y corresponding to pixel p, $\hat{y}_p$ is the predicted depth image $\hat{y}$ corresponding to pixel p, and n means the total number of pixels in each depth image. This notation facilitates a precise comparison and analysis of depth estimation accuracy across the entire image.

3.2 Experimental results

Figure 6 shows a collection of depth estimation outcomes achieved through our proposed methodology juxtaposed with the results obtained from other approaches. The initial column exhibits the RGB input images utilized as the network input. The subsequent column depicts the ground truth depth image. The third column showcases the outcomes achieved by Fu *et al.*,[6] and the fourth column shows the results produced by Alhashim and Wonka.[13] Finally, in the fifth column, the depth maps generated by our method are trained using the proposed approach.

By carefully examining the comparisons, the results of other methods show that their contours lack distinctiveness, and the depth coloration appears blurry and fragmented. The prediction outputs of our approach are better than those of others, particularly regarding the fully connected variant of ResNet. While this variant demonstrates enhanced accuracy, its predictions remain limited to coarse estimations. In contrast, our proposed fully convolutional model in the decoder block significantly enhances the quality of depth maps by improving edge precision and structure definition. These visualizations vividly demonstrate that our method produces depth estimations of superior quality, exhibiting better alignment with the ground truth depth edges and displaying considerably fewer artifacts.

The superior performance of our method can be attributed to incorporating advanced architectural elements and training techniques. The proposed CUB structure applies the advantages of bilinear interpolation to capture and interpolate spatial features efficiently. The

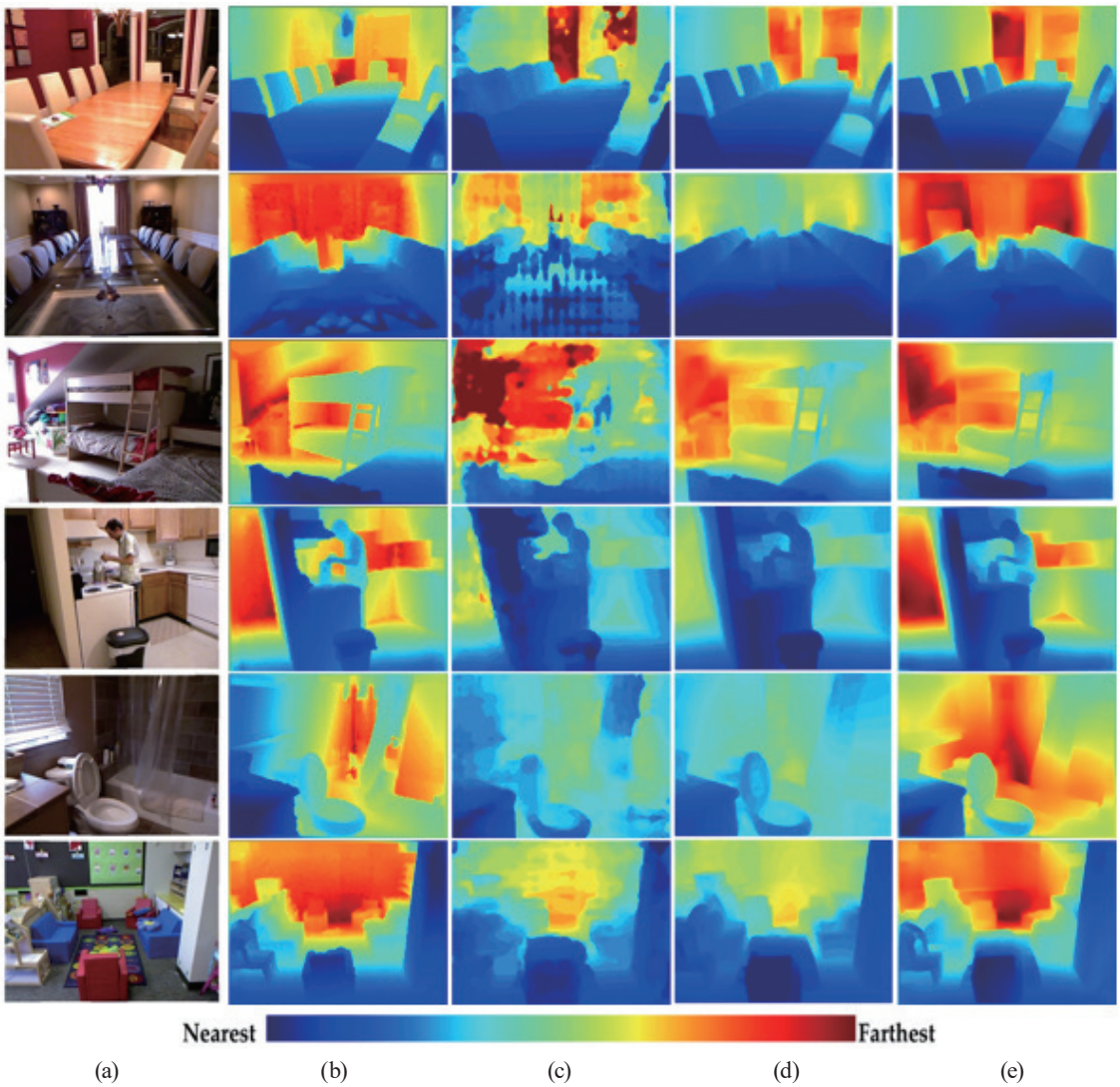

Fig. 6. Comparison of depth maps obtained by different methods on the NYU Depth v2 dataset. (a) RGB images, (b) ground truth, (c) results of model by Fu *et al.*,[6] (d) results of model by Alhashim and Wonka,[13] and (e) our predicted depth maps.

subsequent application of transpose convolution smooths the up-scaled features and aids in the recovery of details lost during the down-sampling process by intelligent learning. The fully convolutional model effectively captures fine-grained details and spatial dependences, leading to more accurate depth predictions. The improved edge quality and reduced presence of artifacts contribute to a more visually pleasing and precise depiction of the scene depth structure.

3.3 Results compared with other methods

Table 1 shows the performance comparisons between our proposed network model and other methods depending on the NYU Depth v2 dataset. Our model exhibits superior performance across most evaluation metrics; it showcases a remarkable advantage over the most compared

Table 1
Comparisons of different methods on the NYU Depth v2 dataset.

Method	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	REL↓	RMS↓	$Log_{10}\downarrow$
Li *et al.*[2]	0.621	0.886	0.968	0.232	0.821	0.094
Wang *et al.*[14]	0.605	0.890	0.970	0.220	0.745	0.094
Li *et al.*[15]	0.788	0.958	0.991	0.143	0.635	0.063
Iro *et al.*[5]	0.811	0.953	0.988	0.127	0.573	0.055
Xu *et al.*[16]	0.811	0.954	0.987	0.121	0.586	0.052
Fu *et al.*[6]	0.828	0.965	0.992	0.115	0.509	0.051
Hendra and Kanazawa[12]	0.819	0.960	0.989	0.143	0.509	0.060
Our method	**0.8266**	**0.9672**	**0.9926**	**0.1345**	**0.5800**	**0.0573**

approaches, with a reduced network parameter count of 67 million compared with other methods, which have more than 110 million. Moreover, our network demands fewer training iterations, 2 million instead of 3 million. It successfully achieves competitive results even with a smaller training dataset comprising 50000 samples, whereas other methods rely on 120000 samples. The outperformance is ascribed to the proposed CUB structure, which reduces training time and network parameters and overcomes the challenge of problematic convergence.

Achieving accurate absolute scale estimation in single image depth estimation networks is frequently formidable. We introduce a corrective strategy to address this challenge by rescaling the predicted depths using a scalar factor that aligns the median value with the ground truth. Lower values of REL, RMS, and Log_{10} error indicate improved performance, whereas higher values of δ_1, δ_2, and δ_3 imply higher accuracy in depth estimation. Our approach showcases efficiency, precision, and innovation in depth estimation, offering superior performance with fewer parameters and a more efficient training process.

3.4 Ablation studies and discussion

In our depth estimation model, the pretrained ResNet152V2 network is selected as the backbone network in the encoder of U-net. Moving into the decoding phase, the CUBs are considered a crucial decoder component. Each block includes 2× and 4× bilinear interpolation up-sampling and transpose convolution, accompanied by two 3 × 3 convolutional layers. The skip connection, which provides a direct way in conjunction with the corresponding encoder output, enhances the decoder representation capability.

Some ablation studies have been performed to find a better solution for depth estimation. The results of the individual ablation studies are presented in Table 2 for comparison. The bicubic interpolation method is used instead of the bilinear interpolation up-sampling; however, the results are almost the same as the performance of the original bilinear approach but with increased training time. We also try to apply pure transpose convolution or only the first two decoder blocks used for the up-sampling operation. The experimental results are average in performance and could be more satisfactory. Subsequently, the structure used one-stride transpose convolution after the bilinear up-sampling layer, followed by the original convolutional layers, to obtain further improvement in network performance. These adjustments and experiments from the ablation studies contribute to the enhanced performance and generalizability of our depth estimation network.

Table 2
Ablation studies.

Method	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	REL↓	RMS↓	$\text{Log}_{10}\downarrow$
Bilinear	0.8183	0.9642	0.9920	0.1365	0.5896	0.0584
Bicubic	0.8235	0.9645	0.9920	0.1363	0.5913	0.0576
Bilinear+Conv2DTranspose	**0.8266**	**0.9672**	**0.9926**	**0.1345**	**0.5800**	**0.0573**
Pure Conv2DTranspose	0.8210	0.9661	0.9917	0.1356	0.5928	0.0579
DecoderBlock1,2Conv2DTranspose	0.8195	0.9640	0.9918	0.1360	0.5932	0.0581

4. Conclusions

In this paper, we proposed an optimized U-net network that applies the ResNet152v2-based encoder with transfer learning mode and advanced decoder structures to estimate depth on a single 2D image. The optimized U-net combined with the proposed CUB in the advanced decoder and data augmentation techniques improves the training convergence, recovers the lost details, and smooths the up-scaled feature maps. Moreover, our system enables the accurate classification of different depth ranges, covering depths from 0.4 to 10 m. This classification is achieved by mapping various depths to corresponding colors using gradational color scales. The experimental results demonstrate superior performance that successfully transforms complex RGB images into precise depth maps.

Acknowledgments

This work was supported by the National Science and Technology Council under Grant no. NSTC 111-2222-E-167-003.

References

1 D. Eigen, C. Puhrsch, and R. Fergus: Proc. 2014 27th Int. Conf. Neural Information Processing Systems (NIPS, Cambridge, 2014) 2366–2374. https://doi.org/10.48550/arXiv.1406.2283
2 B. Li, C. Shen, Y. Dai, A. v. Hengel, and M. He: Proc. 2015 IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR, Boston, 2015) 1119–1127. https://doi.org/10.1109/CVPR.2015.7298715
3 F. Liu, C. Shen, and G. Lin: Proc. 2015 IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR, Boston, 2015) 5162–5170. https://doi.org/10.1109/CVPR.2015.7299152
4 F. Liu, C. Shen, G. Lin, and I. Reid: IEEE Trans. Pattern Anal. Mach. Intell. **38** (2016) 2024. https://doi.org/10.1109/TPAMI.2015.2505283
5 L. Iro, R. Christian, B. Vasileios, T. Federico, and N. Nassir: Proc. 2016 IEEE Int. Conf. 3D Vision (3DV, Stanford, 2016) 239–248. https://doi.org/10.1109/3DV.2016.32
6 H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao: Proc. 2018 IEEE Int. 2018 IEEE Computer Vision and Pattern Recognition (CVPR, Salt Lake City, 2018) 2002–2011. https://doi.org/10.1109/CVPR.2018.00214
7 W. Su, H. Zhang, J. Li, W. Yang, and Z. Wang: Proc. 2019 ACM 27th Int. Conf. Multimedia (ACM, Nice, 2019) 2161–2169. https://doi.org/10.1145/3343031.3350930
8 H. Jung, Y. Kim, D. Min, C. Oh, and K. Sohn: Proc. 2017 IEEE Int. Conf. Image Processing (ICIP, Beijing, 2017) 1717–1721. https://doi.org/10.1109/ICIP.2017.8296575
9 C. Zheng, T.-J. Cham, and J. Cai: Proc. 2018 Eur. Conf. Comput. Vis. (ECCV, Munich, 2018) 767–783. https://doi.org/10.48550/arXiv.1808.01454
10 A. Bhatia: Proc. 2020 IEEE Int. SoutheastCon (IEEE, Raleigh, 2020) 1–6. https://doi.org/10.1109/SoutheastCon44009.2020.9249700

11 D. H. Kwak and S. H. Lee: Sensors **20** (2020) 2567. https://doi.org/10.3390/s20092567
12 A. Hendra and Y. Kanazawa: Proc. IEEE Access (IEEE, 2023) 44176–44191. https://doi.org/10.1109/ACCESS.2023.3272292
13 I. Alhashim and P. Wonka: Proc. 2018 Computer Vision and Pattern Recognition (cs.CV, 2018) 1–12. https://doi.org/10.48550/arXiv.1812.11941
14 P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille: Proc. 2015 IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR, Boston, 2015) 2800–2809. https://doi.org/10.1109/CVPR.2015.7298897
15 J. Li, R. Klein, and A. Yao: Proc. 2017 IEEE Int. Conf. Computer Vision (ICCV, Venice, 2017) 3392–3400. https://doi.org/10.1109/ICCV.2017.365
16 D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe: Proc. 2017 IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR, Honolulu, 2017) 161–169. https://doi.org/10.1109/CVPR.2017.25