IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS. 2017.Doi Number

Joint Attention Mechanisms for Monocular Depth Estimation with Multi-Scale Convolutions and Adaptive Weight Adjustment

PENG LIU[1,2,3], ZONGHUA ZHANG[1,2], ZHAOZONG MENG[2], (Member, IEEE), and NAN GAO[2]

[1]State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300130, China
[2]School of Mechanical Engineering, Hebei University of Technology, Tianjin 300130, China
[3]School of Intelligence and Information Engineering, Tangshan University, Tangshan 063000, China

Corresponding author: Zonghua Zhang (e-mail: zhzhang@ hebut.edu.cn)

This work was supported in part by National Key R&D Program of China under Grant No. 2017YFF0106404 and the National Natural Science Foundation of China under Grant No. 51675160, 52075147.

ABSTRACT Monocular depth estimation is a fundamental problem for various vision applications, and is therefore gaining increasing attention in the field of computer vision. Though a great improvement has been made thanks to the rapid progress of deep convolutional neural networks, depth estimation of the object at finer details remains an unsatisfactory issue, especially in complex scenes that has rich structure information. In this paper, we proposed a deep end-to-end learning framework with the combination of multi-scale convolutions and joint attention mechanisms to tackle this challenge. Specifically, we firstly elaborately designed a lightweight up-convolution to generate multi-scale feature maps. Then we introduced an attention-based residual block to aggregate different feature maps in joint channel and spatial dimension, which could enhance the discriminant ability of feature fusion at finer details. Furthermore, we explored an effective adaptive weight adjustment strategy for the loss function to further improve the performance, which adjusts the weight of each loss term during training without additional hyper-parameters. The proposed framework was evaluated using challenging NYU Depth v2 and KITTI datasets. Experimental results demonstrated that the proposed approach is superior to most of the state-of-the-art methods.

INDEX TERMS Monocular depth estimation, multi-scale convolutions, joint attention mechanisms, weight adjustment.

I. INTRODUCTION

Monocular depth estimation is a fundamental but challenging task in computer vision, the goal of which is to predict a dense depth map from a given image. The technical progress in this area can be applied to widespread applications, such as scene understanding [1], [2], action recognition [3], 3D reconstruction [4], robotics [5], [6], etc. However, it is still a very challenging topic since one image may correspond to several real scenes and there are no other available clues, e.g. stereo correspondences, or motions.

Traditional solutions for this problem [7]-[11] mainly include two steps: hand-crafted feature extraction and structural prediction. However, these methods have no generality to suit the needs of various kinds of real-world scenes, which has limited the performance of such predicted models.

In recent years, deep learning methods make a huge breakthrough on many computer vision's studies, including image classification [12]-[14], scene parsing [15]-[17] and pose estimation [18], etc. Efforts based on deep convolutional neural networks (DCNNs) have also been successfully introduced to monocular depth estimation tasks [19], [28], [29]. The use of deep features is superior to hand-crafted features, and therefore significantly improves the estimation performance.

Depth estimation networks often consist of encoder which decreases spatial resolution while learning the feature representation, followed by decoder who gradually recovers the original depth map resolution to realize the end to end learning. But the pooling operation in the classification networks greatly reduces the spatial resolution of feature maps which is undesirable for depth estimation. Many

methods have been adapted to obtain high quality depth map, including skip connection [21], [22], multi-scale networks [23], [26], [27] or concatenated hierarchical features [24], [25] to sort out feature map with higher resolution. Though great achievements have been made, there are three reasons hindering the better improvement, which are the motivations for our new approach.

Firstly, in order to complete a fine-grained prediction, the fine-grained multi-scale representation is very important. Despite the gain in performance, common multi-scale convolutions often stack the convolutional layers with big kernel size in parallel. This approach increases the memory requirement and the model size substantially.

Secondly, different RGB image regions, such as the smooth regions and the regions with dense detailed textures, have different context information, which results in a different contribution for the final depth prediction. Therefore, due to the regional incoherence, fusing feature maps with a global image feature vector that considers each image region equally may result in a sub-optimal precision.

Thirdly, for predicting dense depth map with better quality, several loss terms are often combined to construct the total training loss [24]-[28], [37]-[40]. Weights of the loss terms are empirically determined and do not change during training. However, the ratios between different loss terms vary in a large range during training. Therefore, setting value of each weight to be invariable will obviously leads the total loss for gradient descent algorithm to be sub-optimal. Compared with the module architecture design and loss term selection in the literature, there are few discussions about these weights setting methods that one task with multiple loss terms.

Based on the above considerations, we build a deep convolutional network to settle these issues. Specifically, we realize a lightweight up-convolution that uses the dense connected way in sequential multi-scale convolutions. We name it dense multi-scale up-sample block (DMUB). The DMUB enriches the multi-scale representation during up-sampling with little parameter. To address the second problem above, we design a residual block with joint attention mechanisms to fuse the feature maps of the encoder and decoder part. We name it attention-based residual fusion block (ARFB). The ARFB aggregates the feature maps with fully consideration of the characteristics of the scenes. It enhances the fine-grained information processing ability of the model. We also propose an adaptive weight adjustment strategy in our loss function. Our strategy adjusts the weight of each loss term during training without additional hyper-parameters. It further improves various metrics on the benchmark datasets. Our main contributions can be summarized as follows:

- We propose a deep end-to-end learning framework with the combination of multi-scale convolutions and joint attention mechanisms. It includes a lightweight up-convolution to generate multi-scale feature maps, and an attention-based residual fusion block to enhance the discriminant ability of feature fusion at finer details.
- An adaptive weight adjustment strategy for combination of multiple loss terms is adapted to optimize the network training, which can further improve the predicted accuracy.
- Our proposed network is trained in an end-to-end fashion and achieves the state-of-the-art depth estimation performance on two public benchmark datasets (NYU Depth v2 and KITTI).

The rest of this paper is organized as follows: Sect. II. introduces the review of related work. Sect. III. presents our proposed depth estimation method. Sect. IV. gives the experimental results and analyses. Finally, Sect. V. concludes the work.

II. RELATED WORK

In this section, we concentrate on how to better use supervised DCNNs to settle the problem of monocular depth estimation and our method mainly concerns with the attention-based methods and the loss function design. Therefore, we will briefly classify the related works into three aspects: the supervised DCNNs-based methods for depth estimation, attention-based methods, and loss function design.

A. SUPERVISED DCNNS-BASED METHODS

The supervised DCNNs-based methods have effectively improved the performance of depth estimation by their powerful feature extraction ability. Eigen et al. [19] first proposed a multi-scale deep network for dense depth estimation. Laina et al. [28] employed the ResNet [41] structure with a well-designed up-sampling operator and achieved a better performance. Cao et al. [29] considered the depth estimation problem as a pixel-wise classification task by training a fully convolutional deep residual network based on the long tail property of depth data distribution.

The methods above usually applied a deep neural network designed from image classification in a full convolutional way as the feature extractors. The repeated pooling operations in these feature extractors inevitably decrease the spatial resolution of feature maps and have a bad effect on the local detail depth prediction. In order to solve this problem, Godard et al. [21] and Xie et al. [22] adopted the skip-connection strategy to fuse low-spatial resolution depth maps in deeper layers with high-spatial resolution depth maps in lower layers. Zheng et al. [24] and Hu et al. [25] applied concatenated hierarchical features to finish the coarse-to-fine process. They all integrated hierarchical depth features by combining various-level information of depth feature maps with up-convolution to further realize the fine-grained depth prediction. Some other works [23], [26], [27] aggregated multi-scale contexts to improve the prediction performance. For instance, Zhao et al. [23] exploited image super-resolution techniques to finish the multi-scale feature

fusion that can get an accurate depth map. Chen et al. [27] introduced an adaptive dense feature fusion module to adaptively fuse effective features from multi-scale for inferring structures of the scene.

B. ATTENTION-BASED METHODS

Attention mechanisms have been proved efficient in modeling long-range dependencies and have been widely applied in depth prediction tasks. For instance, Jiao et al. [32] proposed an attention-driven learning approach for monocular depth estimation, which also predicted corresponding semantic labels. Chen et al. [33] proposed an attention-based context aggregation network to aggregate both the image-level and pixel-level context information for depth estimation. Kong et al. [36] presented a pixel-wise attentional gating unit to learn the spatial allocation of computation in dense labeling tasks. Li et al. [34] used channel-wise attention mechanisms to extract discriminative features for depth prediction. Very recently, Wang et al. [54] applied a channel-spatial attention module in their encoder-decoder framework to improve the representation ability of feature maps.

Recently, some researchers have begun to develop lightweight joint attention modules that inferred attention maps along channel and spatial dimensions simultaneously. For instance, Woo et al. [45] applied an attention-based feature refinement with channel and spatial modules that achieved considerable performance improvement on image classification tasks while keeping the overhead small. Park et al. [30] proposed a simple and effective joint attention module at each bottleneck of models. Roy et al. [31] re-calibrated the channel of feature maps by joint attention mechanisms, and proved its effectiveness for medical image segmentation.

C. LOSS FUNCTION

Loss function provides data for the gradient optimization algorithm to update the parameters of the DCNNs. It plays an important role in the learning process. As reported in the literature, different individual loss term or the combining of multiple loss terms were used to construct the loss function. Eigen et al. [28] proposed a scale invariant loss to optimize the model learning. Fabio et al. [40] defined their total loss as a sum of three main contributions including disparity smoothness loss, an image reconstruction loss, and a proxy-supervised loss. Zhang et al. [24] considered the training loss as combination of the point-to-point, shape and distribution similarity between predictions and ground truth, which leveraged hierarchical structure information to guide the network optimization. Alhashim et al. [37] and Gur et al. [38] regarded the training loss as the sum of L1 loss and structural similarity (SSIM) loss, which sought the balance between the point-to-point difference and the distortions of high frequency details in the image domain. Hu et al. [25] and Chen et al. [27] made a simple analysis of orthogonal sensitivities to different types of errors and then proposed to use a combination of three loss function terms, which included the point-to-point loss, the gradients loss, and the normal loss.

Recently, some researchers have used an adaptive way to set the weight of each loss term. For instance, Jiang et al. [39] introduced an adaptive optimized weight allocation algorithm based on a Gaussian model to maximize the effectiveness of weighting for their proposed hybrid loss function. Zhang et al. [24] used a simple sum rule to finish the adaptive adjustment strategy.

While great improvements have been made, there is still room to gain a higher quality depth map. Our proposed method settles this problem to some extent from following three aspects. Firstly, we further improve the performance of multi-scale convolutions. Secondly, we enhance the discriminant ability of feature maps with attention-based method. Thirdly, we optimize the weight allocation of the loss function.

III. PROPOSED METHOD

In this section, we firstly present the architecture of the proposed DCNNs, followed by the introductions of the DMUB and ARFB. Finally, we state our loss function and corresponding adaptive weight adjustment strategy.

A. NETWORK ARCHITECTURE

Fig. 1 is a pictorial description of our proposed network. It is built upon a convolutional auto-encoder architecture. The network comprises an encoder and a decoder pathway, with skip connections between the corresponding layers.

The encoder extracts dense features from the input RGB images so we also name it feature extraction network. Following the previous depth estimation approaches, we choose standard DCNNs originally designed for image classification task as our feature extraction network. The repeated combination of max-pooling and striding in standard DCNNs extract feature maps at different resolution, as shown in Fig. 1.

In our proposed decoder, we first attach a 1×1 convolutional layer to reduce the dimension of the highest-level feature maps. The DMUB is sequentially applied to up-sample the feature maps in a multi-scale manner. Then, output of the DMUB and the feature maps with the same resolution in feature extraction network are aggregated by our ARFB. The ARFB further aggregates the feature maps to realize better feature representation by explicitly modeling inter-dependencies of the feature maps in channel and spatial dimensions. Alternate use of the DMUB and ARFB gradually recovers the feature maps back to the resolution of expected depth map. Output of the final DMUB is fed to a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030097, IEEE Access

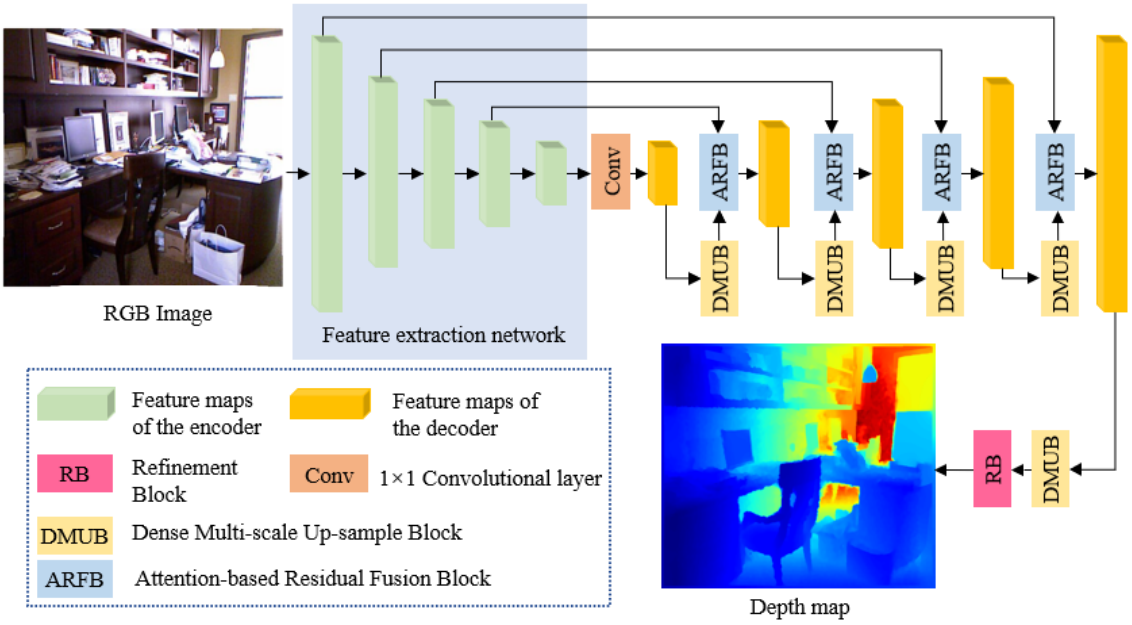


FIGURE 1. **The architecture of our proposed network.**

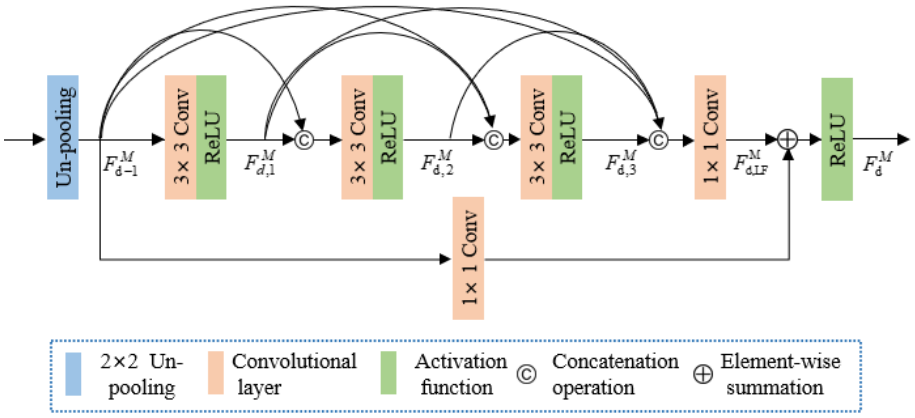


FIGURE 2. **The architecture of Dense Multi-scale Up-sample Block.**

refinement block (RB) to give the final fine-grained depth prediction. The RB has three 5×5 convolutional layers, the efficiency of which has been proved by [25].

B. DENSE MULTI-SCALE UP-SAMPLE BLOCK

The DMUB is designed to enhance the multi-scale representation ability and decrease model size compared with the common multi-scale convolutions. Inspired by the success of DenseNet [42] and MultiRes [43] in image segmentation task, we add the dense connected way in sequential 3×3 multi-scale convolutions, as shown in Fig. 2. It enhances the multi-scale representation ability by fully making use of all hierarchical features of the convolutions.

To be specific, our DMUB contains a 2×2 un-pooling layer to up-sample the feature maps, dense connected layers to generate hierarchical features, a local feature fusion operation to adaptively fuse the hierarchical feature, and a 1×1 convolutional shortcut to comprehend some additional spatial information.

In the d-th up-sampling, we firstly use 2×2 un-pooling to get the rough up-sample feature maps F_{d-1}^{M}. The output of c-th convolutional layer in our DMUB $F_{d,c}^{M}$ can be formulated as,

$$F_{d,c}^{M} = \sigma(W_{d,c}^{M}([F_{d-1}^{M}, F_{d,1}^{M}, F_{d,2}^{M}, \cdots, F_{d,c-1}^{M}])) \qquad (1)$$

where σ is the ReLU activation function, $[F_{d-1}^{M}, F_{d,1}^{M}, F_{d,2}^{M}, \cdots, F_{d,c-1}^{M}]$ means the concatenation of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030097, IEEE Access

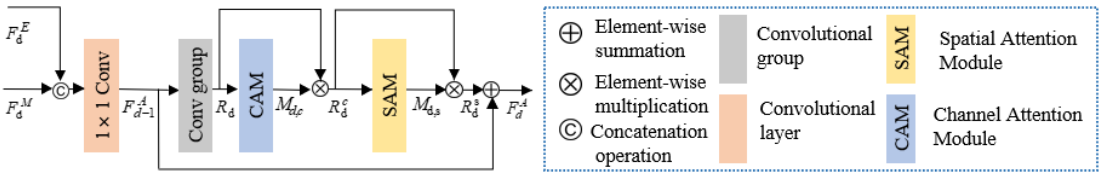


FIGURE 3. The architecture of Attention-based Residual Fusion Block. Details of CAM and SAM are shown in Fig. 4.

feature maps generated by the previous convolutional layer, and $W_{d,c}^M$ denotes the weights of the c-th convolutional layer.

The local feature fusion operation is formulated as,

$$F_{d,LF}^M = W_{d,LF}^M([F_{d-1}^M, F_{d,1}^M, F_{d,2}^M, \cdots, F_{d,c}^M]) \quad (2)$$

where $W_{d,LF}^M$ denotes the function of the 1×1 convolutional layer, which finishes the local feature fusion in the d-th up-sampling.

We introduce a 1×1 convolutional shortcut to further improve the feature maps. The final output of the d-th up-sampling can be obtained by,

$$F_d^M = \sigma(F_{d,LF}^M + W_{d,sc}^M(F_{d-1}^M)) \quad (3)$$

where $W_{d,sc}^M$ denotes the function of the 1×1 convolutional shortcut in the d-th up-sampling.

C. ATTENTION-BASED RESIDUAL FUSION BLOCK

Architecture of the ARFB is shown in Fig. 3. As shown, the operation of concatenation and 1×1 convolutional first play a local feature reduction and fusion role. After that is a residual module, which is inspired by [44]. It releases gradient vanishing of the network and further enhances the discriminative ability of the block by the channel attention module (CAM) and spatial attention module (SAM) integrated in it. The CAM and SAM are connected in a sequential manner, which is the same as [45]. The CAM focuses on the inter-channel relationship of feature maps with the same resolution. The SAM exploits informative region and highlights useful local spatial features for accurate pixel-level prediction. In this way, the ARFB could generate the feature maps with full consideration of their structure-related regions and inter-channel relationship.

In the d-th feature fusion, we firstly merge the output of the DMUB F_d^M and corresponding feature maps F_d^E produced by the feature extraction network with a local feature fusion operation, to generate the input $F_{d-1}^A \in R^{C\times H\times W}$ for the subsequent residual attention maps. It can be summarized as,

$$F_{d-1}^A = W_{d,LF}^A([F_d^M, F_d^E]) \quad (4)$$

where $W_{d,LF}^A$ denotes the function of the 1×1 convolutional layer, which finishes the local feature fusion in the d-th feature fusion.

Then, the ARFB learns the residual sequentially to infer a 1D channel attention map $M_{d,c}$ and a 2D spatial attention map $M_{d,s} \in R^{1\times H\times W}$ from the residual component obtained by a convolutional group (we stacked two convolutional layers here) output R_d. And finally, the output feature maps $F_d^A \in R^{C\times H\times W}$ can be calculated from this residual. The overall calculation process can be summarized as,

$$R_d = BN(W_{d,c2}^A(\sigma(BN(W_{d,c-1}^A(F_{d-1}^A))))) \quad (5)$$

$$R_d^c = M_{d,c}(R_d) \otimes R_d \quad (6)$$

$$R_d^s = M_{d,s}(R_d^c) \otimes R_d^c \quad (7)$$

$$F_d^A = F_{d-1}^A + R_d^s \quad (8)$$

where $\otimes$ means element-wise multiplication. $W_{d,c1}^A$ and $W_{d,c2}^A$ are two different standard 3×3 convolution operations. *BN* means batch-normalization operation, which is not contained in [44] but experimentally proved efficiency in our network.

Our channel and spatial attention module both follow the classic structure proposed by [45]. As shown in Fig. 4, the CAM calculates channel-wise weights that contains squeezing and excitation step, the channel attention calculation is formed as,

$$M_{d,c}(R_d) = F_s(FC_{d,2}(\sigma(FC_{d,1}(F_{av}(R_d) + F_{\max}(R_d))))) \quad (9)$$

where F_{av} and $F_{\max}$ are average-pooling and max-pooling operations respectively. F_s is a sigmoid function. $FC_{d,1}$ and $FC_{d,2}$ are fully connected operations.

The spatial attention is computed as,

$$M_{d,s}(R_d^c) = F_s(W_d^A([F_{av}(R_d^c), F_{\max}(R_d^c)])) \quad (10)$$

where W_d^A is a standard 7×7 convolution operation.

Wang et al. [54] also used the joint attention mechanisms to improve the presentation ability of the feature maps. Except for different module architecture, we put the joint attention mechanisms into whole hierarchical guidance progress with fully consideration of the rich information in each feature maps resolution, while Wang et al. [54] applies the joint attention mechanisms to improve presentation ability of the highest-level feature maps.

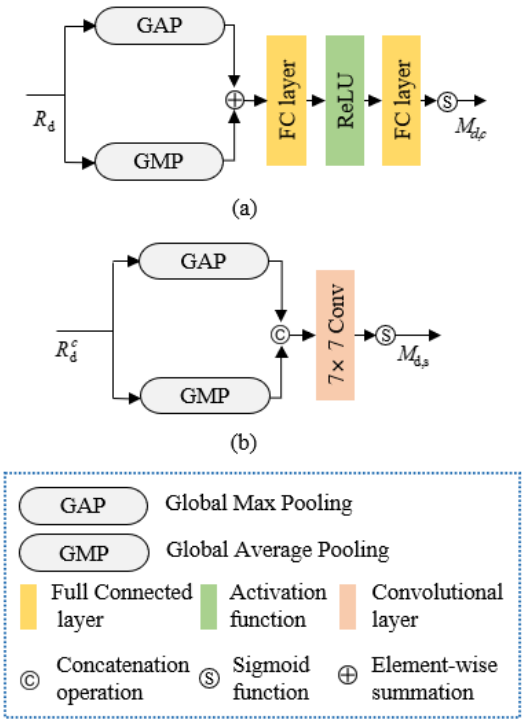


FIGURE 4. Details of our channel and spatial attention module. (a) channel attention module (CAM). (b) spatial attention module (SAM).

D. LOSS FUNCTION WITH ADAPTIVE WEIGHT ADJUSTMENT STRATEGY

We make a combination of different loss terms that also selected in the state-of-the-art methods [25], [28]. It includes a point-to-point loss, a gradients loss, and a normal loss. This combination balances the reconstructing depth by minimizing the difference between the ground-truth while also penalizing the loss of high frequency details typically correspond to the boundaries of objects. It is formulated as,

$$L_{total} = \lambda_1 L_{depth} + \lambda_2 L_{grad} + \lambda_3 L_{normal}. \quad (11)$$

We introduce each loss term as follows.

1) Point-to-point Loss: The logarithm of depth errors is selected to obtain the pixel-level difference. It has a discriminative contribution to the loss between different distance in a scene and is formulated as,

$$L_{depth} = \frac{1}{n}\sum_{i=1}^{n} \ln(e_i + \alpha_1) \quad (12)$$

where e is the L1 Euclidean distance between a predicted depth map and the corresponding ground truth, i is pixel index, n is the total number of pixels in each map, and α_1 is a parameter that always set to be 0.5.

2) Gradients Loss: The depth of complicated scene is often discrete and change acutely on the object boundaries, which shows abundant of local features. In order to better preserve these details, we use the gradient loss layers with kernels set as the Sobel detector on both horizontal and vertical directions to penalize such errors. It can be formulated as,

$$L_{grad} = \frac{1}{n}\sum_{i=1}^{n} \ln\left(\left|\nabla_x^{sobel} e_i\right| + \left|\nabla_y^{sobel} e_i\right| + \alpha_2\right) \quad (13)$$

where ∇_x^{sobel} and ∇_y^{sobel} represent the horizontal and vertical convolutional Sobel operator, which calculate the gradient information and are sensitive to the shift of edges in x and y directions. α_2 is a parameter that always set to be 0.5.

3) Normal Loss: we consider accuracy of the normal to the surface of the predicted depth map to handle small depth structures. It can be formulated as,

$$L_{normal} = \frac{1}{n}\sum_{i=1}^{n}\left|1 - \frac{\left\langle n_i^d, n_i^g \right\rangle}{\sqrt{\left\langle n_i^d, n_i^d \right\rangle}\sqrt{\left\langle n_i^g, n_i^g \right\rangle}}\right| \quad (14)$$

where n_i^d and n_i^d are the surface normal of the predicted depth map and its ground truth, which are computed as, $n_i^g = [-\nabla_x(g_i), -\nabla_y(g_i), 1]$ and $n_i^d = [-\nabla_x(d_i), -\nabla_y(d_i), 1]$.

From equation (12), (13) and (14), we find that the logarithmic operation causes values of the point-to-point loss and gradients loss to be negative all the time after training several epochs. Therefore, if each weight of the loss term is set to be invariable, total training loss will always be dominated by the normal loss that is positive, which flies in the face of the principled approach about weight distribution in multi-task learning task [51], that total loss should balance the contribution of each loss term and not be dominated by one loss term. Therefore, we propose the weight adjustment strategy to fix this problem. Specifically, we firstly use negative log-softmax function to shrink the value of each loss term. The softmax operator transfers the value setting as a probability distribution issue that significantly reduces the imbalance effect and the negative logarithm operation ensures the smaller loss terms get bigger weights as we expect. It is formed as,

$$\omega_i(t) = -\ln\frac{e^{L_i(t)}}{\sum_{j=1}^{n} e^{L_j(t)}} \quad (15)$$

where t is the iteration step, $L_i(t)$ is the loss term we select above respectively, and n is the number of different loss terms. Obviously, it provides adaptive influence during optimizing process to balance the three different loss terms as we expected.

DCNNs use the total loss to complete gradient optimization during training process. So, we enlarge the output above linearly to ensure sum of the weights is n, which is the same as the common invariable setting. By this linear enlarge operation, we adaptively enlarge the weights to ensure that total loss has approximately equivalent influence for the optimization algorithm compared with common invariable setting. This linear enlarge operation is formed as,

$$\lambda_i(t) = \frac{n \cdot \omega_i(t)}{\sum_{j=1}^{n} \omega_j(t)} \tag{16}$$

Finally, from equation (15) and equation (16), we deduce the final weight adjustment strategy. It is formed as,

$$\lambda_i(t) = \frac{n \cdot (L_i - \ln \sum_{j=1}^{n} e^{L_j})}{\sum_{j=1}^{n} L_j - n \cdot \ln \sum_{j=1}^{n} e^{L_j}} \tag{17}$$

Relative works have been done in [24] and [39]. [39] also considers this weight determination as probability distribution issue, but it needs additional two hyper-parameters, which are hard to determine. The simple sum rule used in [24] is unsuited for the training loss with logarithmic combination because the negative value for logarithm operation may lead the denominator of their formula to be zero.

IV. EXPERIMENTS

To evaluate our proposed method, we have carried out comprehensive experiments on two public datasets: the NYU Depth v2 [46] and the KITTI [47]. In the following subsections, we firstly introduce the experimental setup and the implementation details, and then show the experimental results. We also analyze the effectiveness of our proposed blocks and the weight adjustment strategy in ablation study.

A. EXPERIMENTAL SETUP

1) DATASETS

The NYU Depth v2 dataset consists of 464 scenes from indoor scenes, captured by Microsoft Kinect devices. Followed by the official split, the training dataset includes 249 scenes with the 795 pair-wise images, and the testing dataset consists of 215 scenes with 654 pair-wise images. The spatial resolution of the RGB and depth images is 480×640. We firstly cropped each image to the size of 228×314 and offline data augmentation the same as done in mainstream approach [24], [25], [27]. We performed data augmentation on the training samples by random rotated [-5, 5] degrees, random scaled depths by s $\in$ [1, 1.5] times, color shift (multiplied by a random value $\in$ [0.8, 1.2]), horizontal flip, translation, and contrast (multiply with value $\in$ [0.5, 2.0]) shift.

KITTI dataset [9] is made up of several outdoor scenes captured by LIDAR sensor and car-mounted cameras at resolution of roughly 375×1242. The train and test splits we employ followed the method in [6]. We used about 22k images for training, and 697 test images from different scenes. Both the input images and corresponding depth-maps were resized to 160×512 to form the inputs. Because the depth maps projected by the LIDAR point cloud are sparse, we masked them out and evaluated the loss only on valid points with ground depth in testing process. We capped the maximum predictions of all networks to 80 meters, which is the maximum depth of KITTI dataset. We performed data augmentation on the training samples by following the method in [33].

2) EVALUATION METRICS

We quantitatively compared our method with the state-of-the-art methods using the following metrics that are commonly employed in previous studies.

Abs Relative Difference (Abs Rel): $\frac{1}{n}\sum_i \frac{|y_i - y_i^*|}{y_i^*}$.

Squared Relative Difference (Sq Rel): $\frac{1}{n}\sum_i \frac{\|y_i - y_i^*\|^2}{y_i^*}$.

Mean log10 Error (log10): $\frac{1}{n}\sum_i |\log_{10} y_i - \log_{10} y_i^*|$.

Root Mean Squared Error (RMS): $\sqrt{\frac{1}{n}\sum_i (y_i - y_i^*)^2}$.

Log10 root mean squared error (logRMS): $\sqrt{\frac{1}{n}\sum_i (\log_{10} y_i - \log_{10} y_i^*)^2}$.

Threshold accuracy δ% of y_i *s.t.* $\max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$ for thr =1.25,1.25^2,1.25^3.

where y_i^* is the ground-truth depth, y_i is the estimated depth, and n is the total pixels in all evaluated images. Values of Abs REL, Sq Rel, log10, logRMS and RMS are smaller better and values of percentage (%) are bigger better.

B. IMPLEMENTATION DETAILS

We implemented our proposed network using PyTorch framework, running on two Nvidia RTX 2080ti GPUs with11GB memory each. We adapted resnet-101, densenet-161, and SENet-154 as our feature extraction network, which were pretrained on ImageNet dataset [49]. The parameters for the other parts were initialized randomly following [27]. We selected the Adam [48] optimizer in all experiments with the base learning rate 0.0001 and reduced it to 10% for every 5 epochs. We set β1=0.9, β2=0.999, and used weight decay of 0.0001. The batch size was set to 16. We trained the network for 30 epochs. Our final DCNNs used the SENet-154 as feature extraction network, and its decoder part has approximately 177.2M parameters. In our experiments, training was implemented with 3168 iterations for NYU Depth v2, needing about 24 hours to finish. For KITTI dataset, our module was implemented with 2895 iterations, which needed about 18 hours to finish.

C. EXPERIMENTAL RESULTS

1) QUALITATIVE AND QUANTITATIVE COMPARISONS

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030097, IEEE Access

TABLE 1. Performance evaluation of state-of-the-art methods on NYU Depth v2. The best scores are highlighted in bold font.

Method	Error (lower is better)			Accuracy (higher is better)		
	Abs Rel	RMS	log10	$\delta_1(\delta<1.25)$	$\delta_2(\delta<1.25^2)$	$\delta_3(\delta<1.25^3)$
Eigen et al. [19]	0.212	0.873	-	0.611	0.887	0.969
Laina et al. [28]	0.127	0.573	0.055	0.811	0.953	0.988
Alhashim et al. [37]	0.123	0.465	0.053	0.846	0.974	0.994
Jiao et al. [32]	0.126	0.416	0.050	0.868	0.973	0.993
Xu et al. [26]	0.121	0.586	0.052	0.811	0.954	0.987
Hu et al. [25]	0.115	0.530	0.050	0.866	0.975	0.993
Li et al. [35]	0.134	0.540	0.056	0.832	0.965	0.989
Chen et al. [33]	0.138	0.496	-	0.826	0.964	0.990
Zhao et al. [23]	0.128	0.523	0.059	0.813	0.964	0.992
Liu et al. [53]	0.127	0.506	-	0.836	0.966	0.991
Fu et al. [50]	0.115	0.509	0.051	0.828	0.965	0.992
Wang et al. [54]	0.115	0.519	0.049	0.871	0.975	0.993
Chen et al. [27]	**0.111**	0.514	0.048	0.878	**0.977**	**0.994**
Lee et al. [20]	0.112	**0.352**	**0.047**	**0.882**	0.963	0.992
Ours (ResNet-101)	0.128	0.549	0.054	0.850	0.969	0.992
Ours (DenseNet-161)	0.125	0.546	0.053	0.850	0.971	0.993
Ours (SENet-154)	0.113	0.523	0.049	0.872	0.975	0.993

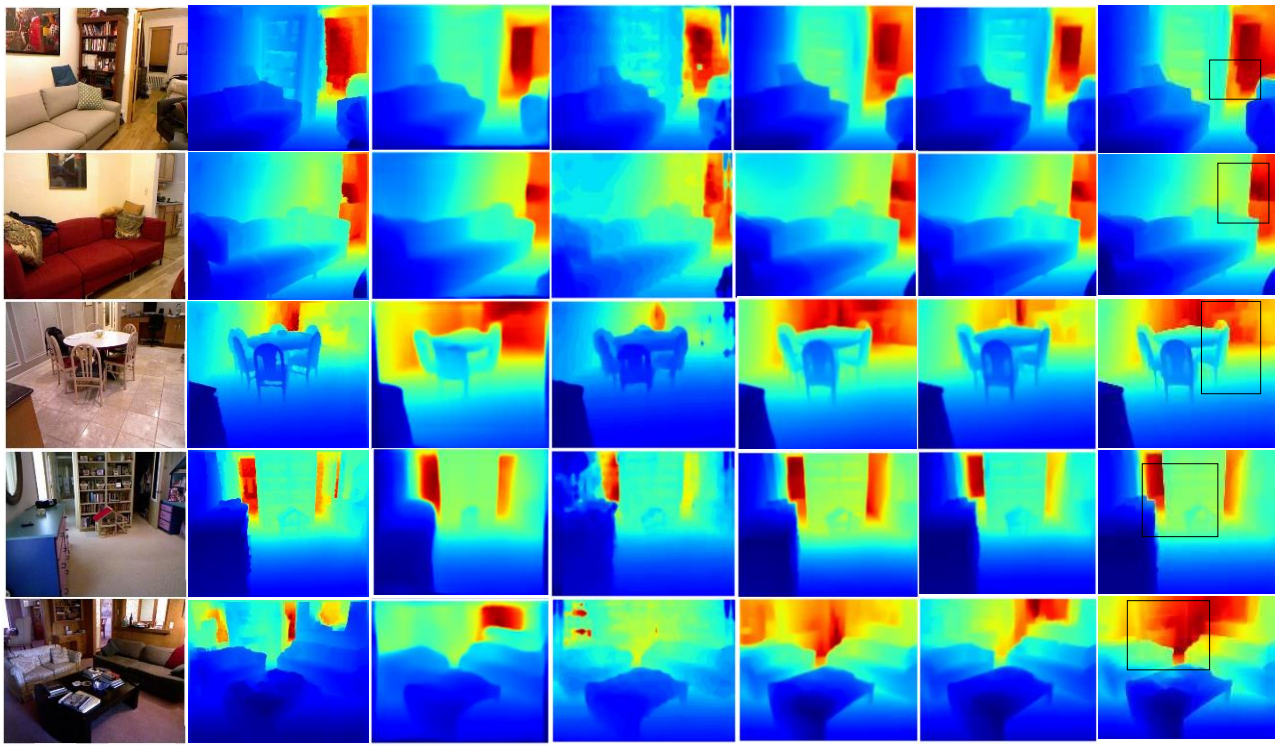


FIGURE 5. Qualitative evaluations on NYU Depth v2. Compared with the state-of-the-art methods. Color indicates depth (red is far, blue is close). The columns from left to right are RGB images, ground truth depth maps, results of Laina et al. [28], Fu et al. [50], Hu et al. [25], Chen et al. [27], and the proposed method.

We compared our method with the state-of-the-art and results of other algorithms were given in the literature. To verify the generality of our method in decoder part, we used ResNet-101, DenseNet-161, and SENet-154 as the feature extraction network. The comparative results of evaluation metrics on NYU Depth v2 dataset were reported in Table 1.

As shown in Table 1, it is a sufficient evidence that together with different feature extraction networks our method can get good results all the time, and the feature extraction network with SENet-154 architecture could get the best performance. Our method is competitive with most state-of-the-art methods.

Some qualitative comparisons were also presented in Fig. 5 and more qualitative results were given in Fig. 13(a). All predicted depth maps were shown in the same pixel with the ground truth for better comparison. The results demonstrate that our method presents better object boundaries and geometric details than other state-of-the-art methods, for instance, the nearby sofa back cushion in the first two rows, and the distant table in the second and last rows. The depth

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030097, IEEE Access

TABLE 2. Performance evaluations of state-of-the-art methods on KITTI. Depths are capped at 80 meters. The best scores are highlighted in bold font.

Method	Error (lower is better)				Accuracy (higher is better)		
	Abs Rel	RMS	Sq Rel	logRMS	$\delta_1(\delta<1.25)$	$\delta_2(\delta<1.25^2)$	$\delta_3(\delta<1.25^3)$
Eigen et al. [19]	0.190	7.156	1.515	0.270	0.692	0.899	0.967
Garg et al. [52]	0.152	5.894	1.226	0.264	0.784	0.921	0.967
Godard et al. [21]	0.148	5.927	1.515	0.247	0.802	0.922	0.964
Jiang et al. [39]	0.128	5.299	1.037	0.224	0.837	0.939	0.971
Li et al. [35]	0.104	4.513	0.697	0.164	0.868	0.967	0.990
Wang et al. [54]	0.096	0.655	4.327	0.171	0.893	0.963	0.983
Alhashim et al. [37]	0.093	4.170	0.589	0.171	0.886	0.965	0.986
Liu et al. [53]	0.120	4.977	-	-	0.838	0.948	0.980
Chen et al. [33]	0.083	3.599	0.437	0.127	0.919	0.982	0.995
Fu et al. [50]	0.072	**2.727**	0.307	0.120	0.932	0.984	0.994
Lee et al. [20]	**0.064**	2.815	**0.254**	**0.100**	**0.950**	**0.993**	**0.999**
Our (SENet-154)	0.070	2.912	0.382	0.121	0.942	0.986	0.992

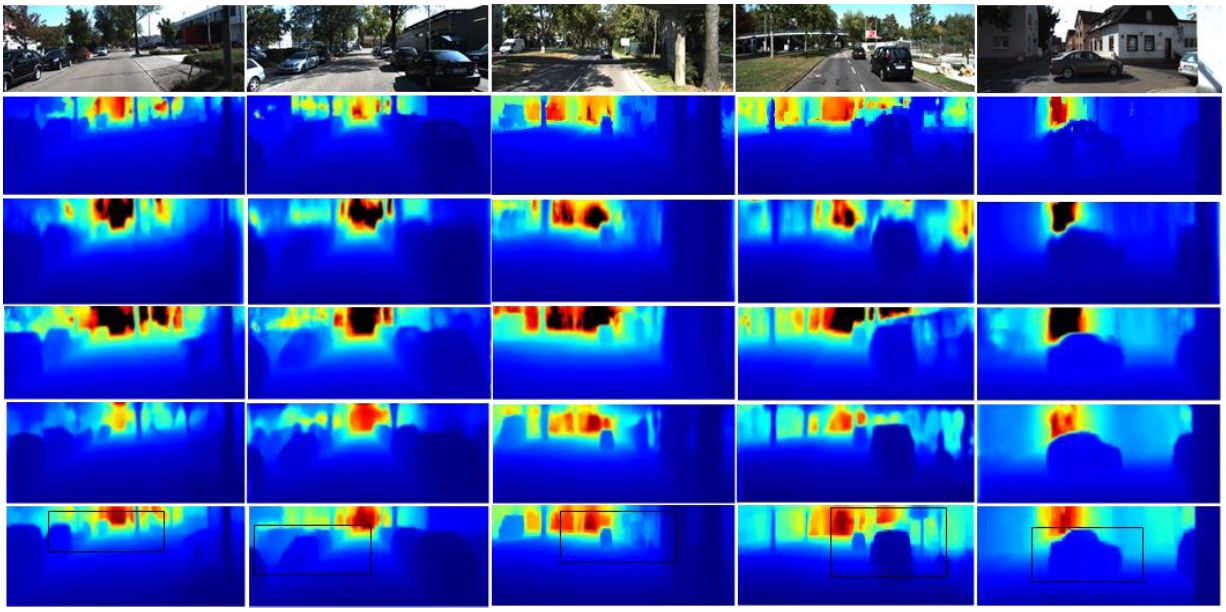


FIGURE 6. Qualitative evaluations on KITTI. Compared with the state-of-the-art methods. Color indicates depth (red is far, blue is close). The columns from top to bottom are RGB images, ground truth depth maps, results of Godard et al. [21], Garg et al. [52], Liu et al. [53], and the proposed method.

maps we predicted also have a gradually changing property in the distance which is consistent with the real scene distribution, as shown in the first, third and last rows. Although our method could provide better object boundaries and geometric details, but the evaluation metrics are not by large better than other state-of-the-art methods. It is because the object boundaries and geometric details in RGB images are not clearly demonstrated in the ground truth depth maps, such as the first and last rows in Fig. 5, so our fine-grained prediction lowered the scores instead. It demonstrates that quality of the depth map is the bottleneck to get better depth prediction.

We also evaluated our module on KITTI dataset. We selected the SENet-154 as our feature extraction network. The comparative results of evaluation metrics were reported in Table 2. It can be observed that our method outperforms most competitors. Quantitative comparisons were shown in Fig. 6 and more qualitative results were given in Fig. 13(b). As shown, our method provides finer boundaries of objects even in complex environments.

2) MODEL SIZE, RUNNING TIME, AND CONVERGENCY

Model size and running time are both crucial factors in determining the potential of DCNNs. Because both previous approaches and ours use standard DCNNs as encoder, we gave a comparison of the decoder size and the accuracy in Fig. 7. For the running time, we compared frames per second (FPS) and accuracy in test phase, which was depicted in Fig. 8. Figures of the mode size and running time for the compared methods were all obtained by running their open-source code in our machine.

As shown in Fig. 7 and Fig. 8, our proposed method makes a good balance between performance, model size, and running time. The accuracy of our proposed model is only less than [20] and [27]. But [20] has 2.5 times and [27] has 3.1 times more parameters compared with ours. Our running time is also less as compared with [27].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030097, IEEE Access

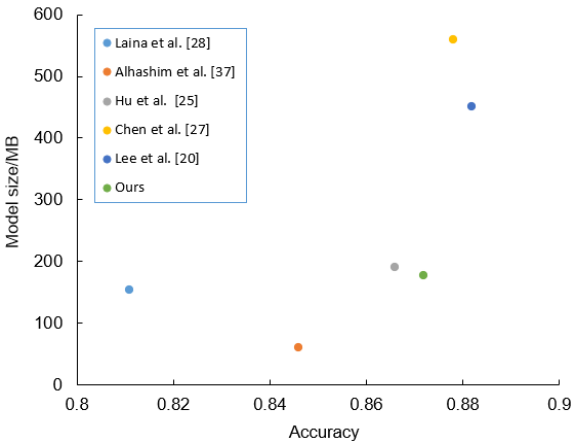


FIGURE 7. Model size and performance comparison. Comparison with previous methods. Model size (lower is better) vs. δ_1 (higher is better).

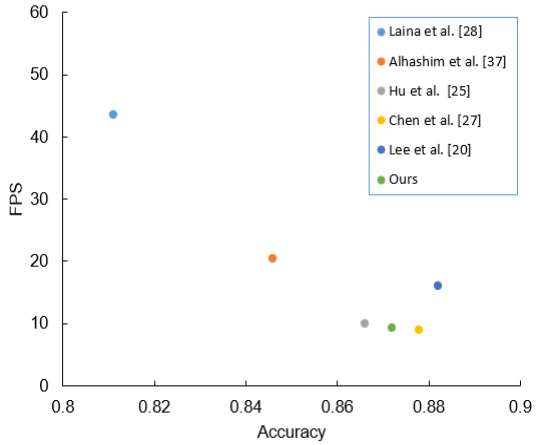


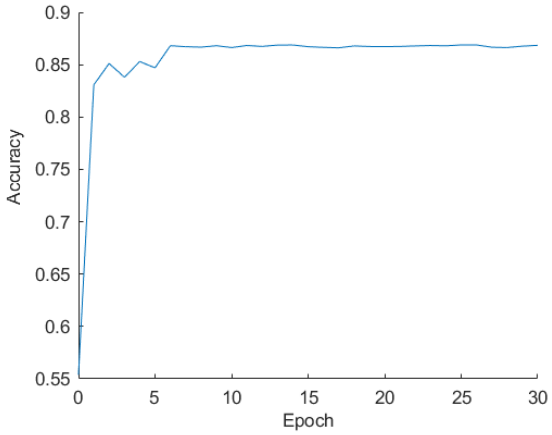
FIGURE 8. Running time and performance comparison. Comparison with previous methods. FPS (higher is better) vs. δ_1 (higher is better).

We gave the training loss curve and accuracy curve of our proposed method in training phase in Fig. 9. The training loss and accuracy were recorded once after every epoch. As shown, after several epoch, the training loss and accuracy are both stable within a small range. These curves support the convergency of our proposed method.

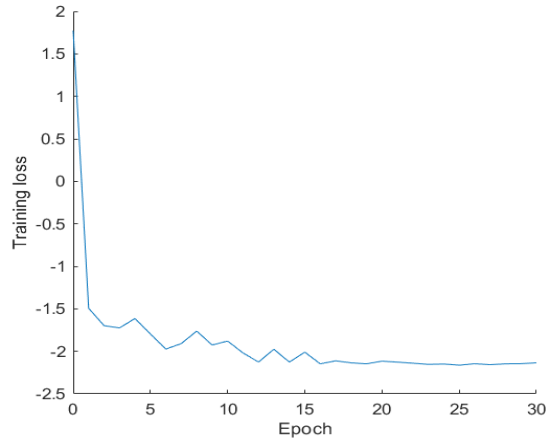
The experiments above were all made on NYU Depth v2 dataset.

D. ABLATION STUDIES

In this Section, we conducted ablation studies about our DMUB, ARFB and weight adjustment strategy. These ablation studies were all made on NYU Depth v2 dataset. We selected the SENet-154 as our feature extraction network. In order to clearly justify the effectiveness of each part, we set four baselines in the ablation studies that are illustrated as follows.



(a) Accuracy convergence during the first 30 epochs. Accuracy is represented by δ_1.



(b) Training loss convergence during the first 30 epochs.

FIGURE 9. Convergence of our proposed network.

- SENet- UAV (SENet-154 with Up-projection, ARFB, and weight of each loss term is Variable by the weight adjustment strategy): The proposed method without DMUB, but replaced with Up-projection block [28], which is a commonly used multi-scale convolutions.
- SENet-DRV (SENet-154 with DMUB, Residual fusion block, and weight of each loss term is Variable by the weight adjustment strategy): The proposed method without ARFB, but replaced with commonly used residual block.
- SENet-DAC (SENet-154 with DMUB, ARFB, and weight of each loss term is Constant): The proposed method without weight adjustment strategy.
- SENet-DAV (SENet-154 with DMUB, ARFB, and weight of each loss term is Variable by the weight adjustment strategy): The proposed method.

We explain the study of each baseline in detail as follows.

1) ABLATION STUDY FOR DMUB

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030097, IEEE Access

TABLE 3. Baseline comparison on NYU Depth v2. SENet-UAV and SENet-DAV verify the effectiveness of the DMUB. SENet-DRV and SENet-DAV verify the effectiveness of the ARFB. SENet-DAC and SENet-DAV verify our weight adjustment strategy.

Method	Error (lower is better)			Accuracy (higher is better)		
	Abs Rel	RMS	log10	$\delta_1(\delta<1.25)$	$\delta_2(\delta<1.25^2)$	$\delta_3(\delta<1.25^3)$
SENet-UAV	0.117	0.541	0.051	0.862	0.971	0.992
SENet-DRV	0.114	0.527	0.049	0.870	0.974	0.992
SENet-DAC	0.117	0.525	0.050	0.862	0.973	0.993
SENet-DAV	0.113	0.523	0.049	0.872	0.975	0.993

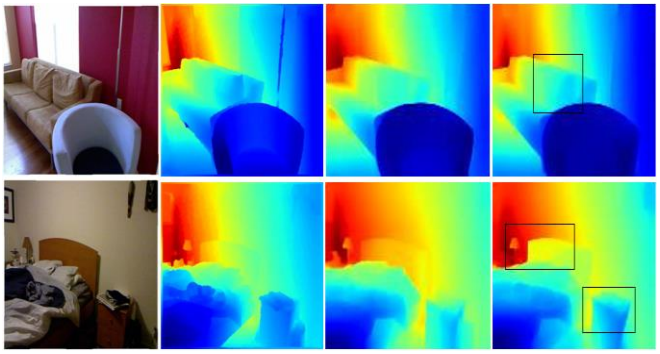


FIGURE 10. Qualitative evaluations of different baseline. Color indicates depth (red is far, blue is close). The columns from left to right are RGB images, ground truth depth maps, results of SENet-UAV and SENet-DAV, respectively.

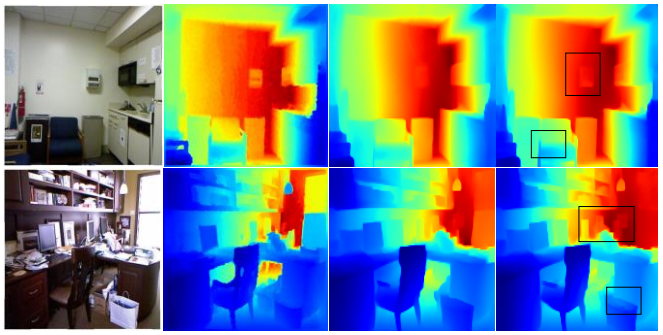


FIGURE 11. Qualitative evaluations of different baseline Color indicates depth (red is far, blue is close). The columns from left to right are RGB images, ground truth depth maps, results of SENet-DRV and SENet-DAV, respectively.

In order to justify the effectiveness of our DMUB, SENet-UAV and SENet-DAV were selected. The comparative results of evaluation metrics were reported in Table 3. Quantitative comparison was presented in Fig. 10.

As shown in Table 3, after replacing Up-projection block with our proposed DMUB, all metrics are improved. For instance, compared with Up-projection block, DMUB causes REL decrease by 2.5%, RMS decrease by 4.3%, log 10 error decrease by 3.9%. Our DMUB also has little model size than Up-projection block. Total model size of DMUB is 29.2M less than that of Up-projection block in our module.

As shown in Fig. 10, our DMUB provides better details and object boundaries, such as sofa cushion in the first row, small table lamp and wooden bed furniture in the second row. We could make a conclusion in this ablation study that our proposed DMUB is smarter and more accuracy than Up-projection block in monocular depth prediction.

2) ABLATION STUDY FOR ARFB

For the ablation study of our ARFB, SENet-DRV and SENet-DAV were selected. We implemented the residual fusion block in SENet-DRV by deducting the CAM and SAM in Fig. 3. The comparative results of evaluation metrics were reported in Table 3. Quantitative comparison was presented in Fig. 11.

As shown in Table 3, our ARFB is better than residual fusion block with little performance improvement. But discrimination of details is enhanced by our ARFB, as shown in Fig. 11. Meanwhile, with our ARFB, some unclear boundaries and categories are now clear, clean-cut, contrast good, such as the item hanging on the wall in the first picture and bags on the ground in the second picture. The feature integration among channel maps based on the joint attention mechanisms helps to capture context information that is useful to maintain semantic consistency, which is useful to the fine-grained prediction.

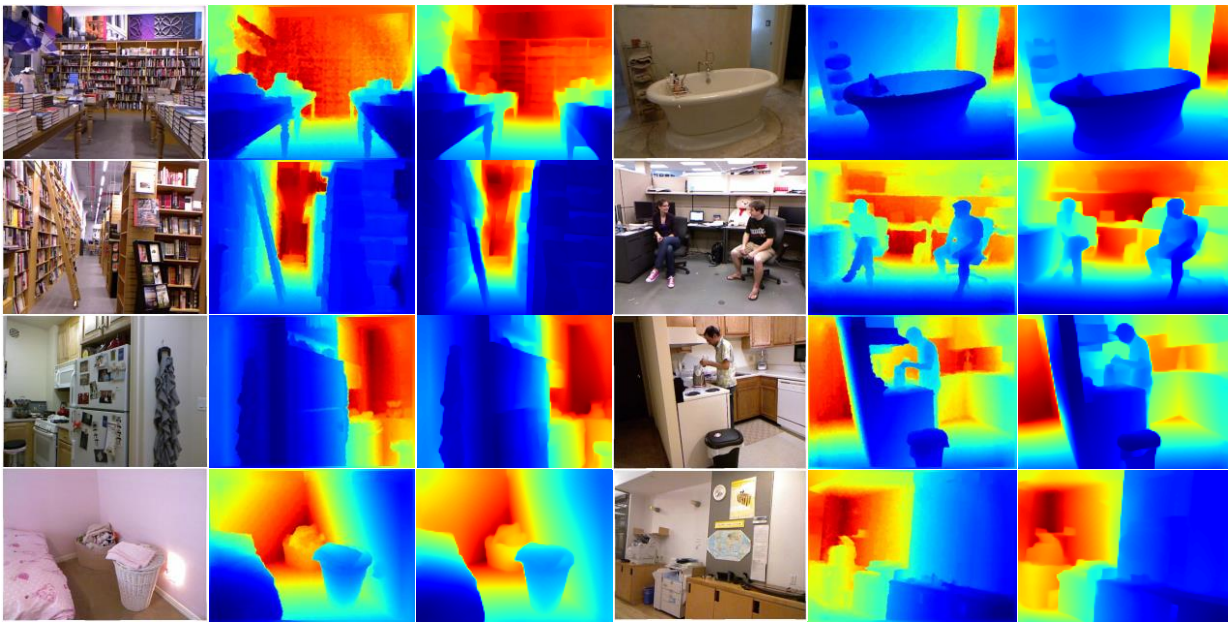
3) ABALATION STUDY FOR ADAPTIVE WEIGHT ADJUSTMENT STRATEGY

To evaluate the influence of our adaptive weight adjustment strategy, SENet-DAC, and SENet-DAV were selected. The weight of each loss term in SENet-DAC was set to be invariable (one as the normally selected), to compare the performance with our weight adjustment strategy. The results were presented in Table 3. Effects of the adjustment strategy was visualized in Fig. 12.

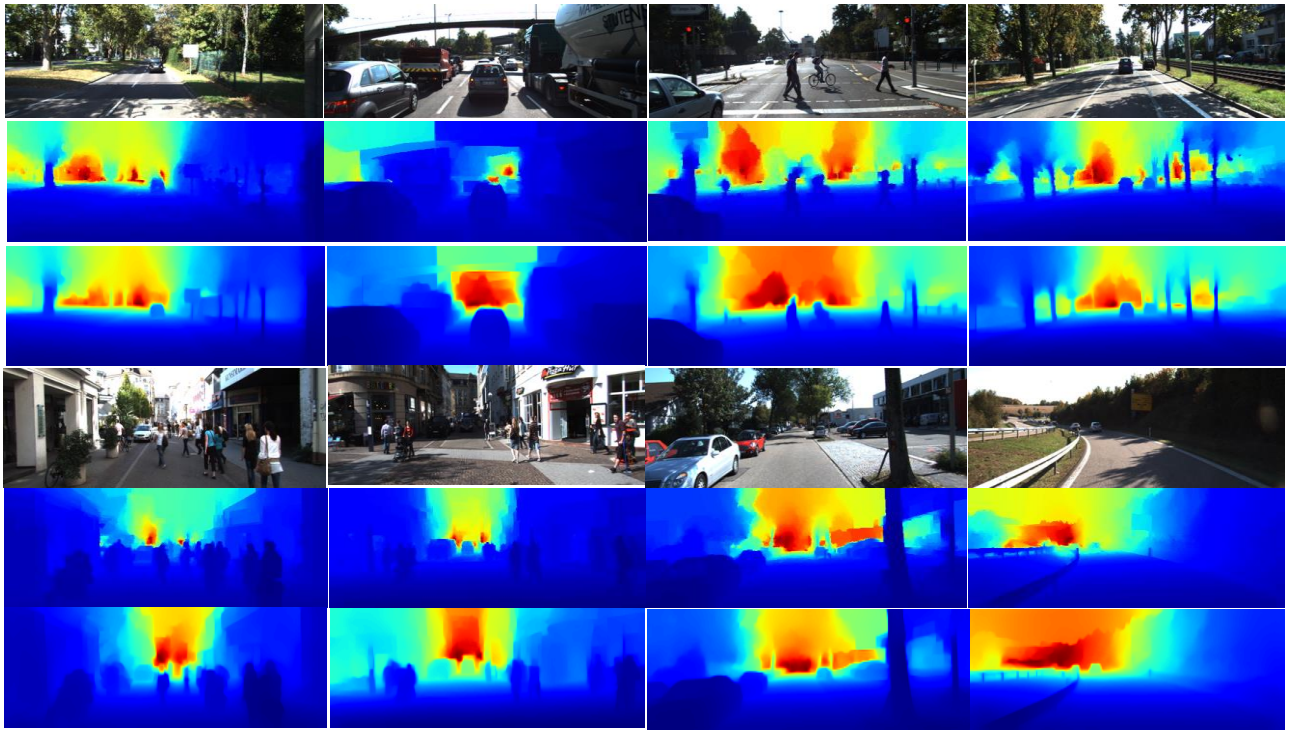
As shown in Table 3, compared with the model with fixed weights, our model with optimized weights obtains better results in nearly all evaluation metrics.

As shown in Fig. 12, it obviously demonstrates that our weight adjustment strategy could present clearer object boundaries compared with weight invariable method. These results reveal that the weight adjustment strategy gives a suitable consideration of these loss functions and optimize the weights respectively according to the value of each loss term. In this way, our loss function optimizes the learning process of the network better than the weight invariable loss function.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030097, IEEE Access



(a) More results on NYU Depth v2. Color indicates depth (red is far, blue is close). The columns from left to right are RGB images, ground truth depth maps and predicted depth maps by our model.



(b) More results on KITTI. Color indicates depth (red is far, blue is close). The columns from top to bottom are RGB images, ground truth depth maps and predicted depth maps by our model.

FIGURE 13. More qualitative results on NYU Depth v2 dataset and KITTI dataset.

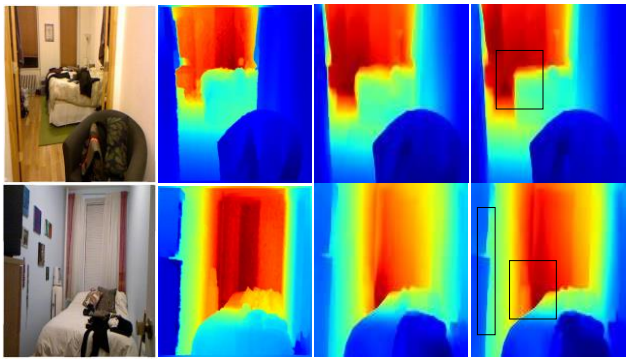


FIGURE 12. Qualitative evaluations of different baseline. Color indicates depth (red is far, blue is close). The columns from left to right are RGB images, ground truth depth maps, results of SENet-DAC and SENet-DAV, respectively.

V. CONCLUSION

In this paper, we have designed a deep end-to-end learning framework for monocular depth estimation. Our proposed framework improves the quality of predicted depth map from the following three aspects. Firstly, we design a lightweight and accuracy up-convolution to generate multi-scale feature maps. Secondly, we introduce joint attention mechanisms in our framework to enhance the discriminant ability of feature fusion at finer details. Thirdly, the weight adjustment strategy adaptively and dynamically balances the contribution of different loss terms. The experimental results have proved the effectiveness of our proposed framework. Our network achieves outstanding performance consistently on NYU Depth v2 and KITTI datasets and makes a good trade-off between accuracy, running time, and model size. Experiments are also conducted to verify the contribution of each individual aspect. Future work will focus on the improvement of the quality of ground-truth depth map which is critical for better depth prediction.

REFERENCES

[1] M. Naseer, S. Khan, and F. Porikli, "Indoor scene understanding in 2.5/3D for autonomous agents: A survey," IEEE Access, vol. 7, pp. 1859-1887, 2018.

[2] P. Chen, A. Liu, Y. Liu, and Y. Wang, "Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation," in Proc. CVPR, 2019, pp. 2624-2632.

[3] Z. Li, T. Dekle, F. Cole, and R. Tucker, "Learning the depths of moving people by watching frozen people," in Proc. CVPR, 2019, pp. 4521-4530.

[4] K. Makantasis, A. Doulamis, N. Doulamis, and M. Ioannides, "In the wild image retrieval and clustering for 3D cultural heritage landmarks reconstruction," Multimedia Tools Appl., vol. 75, no. 7, pp. 3593-3629, 2016.

[5] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," Int.J. Robot. Res., vol. 34, no. 4, pp. 705-712, 2015.

[6] D. Ball, P. Ross, A. English, P. Milani, D. Richards, and A. Bate, "Farm workers of the future: Vision-based robotics for broad-acre agriculture," IEEE Robot. Autom. Mag., vol. 24, pp. 97-107, 2017.

[7] H. Xu and M. Jiang, "Depth prediction from a single image based on non-parametric learning in the gradient domain," Optik, vol. 181, pp. 880-890, 2019.

[8] V. Hedau, D. Hoiem, and D. Forsyth, "Thinking inside the box: Using appearance models and context based on room geometry," in Proc. ECCV, 2010, pp. 224-237.

[9] A. Sellent and P. Favaro, "Optimized aperture shapes for depth estimation," Pattern Recognit. Lett. vol. 40, pp. 96-103, 2014.

[10] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in Proc. CVPR, 2010, pp. 1253-1260.

[11] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in Proc. CVPR, 2014, pp. 89-96.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Image-Net classification with deep convolutional neural networks," in Proc. NIPS, 2012, pp. 1097-1105.

[13] C. Xu, C. Lu, X. Liang, J. Gao, Z. Wei, T. Wang, and S. Yan, "Multi-loss regularized deep neural network," IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 12, pp. 2273-2283, 2016.

[14] K. Nogueira, O. Penatti, and J. A. D. Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," Pattern Recognit., vol. 61, pp. 539-556, 2017.

[15] Q. Zhou, B. Zheng, W. Zhu, and J. L. Longin, "Multi-scale context for scene labeling via flexible segmentation graph," Pattern Recognit., vol. 59, pp. 312-324, 2016.

[16] F. Liu, G. Lin, and C. Shen, "CRF learning with CNN features for image segmentation," Pattern Recognit., vol. 48, pp. 2983-2992, 2015.

[17] S. Bu, P. Han, Z. Liu, and J. Han, "Scene parsing using inference embedded deep networks," Pattern Recognit., vol. 59, pp. 188-198, 2019.

[18] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," Pattern Recognit., vol.71, pp. 132-143, 2017.

[19] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in Proc. NIPS, 2014, pp. 2366-2374.

[20] J. Lee, M. Han, D. W. Ko, and H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," arXiv preprint, arXiv: 1907.10326v5.

[21] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in Proc. CVPR, 2017, pp. 270-279.

[22] J. Xie, R. Girshick, and A. Farhadi, "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks," in Proc. NIPS, 2016, pp. 842-857.

[23] S. Zhao, L. Zhang, Y. Shen, S. Zhao, and H. Zhang, "Super-resolution for monocular depth estimation with multi-scale sub-pixel convolutions and a smoothness constraint," IEEE Access, vol. 7, pp. 16323-16335, 2018.

[24] Z. Zheng, C. Xu, J. Yang, Y. Tai, and L. Chen, "Deep hierarchical guidance and regularization learning for end-to-end depth estimation," Pattern Recognit., vol. 83, pp. 430-442, 2018.

[25] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in Proc. WACV, 2019, pp. 1043-1051.

[26] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multiscale continuous CRFs as sequential deep networks for monocular depth estimation," in Proc. CVPR, 2017, pp. 5354-5362.

[27] X. Chen, X. Chen, and Z. Zha. "Structure aware residual pyramid network for monocular depth estimation," in Proc. IJCAI, 2019. pp. 694-700.

[28] I. Laina, C. Rupprecht, and V. Belagiannis, "Deeper depth prediction with fully convolutional residual networks," In Proc. 3DV, 2016, pp. 239-248.

[29] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 11, pp. 431-444, 2018.

[30] J. Park, S. Woo, and J. Y. Lee, "BAM: bottleneck attention module," arXiv preprint, arXiv: 1807.06514v2.

[31] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks," in Proc. MICCAI ,2018, pp. 421-429.

[32] J. Jiao, Y. Cao, Y. Song, and R. Lau, "Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss," in Proc. ECCV, 2018, pp.53-69.
[33] Y. Chen, H. Zhao, and Z. Hu, "Attention-based context aggregation network for monocular depth estimation," arXiv preprint, arXiv: 1901.10137v1.
[34] R. Li, K. Xian, and C. Shen, "Deep attention-based classification network for robust depth prediction," in Proc. ACCV 2018, pp 663-678.
[35] B. Li, Y. Dai, and M. He, "Monocular depth estimation with hierarchical fusion of dilated CNNs and soft-weighted-sum inference," Pattern Recognit., vol. 83, pp. 328-339, 2018.
[36] S. Kong and C. Fowlkes, "Pixel-wise attentional gating for parsimonious pixel labeling," arXiv preprint, arXiv: 1805.01556.
[37] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," arXiv preprint, arXiv: 1812.11941v2.
[38] S. Gur and L. Wolf, "Single image depth estimation trained via depth from defocus cues," in Proc. CVPR, 2019, pp. 7683-7692.
[39] J. Jiang, H. Ehab, and X. Zhang, "Gaussian weighted deep modeling for improved depth estimation in monocular images," IEEE Access, vol. 7, pp. 134718-134729, 2019.
[40] T. Fabio, A. Filippo, P. Matteo, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo knowledge," in Proc. CVPR, 2019, pp. 9799-9809.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. CVPR, 2016, pp. 770-778.
[42] G. Huang, Z. Liu, V. D. M. Laurens, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. CVPR, 2017, pp. 2261-2269.
[43] N. Ibtehaz and M. Rahman, "MultiRes U-Net: Rethinking the U-Net architecture for multimodal biomedical image segmentation," Neural Netw., vol.121, pp. 74-87, 2020.
[44] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in Proc. CVPR, 2018, pp. 286-301.
[45] S. Woo, J. Park, J. Lee, and I. S. Korea, "CBAM: Convolutional block attention module," in Proc. ECCV, 2018, pp. 3-19.
[46] N. Silberman, D. Hoiem, D. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in Proc. ECCV, 2012, pp. 746-760.
[47] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in Proc. CVPR, 2012, pp. 3354-3361.
[48] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint, arXiv: 1412.6980.
[49] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "ImageNet: A large-scale hierarchical image database," in Proc. CVPR, 2009, pp. 248-255.
[50] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in Proc. CVPR, 2018, pp. 2002-2011.
[51] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," In Proc. CVPR, 2018, pp. 7482-7491.
[52] R. Garg, K. Vijay, G. Carneiro, and I. Reid. "Unsupervised CNN for single view depth estimation: geometry to the rescue," in Proc. ECCV, 2016, pp. 740-756.
[53] J. Liu, Y. Zhang, J. Cui, Y. Feng, and L. Pang, "Fully convolutional multi-scale dense networks for monocular depth estimation," IET Comput. Vis., vol.13, no. 3, pp. 515-522, 2019.
[54] J. Wang, G. Zhang, M. Yu, and T. Xu, "Attention-Based Dense Decoding Network for Monocular Depth Estimation," IEEE Access, vol. 8, pp. 85802-85812, 2020.

PENG LIU received his B.S. degree in automation from Nanchang Hangkong University, Nanchang, China, in 2005 and M.S. degree in control theory and control engineering from Dalian University of Technology, Dalian, China, in 2008, and he is currently pursuing his Ph.D. degree in mechanical engineering from Hebei University of Technology, Tianjin, China.

In 2011, he joined School of Intelligence and Information Engineering, Tangshan University, Tangshan, China, where he is currently a lecturer. His current research interests include 3D reconstruction, pattern recognition, deep learning, and computer vision.

ZONGHUA ZHANG is a full professor in the School of Mechanical Engineering, Hebei University of Technology, Tianjin, China. He received his Ph.D. degree from Tianjin University, Tianjin, China, in 2000.

He worked in Ruhr University Bochum of Germany, Queen's University of Canada, Heriot-Watt University and University of Leeds of UK. His main research interests include 3D optical measurement, fringe projection profilometry, and phase measuring deflectometry. He has published more than 170 papers. From 2016 to 2018, he was an EU Marie Curie Individual Fellow. He is an Associate Editor for Optics Express now.

ZHAOZONG MENG (M'14) received his B.S. and M.S. degrees in measurement and control technology and instrument from Sichuan University, Chengdu, China in 2006, and Beihang University, Beijing, China in 2009, respectively, and his Ph.D. degree in computer science from University of Huddersfield, West Yorkshire, UK, in 2014.

From 2014 to 2016, he was a research associate with the University of Manchester, and from 2016 to 2018, he was a research fellow with the University of Southampton. Since 2018, he has been a lecturer with School of Mechanical Engineering, Hebei University of Technology, Tianjin, China. Since 2020, he has been promoted an associate professor. His research interests include advanced sensor techniques, wearable devices and body area network, manufacturing intelligence, industrial IoT and cyber-physical systems.

NAN GAO was born in Tianjin, China in 1982. He received the B.S., M.S., and Ph.D. degrees in biomedical engineering from Tianjin University, Tianjin, China, in 2006, 2008 and 2012.

Since 2012, he has been a Lecturer and Associate Professor with the School of Mechanical Engineering, Hebei University of Technology. He is the author of more than 30 articles. His research interests include three-dimensional measurement of structured light projection, infrared absorption spectrum detection, machine vision and image processing.