Monocular Depth Prediction With Residual DenseASPP Network

KEWEI WU, SHUNRAN ZHANG, AND ZHAO XIE (Member, IEEE)

1 Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Ministry of Education, Hefei 230601, China
2 School of Computer and Information, Hefei University of Technology, Hefei 230601, China

Corresponding author: Kewei Wu (wu_kewei1984@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1002203, in part by the National Natural Science Foundation of China under Grant 61503111, in part by the Anhui Natural Science Foundation under Grant 1808085MF168, and in part by the Anhui Province Key Laboratory of Industry Safety and Emergency Technology.

ABSTRACT Monocular depth estimation is an ill-posed problem because infinite 3D scenes can be projected to the same 2D scenes. Most recent methods focus on image-level information from deep convolutional neural networks, while training them may suffer from slow convergence and accuracy degeneration, especially for deeper network and more feature channels. Based on an encoder-decoder framework, we propose a novel Residual DenseASPP Network. In our Residual DenseASPP network, we define features as low/mid/high vision features and use two-kinds of skip connection to learn useful features with certain layers, where feature concentration in the dense block is used to generate more features in the same layer, and feature summation in the residual block is used to increase backward gradient. The experimental results show that high vision features require more channels by feature concentration, while low/mid vision features need better convergence by feature summation. Experiments show that our proposed approach achieves state-of-the-art performance on both NYUv2 and Make3D datasets.

INDEX TERMS Depth estimation, residual DenseASPP network, feature reuse, deep learning.

I. INTRODUCTION Monocular depth estimation aims to estimate depth information of a scene from an RGB image, which is a fundamental problem of computer vision with many potential applications, such as semantic segmentation [1], [2], object detection [3]–[5], human pose estimation [6], [7], 3D reconstruction [8], simultaneous localization and mapping [9]. Inspired by the remarkable success of image classification, most recent methods study this work based on deep networks. They learn visual representations for depth estimation in an end-to-end multi-layer fashion [10], in which features of various receptive fields are generated by convolution and pooling operations. The resolution of the features reduces along with pooling operations, which leads that the final features do not contain sufficient detail information. These features are not beneficial for model convergence. Atrous Spatial Pyramid Pooling [11] (ASPP) has been studied to alleviate this problem, which can generate features with abundant receptive fields by concatenating outputs of multiple atrous convolutions [12] with different dilation rates. More works search for a better architecture for Atrous Spatial Pyramid Pooling and provide a framework of Densely connected Atrous Spatial Pyramid Pooling [13] (DenseASPP). They fully exploit the advantages of parallel and cascade architectures of the dilated convolution layer. Nevertheless, the above methods still suffer from vanishing gradient, especially for the deeper network. Vanishing gradients [14], [15] is the case that accuracy gets saturated and then degrades rapidly with the network depth increasing. Two success methods for vanishing gradient are ResNet [16] and DenseNet [17]. ResNet considers a shortcut connection [18]–[20] skipping one or more layers can avoid vanishing gradient, and furtherly, DenseNet uses dense connections among multi-layers.

However, monocular depth prediction is still challenging, especially for complex depth patterns, including small objects, complex boundaries, illuminations, objects with a big change in depth. There are two disadvantages to the process of learning complex depth patterns in existing methods: (1) In some existing applications, ResNet can outperform DenseNet. This suggests that although skip connections can increase gradient, lots of skip connections in DenseNet may
not more effective than simple connections in ResNet. Skip connections among certain features can be invalid especially when the combinations of these features do not exist in the real world. Existing methods can not fully interpret what connections are contributing. (2) It is hard to learn mass parameters for skip connections in DenseNet. This case aggravates in a DenseNet with multi-scale ASPP, which can cause the model overfitting. This suggests that searching better architecture to reduce model complexity is not sufficient in depth estimation.

To solve the above problems, we propose a novel Residual DenseASPP Network based on an encoder-decoder framework. We consider pattern combinations vary in different receptive fields, and design three decomposed blocks for low/mid/high vision features. We fully exploit two-kinds of skip connection in a feed-forward fashion and suggest an effective architecture for depth estimation by fusing two-kinds of skip connection.

To verify the superiority of the proposed model, extensive experiments are performed on two large-scale databases. Experiments show that our proposed approach achieves state-of-the-art performance on both NYUv2 and Make3D datasets.

The main contributions of our work lie in three folds:
(1) We propose a novel Residual DenseASPP Network, in which we fully exploit network architecture for low/mid/high vision features by fusion two-types of skip connection. In our network, certain types of skip connections are designed for certain layers, which can improve deep feature learning for depth estimation.
(2) We search for flexibility architecture with different model complexity and design various ablation models of our Residual DenseASPP Network. We design low/mid vision features module with less connection and fewer feature channels, which can give improvement for depth estimation.
(3) Experiments show that our proposed approach achieves state-of-the-art performance on both NYUv2 and Make3D datasets. The visibility results prove that our method can give good predictions in many challenges, including small objects, complex boundaries, illuminations, objects with a big change in depth.

II. RELATED WORK
In monocular scene depth estimation, feature extraction from a single image plays a crucial role. We categorize the related depth estimation approaches into four types:

A. FULL CONVOLUTION WITHOUT PARALLEL MULTI-SCALE DEPTH MODEL
D. Eigen et al. used a single multiscale CNN to learn deep features [2]. CNN can be extended with harmonizing overcomplete local predictions [21], multiple candidates in the frequency domain [22], and the whole strip masking module [23]. Nevertheless, these works do not consider the relation between two regional features. Conditional Random Fields (CRFs) can model this pair-wise relation, and has been successfully introduced into CNNs [24]–[27] to improve the precision of depth maps. Yan et al. [28] applied the CRF model with additional constraints of the normal vector of the object surface, which can estimate the depth of multi-level scenes at the super-pixel level and pixel level. The key to these methods is what prior information can be introduced into CNNs to solve some challenges, such as fuzzy structure, unclear layers, and fusion of object depth and background. Depth gradient [29] and surface normal [30] are important edge cues for monocular depth estimation. And segmentation can provide regional cues to optimize depth estimation [1], [31]–[33]. Zhang et al. [34] provide a joint task-recursive learning framework for semantic segmentation and depth estimation. Some works [35], [36] also introduce pose predictor into depth prediction. Godard et al. [37] use per-pixel minimum reprojection to address the problem of occluded pixels. As a special application, monocular video can supply optical flow cues to estimate image depth [38]. Recently, Yang Wang et al. used UnOS (Unified Unsupervised Optical-Flow and Stereo-Depth Estimation) [39] to estimate image depth. Zhao et al. [40] designed a geometry consistency loss for binocular depth prediction. A. A. Abarghouei and Breckon [41] used synthetic images to predict depth.

B. FULL CONVOLUTION WITH PARALLEL MULTI-SCALE DEPTH MODEL
ASPP applies the parallel dilated convolution with different dilation rates to extract features at different scales. On this basis, the CRF model further optimizes the local inconsistencies between two regional features [42]–[44]. H. Fu et al. [45] used ASPP and cross-channel to learn multi-scale features. They design an ordinal regression loss function from pixel to pixel. However, there are many parameters in that full convolution with a parallel multi-scale depth model, and they still suffer from a serious gradient degradation problem.

C. DEPTH CONVOLUTION MODEL WITH DENSE BLOCK
DenseNet alleviates the vanishing gradient problem because it strengthens feature propagation and encourages feature reuse by feature concentration. Based on ASPP, Fu et al. [45] furtherly used multi-scale cascade dense CNN to train the RCNN model for depth estimation. Huang et al. [17] introduced the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. Jégou et al. [46] applied a dense cascade network of DenseNet with down-sampling and up-sampling for segmentation. Yang et al. [12] combined the ASPP with the dense cascade of dense density to construct the DenseASPP module. Gur et al. [47] provide an additional seminal work with CNN enhanced by point spread function and focus cues. They conducted a multi-scale sampling strategy to obtain more features.
D. DEPTH CONVOLUTION MODEL WITH RESIDUAL BLOCK

Unlike feature concentration in DenseNet, a residual module reuses them by feature summation [48]. Cao et al. [49] added a residual module on basic network FCN as an FC-residual module. Kuznietsov et al. [50] used ResNet as the backbone network and proposed special up-sampling for depth estimation.

However, we still cannot get the conclusion that what architecture is more reasonable. We also focus on what kind of low/mid/high vision features is important in depth estimation. Therefore, we propose a Residual DenseASPP model that fully exploits two-kinds of skip connection in a feed-forward fashion and suggest an effective architecture for depth estimation by fusing two-kinds of skip connection.

III. METHODOLOGY

In this section, we describe our Residual DenseASPP model for monocular depth prediction in detail.

A. NETWORK ARCHITECTURE

Our model used an encoder-decoder structure based on DeepLab V3+ [33], as shown in Fig 1. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, and the decoder module refines the depth results from these features. In the encoder, we using ResNet-101 as the backbone network of the Deep Convolutional Neural Network (DCNN) [33]. Unlike the basic encoder, we change the ASPP module to DenseASPP and integrate the residual module into DenseASPP to solve the vanishing gradients problem more effectively.

B. FROM ATROUS CONVOLUTION TO RESIDUAL DenseASPP

1) ATROUS CONVOLUTION

Atrous convolution can increase the receptive field while do not change the feature map resolution. ASPP generates multi-scale features by the parallel atrous convolution. In one dimensional case, consider two-dimensional signals, for each location \(i\) on the output feature map \(y\) and a convolution filter \(w\), atrous convolution is applied over input feature map \(x\) as follow:

\[
y[i] = \sum_{k} x[i + a \cdot k] \cdot w[k]
\]

where the atrous rate \(a\) determines the sampling stride, \(w[k]\) denotes the \(k\)-th parameter of filter, and \(K\) is the filter size. The receptive field can adaptively modify by changing the rate value. Note that standard convolution is a special case in which rate \(a = 1\). For an atrous convolutional layer, the receptive field size \(R\) is calculated as:

\[
R = (a - 1) \times (K - 1) + K
\]

2) DENSEASPP

Furtherly, DenseASPP connects more atrous layers with dense connections to obtain larger receptive fields. In our network, we use five atrous convolutional layers in a cascade fashion in DenseASPP, where the dilation rate of each layer increases layer by layer (3, 6, 12, 18, 24). We feed DenseASPP with a 64-dimensional feature map \(y_0\), which is transformed from a 2048-dimensional feature map \(y_{Resnet–101}\) of ResNet-101 by an FC layer. In DenseASPP,
each layer accepts a concatenated feature map, which contains the output of each lower atrous layer and the input feature map. The output of each atrous layer is an equal-sized 64-dimensional feature. We can write the output in terms of

\[ y^c_l = H_{K,a_l}(y^c_{l-1}, y^c_{l-2}, \ldots, y_0) \]  

(3)

where \( H \) is the atrous convolution, \( K \) is the filter size, \( a_l \) represents the dilation rate of layer \( l \), and \([-] \) denotes the concatenation operation. \([y^c_{l-1}, y^c_{l-2}, \ldots, y_0]\) means the feature map formed by concatenating the outputs of all previous layers, \( c \) indicates the output is extracted by dense connections way. The final output of DenseASPP is a feature map generated by multi-rate, multi-scale atrous convolutions.

C. RESIDUAL DenseASPP

Although DenseASPP increases the reusability of features of each layer, it still cannot tell us features of which layer are important in depth estimation. Therefore, we construct a Residual DenseASPP module, which contains three composed blocks to describe low/mid/high vision features separately. Basic DenseASPP only consists of low and mid vision features, in which the low vision features contain five atrous convolutional output, and the mid vision features indicate the final output. We additionally increase a block to combine the final output and the input from ResNet-101 to extract high vision features.

1) VARIANTS OF RESIDUAL DENSEASPP

**DDD.** DenseASPP can be simply extended to DDD, as shown in Fig 2. DDD, which is an ablation model of our Residual DenseASPP, where all three composed blocks adopt a dense block way. Consequently, the final high vision features of DDD can be formulated as:

\[ y_{DDD} = [y_{ResNet-101}, y^c_l, y^c_{l-1}, \ldots, y_0] \]  

(4)

We design six variants, named RRR, RRD, RDD, DRR, DDR, DDD, to discuss what kind of low/mid/high features is important in Residual DenseASPP. All variants have the same input features and the same dilation rates. Fig 2. shows the six variants, and Fig 3. shows their low/mid/high feature combination ways. Admittedly, DDD provides the most sufficient low/mid/high vision features in all variants, but DDD does not perform best on NYU Depth v2 dataset, which may because the dense block (D) cannot provide good gradients in feature learning, while the residual block (R) is an alternative way.

**RRR.** We generate another ablation model RRR, as shown in Fig 2. R means corresponding composed block by a residual way. In RRR, each layer accepts a summation feature map, which sums the output of each lower atrous layer and the input feature map. The output of each atrous layer is an equal-sized 64-dimensional feature. We can write the output of low vision features as:

\[ y^r_l = H_{K,a_l}(y^r_{l-1}, y^r_{l-2}, \ldots, y_0) \]  

(5)

where \([-] \) denotes the summation operation. \([y^r_{l-1}, y^r_{l-2}, \ldots, y_0]\) means the feature map formed by summing together the outputs of all previous layers. \( + \) indicates the output is extracted by residual connections way. We furtherly adopt residual connections to extract mid/high vision feature, and then the final high vision features of RRR can be formulated as:

\[ y_{RRR} = [y_{ResNet-101}, y^r_l, y^r_{l-1}, \ldots, y_0] \]  

(6)

**RRD.** We replace the high vision feature connection in RRR with a dense block to get RRD. As shown in Fig 3, RRD can provide more high vision features than RRR. The final high vision features of RRD can be formulated as:

\[ y_{RRD} = [y_{ResNet-101}, y^r_l, y^r_{l-1}, \ldots, y_0] \]  

(7)

**RDD.** RRD can provide more mid/high vision features than RRR, and its final high vision features can be formulated as:

\[ y_{RDD} = [y_{ResNet-101}, y^r_l, y^r_{l-1}, \ldots, y_0] \]  

(8)

**DRR.** DRR can provide more low vision features than RRR, and its final high vision features can be formulated as:

\[ y_{DRR} = [y_{ResNet-101}, y^r_l, y^r_{l-1}, \ldots, y_0] \]  

(9)

**DDR.** DDR can provide more low/mid vision features than RRR, and its final high vision features can be formulated as:

\[ y_{DDR} = [y_{ResNet-101}, y^r_l, y^r_{l-1}, \ldots, y_0] \]  

(10)

D. LOSS

In our method, we use L2 as our loss function. L2 is sensitive to outliers in the training data since it penalizes more heavily on larger errors. It minimizing the squared euclidean norm between predictions \( g^* \) and ground truth \( g \):

\[ L_2(g^* - g) = \|g^* - g\|_2^2 \]  

(11)

IV. EXPERIMENTS

To verify the superiority of our method, we provide extensive experimental results on challenging NYU Depth v2 and Make3D to compare with state-of-the-art works. The evaluation metrics are following previous works [24].

A. DATASETS

1) NYU DEPTH V2 DATASET [51]

We evaluate our method on one of the largest RGB-D data sets for indoor scene reconstruction, NYU Depth v2. The raw dataset consists of 464 scenes, captured with a Microsoft Kinect, with the official split consisting of 249 training and 215 test scenes. In our experiments, we consider a subset of 1449 RGB-D pairs, of which 795 are used for training and the rest for testing. In particular, we conducted data augmentation (scale scaling, rotation, color transformation, flip) to enlarge the training set. As a result, we obtained 87,000 training pairs in total. Then we down-sample the original frames of size \( 640 \times 480 \) pixels to 1/2 resolution and center-cropped to \( 304 \times 228 \) pixels, as input to the network.
FIGURE 2. Different variants of Residual DenseASPP. We name each variants according to the features combination way in low/mid/high vision feature, where c means the input features are combined by dense connections way (D), and + means the input features are combined by residual connections way (R).
FIGURE 3. The features reuse in different variants of Residual DenseASPP. The dense connections increase the number of feature channels, and the residual connections do not change the number of feature channels. Fig. 3. directly shows the number of feature channels and the complexity of the method. We analyze what kind of low/mid/high features is important for prediction by the visualization of features reuse.
TABLE 1. Comparison with state of the art methods on NYU Depth V2.

| Methods          | Deep network architectures                          | High is better | Low is better |
|------------------|-----------------------------------------------------|----------------|---------------|
|                  |                                                     | \( \delta_1 \) | \( \delta_2 \) | \( \delta_3 \) | rel | rmse | log10 |
| CNN-based        |                                                     |                |               |               |     |      |       |
| P. Wang et al. [27] (2015) | CNN with regional path and global path            | 0.605           | 0.890          | 0.970          | 0.220 | 0.824 | NA    |
| F. Liu et al. [24] (2016) | CNN with CRF                                        | 0.650           | 0.906          | 0.976          | 0.213 | 0.759 | 0.087 |
| D. Eigen et al. [2] (2015) | CNN                                                   | 0.769           | 0.950          | 0.988          | 0.158 | 0.641 | NA    |
| A. Chakrabarti et al. [21] (2016) | CNN with scene path and local path                  | 0.806           | 0.958          | 0.987          | 0.149 | 0.620 | NA    |
| D. Xu et al. [42] (2017) | CNN with CRF                                        | 0.811           | 0.954          | 0.987          | 0.121 | 0.586 | 0.052 |
| J. Lee et al. [22] (2018) | CNN combination in Fourier domain                   | 0.815           | 0.963          | 0.991          | 0.139 | 0.572 | NA    |
| Y. Cao et al. [49] (2018) | CNN with CRF                                        | 0.819           | 0.965          | 0.992          | 0.141 | 0.573 | 0.055 |
| X. Qi et al. [30] (2018) | CNN with surface normal module                      | 0.834           | 0.960          | 0.990          | 0.128 | 0.569 | 0.057 |
| Encoder-decoder  |                                                     |                |               |               |     |      |       |
| J. Li et al. [29] (2017) | Encoder-decoder                                     | 0.789           | 0.955          | 0.988          | 0.152 | 0.611 | 0.064 |
| S. Gur et al. [47] (2019) | Encoder-decoder with DenseASPP, self-attention block | 0.772           | 0.942          | 0.984          | 0.162 | 0.575 | NA    |
| I. Laina et al. [48] (2016) | Encoder-decoder                                     | 0.811           | 0.953          | 0.988          | 0.127 | 0.573 | 0.055 |
| Z. Zhang et al. [34] (2018) | Encoder-decoder with task-attentional module        | 0.815           | 0.962          | 0.992          | 0.144 | 0.501 | NA    |
| M. Heo et al. [23] (2018) | Encoder-decoder using whole strip masking and reliability-based refinement | 0.816           | 0.964          | 0.992          | 0.135 | 0.571 | 0.058 |
| DDD [Ours]       | Encoder-decoder with Residual DenseASPP             | 0.820           | 0.966          | 0.995          | 0.141 | 0.485 | 0.059 |
| H. Fu et al. [45] (2018) | Encoder-decoder with ASPP                           | 0.828           | 0.965          | 0.992          | 0.115 | 0.509 | 0.051 |
| RRD [Ours]       | Encoder-decoder with Residual DenseASPP             | **0.841**       | **0.969**      | **0.994**      | **0.136** | **0.430** | **0.054** |

2) Make3D DATASET [52], [53]

We also test our algorithm on the outdoor dataset Make3D. We use official split with 400 pairs for training and 134 for testing. We train on an augmented data set of around 15k samples. Since Make3D expresses depths up to 80m only, the depths of far objects are often inaccurate. Errors in training are only calculated where depth is less than 70 meters in a central image crop.

B. IMPLEMENTATION DETAILS

We implement the proposed network using the open deep learning framework Pytorch framework with two NVIDIA 1080 GPUs with 16GB memory. As the encoder for feature extraction, we use ResNet-101 with pre-trained weights trained for image classification using the ILSVRC dataset [54]. The stochastic gradient descent (SGD) algorithm was started with a learning rate of 0.001 and decreased by 10 times around every 6 epochs. 24 epochs are performed for training. The momentum and weight decay were set to 0.9 and 0.0005, respectively. The number of batch size sets to 8. It took around 14 hours to train the Residual DenseASPP Network. To avoid over-fitting, we augment images before input to the network using random horizontal flipping, random contrast, brightness and color adjustment in ranges of [0.6, 1.4], with a 50% chance. We also use a random rotation of the input images in a range of \([-5, 5\) degrees. We train our network on a random crop of size \(304 \times 118\).

C. BASELINES

Baselines: We compare 21 state-of-the-art deep learning methods, including CNNs without parallel multi-scale depth model [2], [21]–[25], [27]–[30], [32], [34]–[40], [41], CNNs with parallel multi-scale depth model [42], CNNs with dense block [45], [47], CNNs with residual block [48], [49].

Variants of our method: We discuss what kind of low/mid/high features is important in our model by six variants, named RRR, RRD, RDD, DRR, DDR, DDD, which detailed in Section 3.3.

D. ERROR METRICS

For quantitative evaluation, we report errors obtained with the following error metrics, which have been extensively used. Denote \(g\) as the ground truth depth, \(g^e\) as the estimated depth, and \(T\) denotes the set of all points in the images.

- Mean Absolution Relative Error (rel): 
  \[
  \frac{1}{|T|} \sum_{g^e \in \mathbb{R}} \frac{|g^e - g|}{d}
  \]

- Root Mean Squared Error (rmse): 
  \[
  \sqrt{\frac{1}{|T|} \sum_{g^e \in \mathbb{R}} (g^e - g)^2}
  \]

- Mean log10 Error (log10): 
  \[
  \frac{1}{|T|} \sum_{g^e \in \mathbb{R}} |\log_{10}(g) - \log_{10}(g^e)|
  \]
ordinal regression module release spatial constraints to allow takes into account the ordering of discrete depth values. This probably because it uses an ordinal regression module which final Regression Network outperforms our RRD in rel [45], parameters and cause a little overfitting. Second, Deep Ordinal Regression Network outperforms our RRD in rel [42], and this suggests that multi-scale reception fields in ASPP are not enough for complex depth pattern. We notice that our RRD achieves better performances in terms of threshold accuracy and rmse as compared to CNN with dense block feature [45], and gives worse performances in terms of rel and log10. That is to say, dense block feature is efficient, and network architecture and its optimizer can furtherly impact threshold accuracy. We find that: (1) Our RRD model outperforms most CNNs without parallel multi-scale depth models. But CNN with surface normal cue [30] outperforms RRD in rel accuracy probably because prior knowledge can provide more detail features for depth estimation. (2) Our RRD model outperform CNN wish ASPP model [42], and this suggests that multi-scale reception fields in ASPP are not enough for complex depth pattern. (3) We notice that our RRD achieves better performances in terms of threshold accuracy and rmse as compared to CNN with dense block feature [45], and gives worse performances in terms of rel and log10. That is to say, dense block feature is efficient, and network architecture and its optimizer can furtherly impact threshold accuracy. (4) A residual learning framework can ease the training of networks [48], [49], but they do not fully express the different characteristics of low/mid/high vision patterns, therefore, they give worse performance than our RRD method. (5) Our RRD proves that low/mid/high vision patterns have different characteristics. The number of low/mid vision patterns is lower than the number of high vision patterns. And low/mid vision patterns need to be learned more precisely than high vision patterns, which may because low/mid vision patterns are the components of high vision patterns, and small change in low/mid vision patterns can be amplified in high vision patterns. Meanwhile, RRD has smaller feature channels in low/mid vision features than DDD and gives better performance than DDD. (6) Comparison with encoder-decoder with ASPP. Our work is similar to the DenseASPP with an added attention block [47] and the Deep Ordinal Regression Network [45] because they both use encoder with ASPP and decoder with upsampling and convnet. First, Our RRD outperforms DenseASPP with an added attention block [47], probably because it adds a convolutional path to DenseAspp rather than a direct skip connection, and appends a self-attention block to this new module. But these operations increase model parameters and cause a little overfitting. Second, Deep Ordinal Regression Network outperforms our RRD in rel [45], probably because it uses an ordinal regression module which takes into account the ordering of discrete depth values. This ordinal regression module release spatial constraints to allow a relatively larger difference in training. But our RRD still outperforms it in \( \delta_1, \delta_2 \) and rmse, because it only uses parallel multi-scale ASPP, while we enlarge more feature channels by DenseASPP and learn them better with the residual module. Table 2 shows the results of the comparison with state of the art methods on Make3D. Errors are computed for depths less than 70m in a central image crop [40]. Our methods outperform many methods. D. Xu et al. [42] outperform DDD in rel and rmse, probably because CRF smooths some noisy features. Fu et al. [45] give the best performance in rel, which shows that the denseblock is important for feature extraction. Our RRD gives the best performance in rmse due to optimized network architecture. In this part, we will analyze the results of the six ablation models mentioned above. Each ablation method takes a certain combination way for low/mid/high vision feature. The ablation results are shown in Table 3. We find that: (1) RRR performs worst in all variants, which may due to features are excessively reduced by summation operation. (2) DDD outperforms 0.104 in accuracy (with threshold <1.25) than RRR, which proves that a dense block can provide more sufficient features than a residual block. (3) Admittedly DDD performs lower than other 4 variants, including RRD, RDD, DRR, DDR, which proves that sufficient features are hard to learn and certain residual blocks can facilitate feature training. (4) Compared with DDD and RDD, better low vision features can increase 0.07 in the accuracy (with threshold <1.25). Furthermore, in RRD, better low/mid vision features can increase 0.19 in the accuracy (with threshold <1.25), and RRD performs best in all six variants. That is to say, low/mid vision features are less than high vision features, and more low/mid vision features can be redundant. Moreover, we need a residual block to learn more precise low/mid vision features. (5) Compared with DDD and DDR, better high vision features can increase 0.03 in the accuracy (with threshold <1.25). And in DRR, better mid/high vision features can increase 0.05 in the accuracy (with threshold <1.25). Precise mid/high vision features with a residual block are less important than low vision features. (6) Compared with RRR and RDD, high vision features need more channels, which may because high vision features with a larger receptive field size have more complex patterns. Figure 4 shows the depth prediction results of ablation models on NYU Depth v2. These instances have many challenges, including small objects, complex boundaries, illuminations, objects with a big change in depth. (1) A small object is hard to segment in a scene, and its depth is easy to be interfered with by background information. Small bins in row (c), small screens in row (i), small bottles in row (j) can be extracted by RRD. DDD also performs well on small objects, but RRR will lose them, which indicate that more high vision feature is good for a small object.

\[ \delta_i = \frac{\text{card}\left( \{ g_i^f : \max \left( g_i^f, g_i^l, g_i^s \right) < 1.25^i \} \right)}{\text{card}\left( \{ g_i \} \right)} \]

where \( \text{card} \) is the cardinality of a set. A higher \( \delta_i \) indicates better prediction.

E. COMPARISON WITH THE STATE-OF-THE-ART

Table 1 shows the results of the comparison with state of the art methods on NYU Depth V2. We can notice that our RRD model achieves 84.10% in the accuracy (with threshold <1.25) on the NYUv2 dataset, which is superior to state-of-the-art methods.

1. Our RRD model outperforms most CNNs without parallel multi-scale depth models. But CNN with surface normal cue [30] outperforms RRD in rel accuracy probably because prior knowledge can provide more detail features for depth estimation.

2. Our RRD model outperform CNN wish ASPP model [42], and this suggests that multi-scale reception fields in ASPP are not enough for complex depth pattern.

3. We notice that our RRD achieves better performances in terms of threshold accuracy and rmse as compared to CNN with dense block feature [45], and gives worse performances in terms of rel and log10. That is to say, dense block feature is efficient, and network architecture and its optimizer can furtherly impact threshold accuracy.

4. A residual learning framework can ease the training of networks [48], [49], but they do not fully express the different characteristics of low/mid/high vision patterns, therefore, they give worse performance than our RRD method.

5. Our RRD proves that low/mid/high vision patterns have different characteristics. The number of low/mid vision patterns is lower than the number of high vision patterns. And low/mid vision patterns need to be learned more precisely than high vision patterns, which may because low/mid vision patterns are the components of high vision patterns, and small change in low/mid vision patterns can be amplified in high vision patterns. Meanwhile, RRD has smaller feature channels in low/mid vision features than DDD and gives better performance than DDD.

6. Comparison with encoder-decoder with ASPP. Our work is similar to the DenseASPP with an added attention block [47] and the Deep Ordinal Regression Network [45] because they both use encoder with ASPP and decoder with upsampling and convnet. First, Our RRD outperforms DenseASPP with an added attention block [47], probably because it adds a convolutional path to DenseAspp rather than a direct skip connection, and appends a self-attention block to this new module. But these operations increase model parameters and cause a little overfitting. Second, Deep Ordinal Regression Network outperforms our RRD in rel [45], probably because it uses an ordinal regression module which takes into account the ordering of discrete depth values. This ordinal regression module release spatial constraints to allow
FIGURE 4. The depth prediction results of ablation models on NYU Depth v2. In each row, the first column is the original RGB image, and the second column is ground truth prediction. The rests are the depth prediction results of ablation models, including RRR, RRD, RDD, DRR, DDR, DDD.
### TABLE 2. Comparison with state of the art methods on Make3D. Errors are computed for depths less than 70m in a central image crop.

| Methods         | Deep network architectures               | Low is better |
|-----------------|------------------------------------------|---------------|
|                 |                                          | rel | rmse | log10 |
| CNN-based       |                                          |     |      |      |
| A. Grigorev et al. [32] (2017) | CNN-LSTM                              | 0.275 | 12.510 | 0.113 |
| A. Roy et al. [25] (2016) | CNN with CRF                           | 0.260 | 12.400 | 0.119 |
| F. Liu et al. [24] (2016) | CNN with CRF                           | 0.287 | 7.360  | 0.109 |
| D. Xu et al. [42] (2017) | CNN with CRF                           | 0.184 | 4.380  | 0.065 |
| Encoder-decoder |                                          |     |      |      |
| T. Zhou et al. [36] (2017) | Encoder-decoder with multi-view pose networks | 0.383 | 10.470 | NA   |
| S. Zhao et al. [40] (2019) | CNN with domain adaptation              | 0.403 | 10.424 | NA   |
| A. A. Abarghouei et al. [41] (2018) | Encoder-decoder with domain adaptation | 0.423 | 9.002  | NA   |
| C. Wang et al. [35] (2018) | CNN with video frames                   | 0.387 | 8.090  | NA   |
| S. Gur et al. [47] (2019) | Encoder-decoder with DenseASPP, self-attention block | 0.246 | 7.671  | NA   |
| C. Godard et al. [37] (2019) | Encoder-decoder with monocular video frames | 0.322 | 7.417  | 0.163 |
| H. Yan et al. [28] (2018) | Encoder-decoder with surface normal module | 0.198 | 4.610  | 0.080 |
| I. Laina et al. [48] (2016) | Encoder-decoder                        | 0.176 | 4.460  | 0.072 |
| M. Heo et al. [23] (2018) | Encoder-decoder using whole strip masking and reliability-based refinement | 0.171 | 4.460  | 0.063 |
| DDD [Ours]      | Encoder-decoder with Residual DenseASPP  | 0.192 | 4.383  | 0.067 |
| H. Fu et al. [45] (2018) | Encoder-decoder with ASPP               | 0.157 | 3.970  | NA   |
| RRD [Ours]      | Encoder-decoder with Residual DenseASPP  | 0.161 | 3.554  | 0.065 |

### TABLE 3. Performance of ablation models on NYU Depth v2.

| Ablation Models | High is better | Low is better |
|-----------------|----------------|---------------|
|                 | $\delta_1$    | $\delta_2$    | $\delta_3$    | rel | rmse | log10 |
| RRR             | 0.716 | 0.945 | 0.990 | 0.182 | 0.627 | 0.075 |
| RRD             | 0.841 | 0.969 | 0.994 | 0.136 | 0.430 | 0.054 |
| RDD             | 0.827 | 0.960 | 0.990 | 0.142 | 0.459 | 0.055 |
| DRR             | 0.825 | 0.963 | 0.993 | 0.140 | 0.479 | 0.055 |
| DDR             | 0.823 | 0.964 | 0.994 | 0.139 | 0.474 | 0.057 |
| DDD             | 0.820 | 0.966 | **0.995** | 0.141 | 0.485 | 0.059 |

2) Complex boundaries like long narrow edges, twisty edges, need complex patterns in a certain reception field. The chair in row (d), the sofa in row (e) has been well estimated by RRD, DDD. Similarly, high vision features are good for complex boundaries.

3) Illuminations in row (f) and row (h) will greatly interfere with local depth prediction, DDR and DRR perform worse than RRD, which indicates that low/mid vision features need be reduced and high vision features can capture illumination in-variance in large reception field.

4) Objects with a big change in depth also need complex patterns of high vision features, such as the wall in row (g), RDD performs worse then RRD, which indicates that precise mid vision features may be helpful in this case.

5) It is interesting in row (a), the sculpture can be extracted well in DRR, because more low vision features are good for this complex boundary, while DRR does not well describe the background depth. On the whole, RDD and DDD may give more accurate prediction than DRR.

6) The problem in row (b) is that wall depth prediction is interfered by window and door in all variants, and RRD can alleviate this interference.

### V. CONCLUSIONS

In this work, we propose a novel Residual DenseASPP Network, in which we define features as low/mid/high vision features, and use two-kinds of skip connection, including feature concentration in a dense block and feature summation in a residual block. We discuss what kind of low/mid/high features is important in Residual DenseASPP by six variants, and one ablation method RRD outperforms all other variants and gives better performance in terms of threshold accuracy than state-of-the-art methods. We notice that low/mid/high vision patterns have different characteristics, which residual block can facilitate training for low/mid vision features learning, and dense block can provide sufficient features for high vision patterns. We show our depth prediction results on NYU Depth v2 and Make3D, and prove that our method can give good prediction in many challenges, including small objects, complex boundaries, illuminations, objects with a big change in depth.
[48] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in Proc. 4th Int. Conf. 3D Vis., Oct. 2016, pp. 239–248.

[49] 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. 3174–3182, Nov. 2018.

[50] Y. Kuznietsov, J. Stuckler, and B. Leibe, “Semi-supervised deep learning for monocular depth map prediction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 2215–2223.

[51] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in Proc. Eur. Conf. Comput. Vis., 2012, pp. 746–760.

[52] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in Proc. Neural Inf. Process. Syst., vol. 2005, pp. 1161–1168.

[53] A. Saxena, M. Sun, and A. Y. Ng, “Make3D: Learning 3D scene structure from a single still image,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 824–840, May 2009.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015.

SHUNRAN ZHANG was born in Tongcheng, Anhui, China, in 1993. He is currently pursuing the master’s degree with the Hefei University of Technology.

His research interests include computer vision, pattern recognition, and machine learning.

KEWEI WU was born in Hefei, Anhui, China, in 1984. He received the B.S. and M.S. degrees in computer science and the Ph.D. degree in computer science from the Hefei University of Technology, in 2009 and 2013, respectively.

He is currently an Associate Research Fellow with the Hefei University of Technology. His research interests include computer vision, pattern recognition, and machine learning.

ZHAO XIE (Member, IEEE) was born in Hefei, Anhui, China, in 1980. He received the Ph.D. degree in computer science from the Hefei University of Technology, in 2007.

He is currently an Associate Research Fellow with the Hefei University of Technology. His research interests include computer vision, pattern recognition, and machine learning.

* * *