A Cascade Dense Connection Fusion Network for Depth Completion

Rizhao Fan[1]
rizhao.fan@unibo.it
Zhigen Li[2]
lizhigen974@pingan.com.cn
Matteo Poggi[1]
m.poggi@unibo.it
Stefano Mattoccia[1]
stefano.mattoccia@unibo.it

[1] University of Bologna
Bologna, Italy
[2] Ping An Technology
Shenzhen, China

Abstract

This paper proposes a lightweight yet effective network architecture for depth completion. It enables to fuse multi-modal and multi-level features through a Cascade Dense Connection Fusion Network. This is implemented by means of a dense connection fusion block, multi-scale features and a modality-aware aggregation mechanism. Our model is evaluated on the KITTI benchmark and achieves competitive results compared with state-of-the-art while counting much fewer parameters.

1 Introduction

Depth completion is an important computer vision task aiming at recovering a dense depth map from a few sparse measurements and an RGB image and it has been widely used in many applications, including 3D object detection, robot navigation, augmented reality and structure-from-motion. Popular strategies to collect depth data rely on active sensors, precisely measuring the distance of objects in the scene by perturbing them through some signals. According to the perturbing technology, different sensors are suited for specific environments. For instance, devices such as Time-of-Flight (ToF) can provide accurate depth information in indoor scenes. At the same time, LiDARs are the most popular sensor for accessing depth information in outdoor environments. However, compared to conventional RGB cameras, they only provide sparse depth information, resulting in many empty regions for which no measurement is available. For instance, the Velodyne HDL-64e LiDAR used in the KITTI dataset provides accurate, yet sparse depth data [39] with a density lower than 6% compared to the image resolution. This fact makes it hard to tackle downstream 3D perception tasks such as detection, semantic segmentation or instance segmentation. Consequently, further processing to recover a dense depth map – i.e. depth completion – becomes pivotal.

With the benefit of a high-resolution color image and deep learning, current methods [23, 29, 40, 48] based on convolutional neural networks (CNNs) have made significant progress in

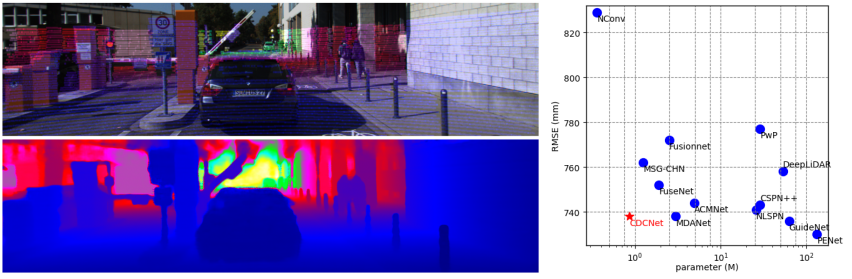


Figure 1: **Cascade Dense Connection Fusion Network in action.** Our model predicts accurate dense depth maps from RGB frame and LiDAR points, using a fraction of the parameters compared to most of the existing methods.

inferring dense depth map from multi-modal data. Nevertheless, most of these existing depth completion methods rely on complex and heavy CNNs, unsuitable for in-vehicle and edge devices. Moreover, these models often use naive aggregation approaches, such as features concatenation or sum, resulting in sub-optimal strategies when fusing multi-modal data.

To tackle these problems, we propose a Cascade Dense Connection fusion network composed of a cascade of Dense Connection Fusion (DCF) blocks. Inspired by [19, 24, 37, 51], we stack our lightweight DCF blocks in a progressive manner instead of building a heavy encoder-decoder network, which allows for saving many parameters. More specifically, the DCF block can learn multi-modal and multi-level features by dense connections and multi-scale learning. We construct a Modality-Aware Aggregation module for learning the multi-modal representations and a Multi-Scale Pyramid Fusion module for learning multi-level features. Figure 1 plots the relationship between parameters and the primary evaluation metric, i.e., RMSE, for the proposed model and state-of-the-art depth completion approaches. We can notice how CDCNet achieves a favorable trade-off compared to existing methods. In summary, the major contributions of this paper can be resumed as follows:

(1) We propose a lightweight Cascade Dense Connection fusion Network (CDCNet) for depth completion, which depends on dense connections to extract and learn depth and RGB features efficiently and effectively.

(2) We design Modality-Aware Aggregation (MAA) and Multi-Scale Pyramid Fusion (MSPF) modules for learning multi-modal and multi-level representations more effectively.

(3) Experimental results show that CDCNet is competitive with state-of-the-art approaches on the KITTI depth completion benchmark while counting much fewer parameters.

2 Related Work

In this section, we review the literature relevant to our work.

Depth Completion. Depth completion aims at recovering dense depth maps from sparse inputs. Early approaches rely only on a sparse depth map as input. Uhrig et al. [39] propose a sparsity-invariant convolution layer, a variant of regular ones, to consider the location of missing data while performing convolutions and address data sparsity within deep networks. Huang et al. [14] extend the principle behind sparsity-invariant convolutions to more operations and propose a hierarchical multi-scale network structure for depth completion. These methods suffer from undesired artifacts, such as ambiguities and mixed-depth values.

More recent works focus on image-guided depth completion. Ma et al. [26, 27] combine sparse depth and an RGB image through early fusion and feed them into an encoder-decoder CNN, which boosts the performance of depth completion. Unlike early fusion of depth and image, recent works [16, 19, 23] share the idea that late fusion can better access multi-modal fusion cues. Zhao et al. [50] utilize graph propagation and symmetric gated fusion strategy to fuse contextual and depth information across different branches. DeepLiDAR [36] introduces pixel-wise surface normals as geometric constraints and proposes multiple branches to generate dense depth maps jointly. Through the years, spatial propagation networks (SPN) [25] became a popular approach for depth estimation. CSPN [8] predicts affinity values of local neighbours and updates pixel values simultaneously by extending SPN. In contrast, NLSPN [30] concentrates on relevant non-local neighbours by learning deformable convolutional kernels to solve the depth completion task.

Feature-level Fusion approaches. Deep learning methods for depth completion usually aggregate depth and image information at the feature level. Lee et al. [18] propose a cross-guidance between image and depth encoder branches and fuse multi-modal features through attention. GuideNet [38] adopts image features as guidance and fuses multi-modal features with skip connections across encoder-decoder networks. FusionNet [40] adopts global branches to guide local branches by concatenating features from different branches.

Multi-level feature fusion also proved effective [21, 43]. Feature pyramid networks [22] utilizes a top-down architecture with lateral connections and fuse multi-scale feature through features sum. UNet++ [51] proposes a nested UNet to learn the importance of features at different layers and adopts a dense skip connection to aggregate multi-scale features. DFANet [20] develops a cross-level feature aggregation strategy to boost accuracy.

Lightweight Dense Prediction. There is practical demand for lightweight networks as more mobile and on-edge devices emerge. MobileNet [10, 11] and ShuffleNet [28], were developed specifically for devices with limited computing power. BiSeNet [46] and BiSeNetV2 [47] are lightweight networks for semantic segmentation, using two-stream paths for modeling low-level details and high level semantic information. PyD-Net [33] and PyD-Net2 [34] are pyramidal architectures for self-supervised monocular depth estimation, deployed on edge devices as well [6, 31, 32]. ICNet [49] uses cascade down-sampled images as input and fuses multi-scale features to pursue efficiency.

Although various approaches have been proposed for depth completion, they share some common, main strategies. Early fusion or late fusion strategies depend on heavy backbones, such as ResNet [9] and transformer [41] to extract multi-modal features. Others depend on spatial propagation network [25] as post-processing to refine depth results. These methods require many parameters and increase the computation efforts during training and inference. Naive aggregation approaches, such as feature concatenation or sum, are utilized to fuse multi-modal and multi-level features, resulting sub-optimal and limiting performance. In contrast, we present a lightweight progressive image-guided fusion strategy for depth completion, consisting of dense connections, modality-aware and multi-scale learning, which help fusing multi-modal and multi-level features effectively.

3 Proposed method

This section describes our proposal for effective and efficient depth completion performed by processing an RGB image and sparse depth data.

We first present our Cascade Dense Connection fusion Network. Then, in subsequent

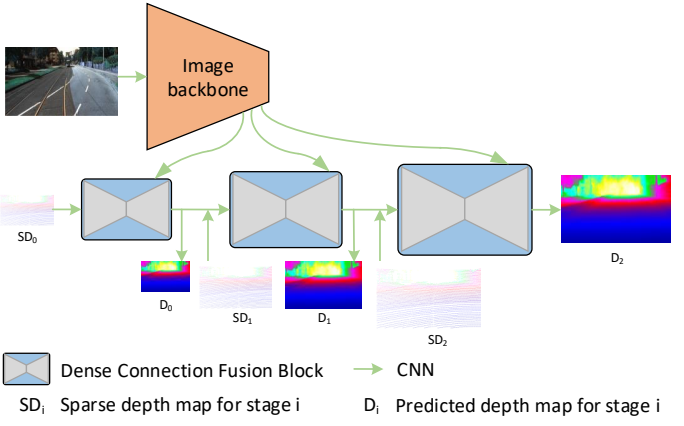


Figure 2: **Pipeline of the proposed method.** The image backbone extracts image features and feeds them to dense connection fusion (DCF) blocks. Three DCF blocks are stacked progressively for three stages. DCF_0, DCF_1, DCF_2, from small to big.

sections, we provide details for the proposed functional modules, i.e., Dense Connection Fusion block, Modality-Aware Aggregation module and Multi-Scale Pyramid Fusion module. The overall architecture of CDCNet is depicted in Figure 2. It takes a color image and a sparse depth map to progressively recover a dense depth map. The image backbone consists of 10 convolutional layers with 3×3 filters. The 3th, 5th, 7th and 9th layers have stride 2 while the others have stride 1. The depth backbones are built using 6 layers defined following the same structure of the first 6 layers of the image backbone. All convolutions are followed by BatchNorm and ReLU functions. One image backbone and one depth backbone have 84K and 46K parameters, respectively. Following [17, 19, 40], we stack lightweight blocks instead of designing a heavy backbone to learn feature representations. The image backbone extracts multi-scale features, providing meaningful information on semantics and texture as guidance to recover depth. We denote the extracted image features as F_I^1, F_I^2, F_I^3, F_I^4, F_I^5, with cumulative strides of 1, 2, 4, 8, 16, respectively. Image features are then fed to the three cascade DCF blocks, namely DCF_0, DCF_1, DCF_2. Apart from image features, quarter-sized sparse map SD_0, half-sized depth SD_1, and full-sized depth map SD_2 are fed to the abovementioned three blocks. Each stage outputs a dense prediction at the same input size, with residual connections integrating the three outputs. Note that all the feature maps in our model have the same number of channels, i.e., $C = 32$, except for multi-scale learning parts. This way, our network is very lightweight. For simplicity, we omit some connection lines about residual connections in Figure 2. More details about how residual connections are integrated into each module can be found in [19].

3.1 Dense Connection Fusion Block

Most depth completion methods use naive concatenation or sum operations to aggregate either heterogeneous depth and image features or homogeneous depth features from different levels. This strategy usually yields sub-optimal results and misleads the fusion process. Recent works [17, 37, 44, 51] show that dense connection and continual fusion are good choices

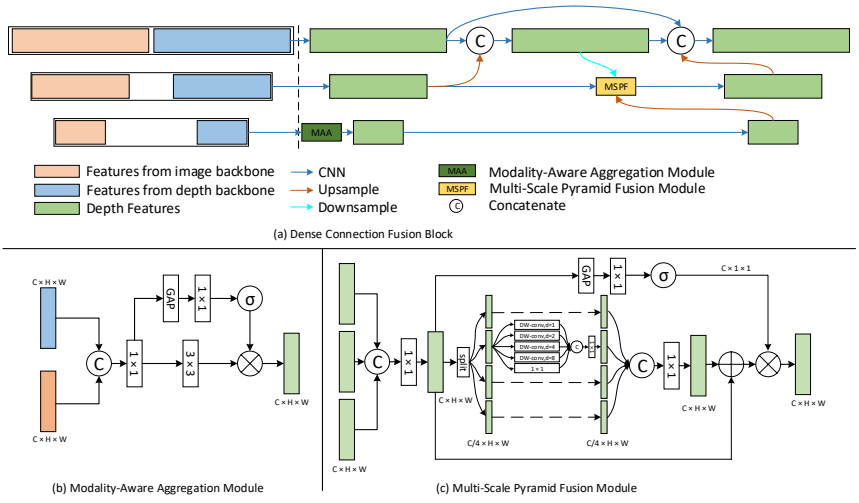


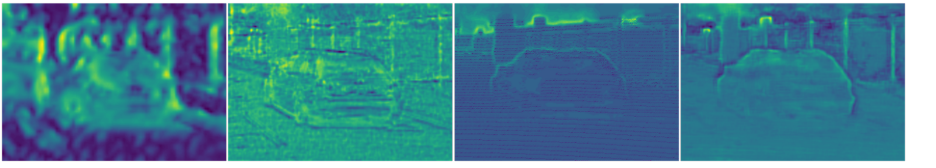
Figure 3: **Illustration of the main modules building CDCNet.** (a) Dense Connection Fusion Block. (b) Modality-Aware Aggregation Module. (c) Multi-Scale Pyramid Fusion Module. Best viewed in color.

to learn representations. Inspired by these works, we design DCF block which fully utilizes dense connection to aggregate multi-modal and multi-level representations, as illustrated in Figure 3 (a). Commonly, deep fusion networks depend on stacking more layers and increasing channel dimensionality to get better results. In contrast, our model adopts a shallow structure and a small number of channels for feature aggregation. Consequently, apart from the standard top-to-down scheme with skip connections, we add an intermediate feature in the aggregation space to compensate for the shallow architecture. The intermediate feature combines features from the same and the higher level at different sizes. Instead of using feature concatenation or sum operation, we also design a modality-aware aggregation module to exploit the discriminative information from the heterogeneous image and depth features, described hereafter.

3.2 Modality-Aware Aggregation Module

Different modalities have different attributes to exploit for feature aggregation; therefore, the critical factor for multi-modal fusion consist of exploiting the valuable information from each of the modalities. Image features contain rich semantic information, yet depth features represent strong distance-perceptive information. Most image and depth feature fusion approaches use concatenation operation, which fails to exploit more information from multiple modalities. Hence, we propose a modality-aware aggregation module (Figure 3 (b)) which aims at enhancing multi-modal representation learning. Concretely, given input image features and depth features $F_I, F_D \in \mathbb{R}^{c\times h\times w}$ in each module, we first concatenate them as $F_{cat} \in \mathbb{R}^{2c\times h\times w}$, then use $conv1\times 1$ to smooth F_{cat} and get $F'_{cat} \in \mathbb{R}^{2c\times h\times w}$, Global Average Pooling, $conv1\times 1$ and $sigmod()$ functions, in order to obtain the modality-aware vector $w \in c\times 1\times 1$ from multi-modal features, which can be formalized as:

$$w = \sigma(conv1\times 1(GAP(conv_{1\times 1}(F_I, F_D)))) \tag{1}$$



(a) high-level **(b) middle-level** **(c) low-level** **(d) output**

Figure 4: **Visualization of multi-level feature maps.** From left to right, (a) high-level, (b) middle-level and (c) low-level feature maps, followed by (d) output of the MSPF module.

For $F_{cat}^{'}$, $conv3\times3$ are used to get $F_{coarse} \in \mathbb{R}^{c\times h\times w}$. The enhanced feature are obtained as:

$$F_M = w \bigotimes conv_{3\times3}(conv_{1\times1}(F_I, F_D)) \tag{2}$$

Then, we get the modality-aware integrated result $F_M \in \mathbb{R}^{c\times h\times w}$.

3.3 Multi-Scale Pyramid Fusion Module

For multi-level features fusion as well, most existing works [3, 17, 19, 26, 30] make use of concatenation or sum operation, which weakens the representation capability of cross-level features. For our task, high-level features have more semantic information while low-level features have more texture information and sparse depth representation, as shown in Figure 4(a,b,c). To aggregate multi-level features effectively, we construct a multi-scale pyramid fusion module embedded into the DCF block, as illustrated in Figure 3(c). Our motivation arises from Atrous Spatial Pyramid Pooling (ASPP) [1] that uses multiple branches to extract multi-scale features. However, ASPP introduces many parameters and high computational overhead. Thus, inspired by [42], we construct MSPF in a lightweight way. Suppose $F_H, F_M, F_L \in \mathbb{R}^{c\times h\times w}$ represent the upsampled high-level feature, the middle-level feature, and the downsampled low-level feature, respectively, which are inputs of the MSPF module. We first apply a 1×1 convolution to perform channel pooling for the concatenated multi-level features and get $F_0 \in \mathbb{R}^{c\times h\times w}$. Then, we equally split F_0 into four feature maps $F_1, F_2, F_3, F_4 \in \mathbb{R}^{c/4\times h\times w}$ along the channel dimension, as

$$\begin{aligned} F_0 &= conv_{1\times1}(F_H, F_M, F_L) \\ F_1, F_2, F_3, F_4 &= Split(F_0) \end{aligned} \tag{3}$$

Then, for each sub-portion F_i of the original feature map F_0, we apply four 3×3 depth-wise separable convolutions with dilation rates of 1, 2, 4, 8 and 1×1 convolution to implement multi-scale learning and get $F_i^1, F_i^2, F_i^3, F_i^4, F_i^5 \in \mathbb{R}^{c/4\times h\times w}, i = 1,2,3,4$, then we use two

consecutive 1×1 convolutions to merge the feature maps $F_i^M \in \mathbb{R}^{c\times h\times w}$ and $F^M \in \mathbb{R}^{c\times h\times w}$:

$$\begin{aligned}
F_i^1 &= conv_{1\times1}(F_i)\\
F_i^2 &= conv_{3\times3}^{d=1}(F_i)\\
F_i^3 &= conv_{3\times3}^{d=2}(F_i)\\
F_i^4 &= conv_{3\times3}^{d=4}(F_i)\\
F_i^5 &= conv_{3\times3}^{d=8}(F_i)\\
F_i^M &= conv_{1\times1}(F_i^1, F_i^2, F_i^3, F_i^4, F_i^5)\\
F^M &= conv_{1\times1}(F_1^M, F_2^M, F_3^M, F_4^M)
\end{aligned} \tag{4}$$

where, $conv_{3\times3}^{d=i}$ denotes 3×3 depth-wise atrous convolution with dilation rate of i.

In the end, we add a residual connection and leverage channel attention [35] to refine the output features, as

$$\begin{aligned}
F^M &= F^M + F_0\\
F^{out} &= F^M \bigotimes \sigma(conv_{1\times1}(GAP(F_0)))
\end{aligned} \tag{5}$$

where $F^{out} \in \mathbb{R}^{c\times h\times w}$ is the final refined feature.

By splitting features and implementing multi-scale learning with depth-wise separable convolutions separately, we dramatically cut down computational complexity and reduce the number of parameters. Moreover, the aggregate results embed semantic and texture information from multi-level features, as shown in Figure 4(d).

3.4 Loss Function

To learn accurate prediction of dense depth maps, we train our network to minimize mean squared error (MSE) and mean absolute error (MAE) losses [19]. A multi-stage and multi-weighted loss function L is the combination of three parts:

$$\begin{aligned}
L =& \omega \sum_{i=1}^{N} (L_2(D_i^2, \hat{D}_i^2) + L_1(D_i^2, \hat{D}_i^2)) +\\
& \omega \sum_{i=1}^{N} (L_2(D_i^1, \hat{D}_i^1) + L_1(D_i^1, \hat{D}_i^1)) +\\
& \sum_{i=1}^{N} (L_2(D_i^0, \hat{D}_i^0) + L_1(D_i^0, \hat{D}_i^0))
\end{aligned} \tag{6}$$

where N represents the set of valid pixels. D^2, D^1, D^0 denote the predicted depth maps from DCF blocks 0, 1 and 2 respectively, and $\hat{D}^2$, $\hat{D}^1$, $\hat{D}^0$ the corresponding semi-dense ground truth maps. Following [19], we set ω to 1 for the first 6 epochs, then decimating it to 0.1 for 11 epochs, and finally disabling it ($\omega = 0$) until the end of the training procedure.

4 Experiment

4.1 Dataset

We evaluate our method and compare it to state-of-the-art solutions on the KITTI depth completion benchmark [8, 39]. KITTI is a popular outdoor dataset providing sparse depth maps captured by Velodyne LiDAR HDL-64e, color images and corresponding semi-dense ground truth. The sparse depth maps provide 5.9% valid depth values on all pixels, while the ground truth maps contain 16% valid depth values over the whole image. The dataset contains 85895 training frames, with 1000 more selected validation frames, and 1000 test data for which ground truth is withheld.

4.2 Evaluation Metrics

We adopt the official evaluation protocol from the KITTI depth completion benchmark [39] to evaluate our network, computing four standard metrics: the mean absolute error (MAE, mm), root mean squared error (RMSE, mm), mean absolute error of the inverse depth (iMAE, 1/km) and root mean squared error of the inverse depth (iRMSE, 1/km). Among them, RMSE is selected to rank all the submitted methods on the KITTI leaderboard.

4.3 Implementation Details

We implement CDCNet using Pytorch and train it with a single NVIDIA RTX 3090 GPU. All the parameters are optimized using Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The learning rate is initialized to 0.001 and multiplied by 0.5 every 5 epochs. A weight decay factor is set to 0.0002. The network is trained for 30 epochs using a batch size of 6 samples. Training images are cropped to a resolution of 1216×352 pixels. The experiments in the ablation study are carried out by training CDCNet on 10000 samples from the training set and by evaluating on the validation split.

4.4 Comparison with state-of-the-art

We compare our model to the state-of-the-art methods published on the KITTI depth completion benchmark. Table 1 shows quantitative results retrieved from the online leaderboard. We report the comparison of our method with others in terms of parameters, accuracy and runtime. The number of parameters and runtime are partially taken, respectively, from [15] and [12]. Our lightweight network, CDCNet, outperforms most previous methods under the primary evaluation metric RMSE and achieves results comparable with those by state-of-the-art models. In particular, CDCNet achieves accuracy close to GuideNet [38], CSPN++ [4], NLSPN [30], MDANet [17], with respectively 1.3%, 3.0%, 3.4% and 28% of their total parameters. Compared with MSG-CHN [19] which inspires our method, CDCNet gets better results by using only 69% of its parameters. Due to the diversity of hardware platforms used by each method, performing a fair comparison for what concerns runtime is not trivial. Nevertheless, these results still suggests that our method is faster than most state-of-the-art methods. SPN-based methods [3, 4, 30] slow down their inference time because of the iterative spatial propagation step. PwP [45], DeepLiDAR [36], GuideNet [38] and PENet [13] adopt multi-branch, heavy backbones – i.e., ResNet – which are time-consuming. In contrast, CDCNet takes shorter inference time, despite it processes data in a stacked manner.

Methods	Parameters (M)	RMSE (mm)	MAE (mm)	iRMSE (1/km)	iMAE (1/km)	runtime (s)	Platform
Sparse-to-Dense [27]	-	814.73	249.95	2.80	1.21	0.08	Tesla V100
PwP [45]	29.10	777.05	235.17	2.42	1.13	0.10	Tesla V100
FusionNet [40]	2.50	772.87	215.02	2.19	0.93	0.02	RTX 2080Ti
FuseNet [6]	1.90	752.88	221.19	2.34	1.14	0.09	-
NConv [7]	0.36	829.98	233.26	2.60	1.03	0.02	Tesla V100
DeepLiDAR [36]	53.40	758.38	226.50	2.56	1.15	0.35	RTX 2080Ti
CSPN [3]	-	1019.64	279.46	2.93	1.15	1.00	Titan X
CSPN++ [4]	28.80	743.69	209.28	2.07	0.90	0.20	Tesla P40
NLSPN [30]	25.80	741.68	199.59	1.99	0.84	0.13	RTX 2080Ti
PENet [13]	133.70	730.08	210.55	2.17	0.94	0.16	RTX 2080Ti
GuideNet [38]	63.30	736.24	218.83	2.25	0.99	0.14	GTX 1080Ti
ACMNet [50]	4.90	744.91	206.09	2.08	0.90	0.35	RTX 2080Ti
MDANet [17]	3.07	738.23	214.99	2.12	0.99	0.03	Tesla P100
MSG-CHN [19]	1.25	762.19	220.41	2.30	0.98	0.01	RTX 3090
CDCNet (ours)	0.87	738.26	216.05	2.18	0.99	0.03	RTX 3090

Table 1: **Quantitative results on the KITTI test set.** We report the amount of parameters, standard evaluation metrics and runtime for state-of-the-art models and CDCNet.

Component	RMSE (mm)	Parameters (K)
Sum	879.81	672
Concat	874.01	727
Gated	882.26	699
MAA	870.39	730

Component	RMSE (mm)	Parameters (K)	Memory (GB)	Speed (ms)
Sum	876.46	616	3.499	18.063
Concat	874.01	727	3.600	18.673
ASPP	870.56	1178	4.286	23.923
MSPF	863.65	695	4.007	27.830

Table 2: **Ablation study on MAA (left) and MSPF (right) modules.** Sum denotes feature sum operation; Concat denotes feature concatenation operation; Gated denotes Gated Fusion. Parameters, Memory and Speed refers to the entire network processing.

For what concerns the main competitor inspiring our work, i.e. MSG-CHN [19], for a fair comparison we use the authors' code and measure its runtime on the same hardware platform used by CDCNet. Our model runs in 0.03 seconds, slower than MSG-CHN [19], because of the feature aggregation modules we introduced. However, this is compensated with higher accuracy and fewer parameters. In summary, CDCNet gets competitive results with clearly fewer parameters on the KITTI depth completion benchmark.

Figure 5 reports a qualitative comparison between results yielded by state-of-the-art methods and ours, with the latter being in the last row. Our dense connection fusion strategy, which can efficiently exploit high-level semantic and low-level context information, yields accurate depth maps, preserves finer details on complex structure boundaries and recovers more accurate contours for thin structures in faraway scenes.

4.5 Ablation Study

In this section, we demonstrate the effectiveness of the components proposed in this paper.

Impact of Modality-Aware Aggregation Module. By surveying most of the works for KITTI depth completion, concatenation and sum emerge as the dominant image and depth fusion strategies [3, 4, 13, 17, 19, 26, 27, 29, 30, 36, 40]. Besides the methods on the KITTI benchmark, [5] proposed a gated fusion for dense image and depth feature fusion. To measure the impact of the MAA module on the final accuracy, we replace it with the alternatives mentioned above in these experiments. It is worth noting that the remaining components are kept unchanged; only the fusion module on which the comparison focuses

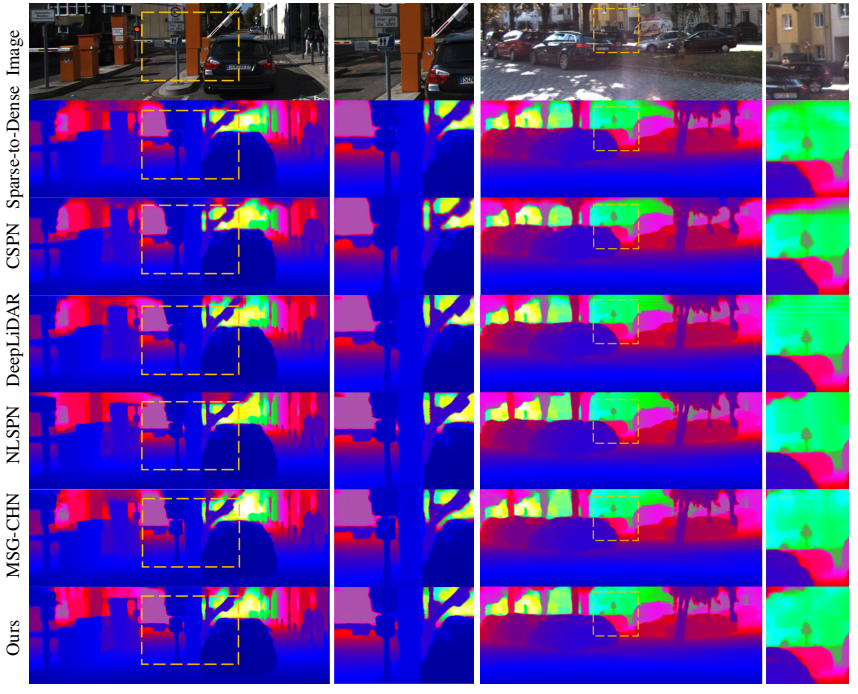


Figure 5: **Qualitative comparison with state-of-the-art methods.** From top to bottom: RGB image, results of Spare-to-Dense[27], CSPN[3], DeepLiDAR[36], NLSPN[30], MSG-CHN[19], and Ours, respectively.We zoom-in the yellow dotted regions at the right.

on changes. From Table 2, on the left, we can observe that our fusion strategy yields better results compared to alternative methods, with a limited increase in the number of parameters. Gated fusion results are the worst for this task since this strategy is designed for dense feature fusion, whereas depth features are usually very sparse in completion task.

Impact of Multi-Scale Pyramid Fusion Module. To validate the effectiveness of MSPF, we compare the performance achieved by CDCNet when using it or when it is replaced by sum, concatenation or ASPP alternatives. As shown in Table 2, on the right, the results demonstrate that the lightweight MSPF module can achieve the best performance. However, this improvement comes at the expense of speed. Yet, when directly compared with ASPP, MSPF introduces fewer parameters and requires fewer GPU memory thanks to feature splitting and depth-wise convolutions.

5 Conclusion

In this paper, we have proposed a lightweight yet effective cascade dense connection fusion network, CDCNet. By stacking the dense connection fusion blocks, image and depth features are aggregated effectively in a progressive manner. We employ a modality-aware aggregation method to enhance the fusion of image and depth features. Then, a lightweight multi-scale learning module boosts multi-level feature fusion. We evaluate CDCNet on the KITTI depth completion dataset achieving competitive results compared to state-of-the-art methods, yet using much fewer parameters.

References

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[2] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10023–10032, 2019.

[3] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018.

[4] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10615–10622, 2020.

[5] Yanhua Cheng, Rui Cai, Zhiwei Li, Xin Zhao, and Kaiqi Huang. Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3029–3037, 2017.

[6] Antonio Cipolletta, Valentino Peluso, Andrea Calimera, Matteo Poggi, Fabio Tosi, Filippo Aleotti, and Stefano Mattoccia. Energy-quality scalable monocular depth estimation on low-power cpus. *IEEE IoT Journal*, 2021.

[7] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. *arXiv preprint arXiv:1805.11913*, 2018.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[12] Mu Hu. Penet. https://github.com/JUGGHM/PENet_ICRA2021, 2021.

[13] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13656–13662. IEEE, 2021.

[14] Zixuan Huang, Junming Fan, Shenggan Cheng, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, 29:3429–3441, 2019.

[15] Lam Huynh, Phong Nguyen, Jiří Matas, Esa Rahtu, and Janne Heikkilä. Boosting monocular depth estimation with lightweight 3d point fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12767–12776, 2021.

[16] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018.

[17] Yanjie Ke, Kun Li, Wei Yang, Zhenbo Xu, Dayang Hao, Liusheng Huang, and Gang Wang. Mdanet: Multi-modal deep aggregation network for depth completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4288–4294. IEEE, 2021.

[18] Sihaeng Lee, Janghyeon Lee, Doyeon Kim, and Junmo Kim. Deep architecture with cross guidance between single image and sparse lidar data for depth completion. *IEEE Access*, 8:79801–79810, 2020.

[19] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. A multi-scale guided cascade hourglass network for depth completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 32–40, 2020.

[20] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9522–9531, 2019.

[21] Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11418–11425, 2020.

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[23] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2136–2144, 2021.

[24] Peng Liu, Zonghua Zhang, Zhaozong Meng, Nan Gao, and Chao Wang. Pdr-net: Progressive depth reconstruction network for color guided depth map super-resolution. *Neurocomputing*, 479:75–88, 2022.

[25] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[26] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018.

[27] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019.

[28] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[29] Danish Nazir, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Semattnet: Towards attention-based semantic aware guided depth completion. *arXiv preprint arXiv:2204.13635*, 2022.

[30] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision*, pages 120–136. Springer, 2020.

[31] Valentino Peluso, Antonio Cipolletta, Andrea Calimera, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Enabling energy-efficient unsupervised monocular depth estimation on armv7-based platforms. In *Design, Automation & Test in Europe Conference & Exhibition*, pages 1703–1708, 2019.

[32] Valentino Peluso, Antonio Cipolletta, Andrea Calimera, Matteo Poggi, Fabio Tosi, Filippo Aleotti, and Stefano Mattoccia. Monocular depth perception on microcontrollers for edge applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[33] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5848–5854. IEEE, 2018.

[34] Matteo Poggi, Fabio Tosi, Filippo Aleotti, and Stefano Mattoccia. Real-time self-supervised monocular depth estimation without gpu. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2022. doi: 10.1109/TITS.2022.3157265.

[35] Wang Qilong, Wu Banggu, Zhu Pengfei, Li Peihua, Zuo Wangmeng, and Hu Qinghua. Eca-net: Efficient channel attention for deep convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[36] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019.

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[38] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30: 1116–1129, 2020.

[39] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

[40] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th international conference on machine vision applications (MVA)*, pages 1–6. IEEE, 2019.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[42] Yu-Huan Wu, Yun Liu, Le Zhang, Ming-Ming Cheng, and Bo Ren. Edn: Salient object detection via extremely-downsampled network. *IEEE Transactions on Image Processing*, 31:3125–3136, 2022.

[43] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[44] Yongjian Xin, Shuhui Wang, Liang Li, Weigang Zhang, and Qingming Huang. Reverse densely connected feature pyramid network for object detection. In *Asian Conference on Computer Vision*, pages 530–545. Springer, 2018.

[45] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2811–2820, 2019.

[46] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[47] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021.

[48] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018.

[49] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018.

[50] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30:5264–5276, 2021.

[51] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.