DeepDNet: Deep Dense Network for Depth Completion Task

Girish Hegde[1], Tushar Pharale[1], Soumya Jahagirdar[1], Vaishakh Nargund[1],
Ramesh Ashok Tabib[1], Uma Mudenagudi[1], Basavaraja Vandrotti[2], Ankit Dhiman [2]
[1]Center of Excellence in Vision Intelligence (CEVI),
KLE Technology University (KLETech), Hubballi, Karnataka, INDIA
[2] Samsung Research Institute Bangalore (SRIB), INDIA

Abstract

In this paper, we propose a Deep Dense Network for Depth Completion Task (DeepDNet) towards generating dense depth map using sparse depth and captured view. Wide variety of scene understanding applications such as 3D reconstruction, mixed reality, robotics demand accurate and dense depth maps. Existing depth sensors capture accurate and reliable sparse depth and find challenges in acquiring dense depth maps. Towards this we plan to utilise the accurate sparse depth as input with RGB image to generate dense depth. We model the transformation of random sparse input to grid-based sparse input using Quad-tree decomposition. We propose Dense-Residual-Skip (DRS) Autoencoder along with an attention towards edge preservation using Gradient Aware Mean Squared Error (GAMSE) Loss. We demonstrate our results on the NYUv2 dataset and compare it with other state of the art methods. We also show our results on sparse depth captured by ARCore depth API with its dense depth map. Extensive experiments suggest consistent improvements over existing methods.

1. Introduction

Depth estimation from 2D images is a fundamental task in many applications including scene understanding and reconstruction [39, 16]. Depth estimation plays a vital role in a wide range of application based technologies, such as autonomous driving, augmented reality (AR) [39], robotics and 3D mapping. A set of objectives can be resolved considering depth such as, producing convincing occlusions and consistent depth prediction across frames to avoid flickering. Existing depth sensors including LiDAR and structured light-based depth sensors provide only sparse depth measurements, resulting in degradation in the performance for many depth-dependent tasks. To address such degradation, densification of sparse depth is need of the hour.

Learning based methods, especially convolutional neural networks (CNNs), perform well in many computer vision tasks such as object detection [34], image classification [25], semantic segmentation [24], and so on. Deep learning performs commendable job in computer vision tasks, works such as [37, 10, 21, 8] use deep networks for depth densification. One of the main reasons for processing of sparse data is to complete missing information. Intuitively appropriate designs of CNN architectures can aid in the prediction of dense depth using valid input data.

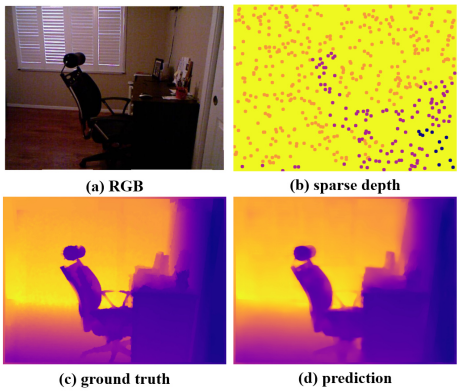

Figure 1. Depth densification using proposed DeepDNet.

While the performance of these methods has been steadily increasing, there are still major problems in the qualities of these estimated depth maps. Based on the analysis of existing architecture and training strategies we set out with the design goal to develop a deeper architecture that makes feature reuse, feature propagation and depth completion task much smoother. Since the depth sensors can capture largely random sparse depth, a system capable of accepting this random sparse as input is necessary. We employ a simple and efficient preprocessing module, as this depth completion task is carried out using CNNs [26] which requires uniform sparse depth aids the neural network to achieve better results. We also need a deeper network which can estimate depth for all the points in the scene using sparse depth. Towards this, we introduce a novel UNet based Dense-Residual-Skip (DRS) Autoencoder [35].

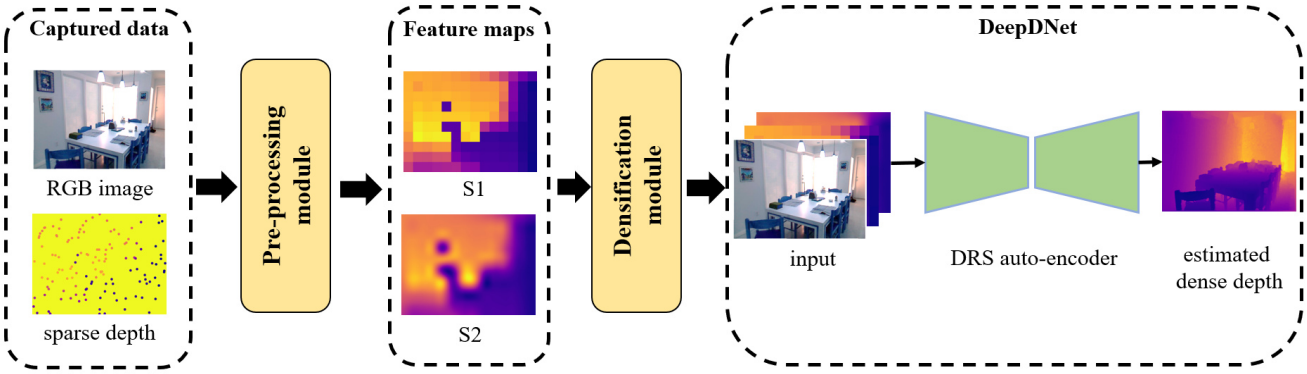

Figure 2. DeepDNet: Deep dense network for depth completion

Based on the experiments conducted, substantial error in the depth completion task is due to inaccurate depth estimation at the edges. Towards this, we provide attention mechanism by introducing gradient term in the loss function which is mathematically well-known solution [22]. We incorporate this mathematical solution as a learning based change making the model "restoration edge aware". Our method produces improvised dense depth and outperforms in comparison with state-of-art methods. In addition, our method produces better quality dense depth maps compared to ARCore dense depth map API.

Contributions of this paper. The major contributions of this paper are presented in three folds:

1. We introduce efficient preprocessing module in which we employ Quadtree based decomposition for transformation from random sparse input to a grid based uniform input.

2. We design a novel deep learning architecture called Dense Residual Skip Autoencoder with an attention mechanism that preserves the edge information.

3. We formulate a novel loss function called Gradient Aware Mean Squared Error Loss (GAMSE) based on existing gradient aware functions towards depth completion task that preserves edge information in generated dense depth map.

4. We demonstrate our results on NYUv2 dataset. We also demonstrate our results on Depth obtained from ARCore depth API.

In Section 2 we present related works on depth densification. In Section 3 we present our methodology for depth densification. In Section 4 we provide results on the proposed methodology. In Section 5 we conclude our contributions with future scope in the depth estimation.

2. Related Work

Depth densification encompass depth prediction and depth completion as its sub-problems. Towards this we provide a brief level of insight on available methods.

RGB-based depth prediction. Early works on depth prediction using RGB images largely relied on hand-crafted features and probabilistic graph models. In [36] authors estimate the absolute scales of different image patches and inferred the depth image using a Markov Random field model. From a single monocular image, the task of depth densification was addressed before the rise of deep learning in the work presented in [30]. Traditional multi-view stereo algorithms such as Patch-based Multi-view Stereo (PMVS) and Clustering Views for Multi-view Stereo (CMVS) are methods are used for dense depth estimation for stereo and multiple images [12].

Advancements in deep learning has paved the way in successful implementation to the depth estimation problem. Authors in [6] [19] have shown that Encoder-decoder based deep neural network predict improved resolutions considering individual pixels for depth. Authors in [9] present two-stack convolutional neural network (CNN), one predicting the global coarse scale and the other refining the local details. Authors in [37] show, traditional neural networks under perform on sparse data with clues on location of missing depth.

Unsupervised based depth prediction. Authors in [13] propose an unsupervised framework to learn a deep convolutional neural network for a single view depth prediction, without considering ground truth depth. Authors in [27] provide an approach to depth map prediction from monocular images that learns in a semi-supervised way. They enforce deep network to produce photo-consistent dense depth maps using direct image alignment loss. Authors in [8] propose to use single deep regression network to learn from the RGB-D raw data, and show impact of number of depth samples on prediction accuracy. The increase in the pre-

diction accuracy due to the introduction of sparse depth is appreciated.

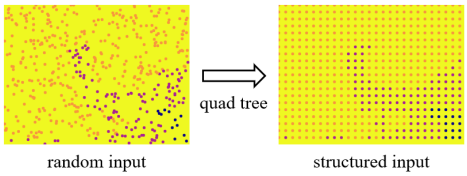

Figure 3. Preprocessing module: conversion of random sparse to grid input

Depth completion. Depth completion is a sub-problem of depth estimation, the task aims to recover dense depth from sparse depth measurements. Authors in [7] solve solve the problem of depth completion from RGBD data by jointly extracting 2D and 3D features considering a dedicated neural network block. Authors in [33] provide end-to-end non-local spatial propagation network for depth completion considering relevant non-local neighbours during propagation.

Exploiting the usage of depth as an input is common in many computer vision tasks such as tracking and segmentation [8] and SLAM based system modules [40] proposes a deep model which fuses complementary information derived from multiple CNN side outputs.

3. DeepDNet for depth densification

In Section 3.1 we present the pre-processing block that introduces the conversion of acquired input RGB and sparse depth to the desired feature maps. In Section 3.2 we introduce DRS autoencoder. In Section 3.3 we provide the advancement on MSE by introducing new gradient aware error term.

3.1. Preprocessing Module.

In this section we present the proposed preprocessing module for depth densification task. Majority of the data captured from depth sensors (Mobile, Camera, etc) is sparse and unstructured, however convolutional neural networks [28] are considered to learn better features with uniform data. A module that transforms unstructured random sparse input to uniform grid-based sparse input is important.

Towards this, we introduce quad tree based preprocessing module, as shown in the Fig. 2 for transforming random sparse points to uniform sparse input. This conversion from random sparse depth to uniform sparse depth is given in the Algorithm 1. The unmediated application of dense convolution on the generated grid-based sparse input results in wastage of computing resources. To overcome this difficulty, we extract two features and feed the same to the autoencoder. These two features are

- F1: Nearest neighbour interpolation of sparse depth

Algorithm 1: Random sparse depth to grid sparse depth conversion

```
Input: sparse depth S(x, y);
Initialize: H, W = shape(S);
Initialize: qt = quadTree();
Initialize: grid(x, y);
Initialize: offset = 3;
for (x, y) such that S(x, y) != 0: do
    Insert point(x, y) to quadTree qt;
end
for all grid points (x, y) do
    nx, ny = query qt for nearest neighbour point;
    grid[x, y] = S[nx, ny];
end
```

- F2: Bi-cubic interpolation of sparse depth

All the points with unknown depth can be assigned to a group, by observing the nearest neighbour with known depth. As there is high correlation between the points with unknown depth to its neighbours, we introduce nearest neighbour interpolation (F1) and bicubic interpolation (F2) as two feature maps to the autoencoder along with the RGB image. Bicubic interpolation is included as the depth of any point is dependent on its neighbours in all directions and as it utilizes the weighted average of four translated pixel values for each output pixel value. Extraction of two feature maps is explained in the Algorithm 2.

Algorithm 2: Feature extraction from grid sparse depth for autoencoder

```
Input: grid sparse depth grid(x, y);
Input: RGB image rgb(x, y);
Initialize: F1(x, y) = 0, for all (x, y);
Initialize: F2(x, y) = 0, for all (x, y);
Initialize: k = downsampling window size;
Initialize: h = H // k;
Initialize: w = W // k;
Initialize: down(x, y) = 0 for all x, y;
for y in (0, h) do
    for x in (0, w) do
        down[x, y] = grid[x*k, y*k];
    end
end
F1 = nearestNeighbourInterpolation(down, (W, H));
F2 = bi-cubicInterpolation(down, (W, H));
Features = Concatenation(rgb, F1, F2);
```

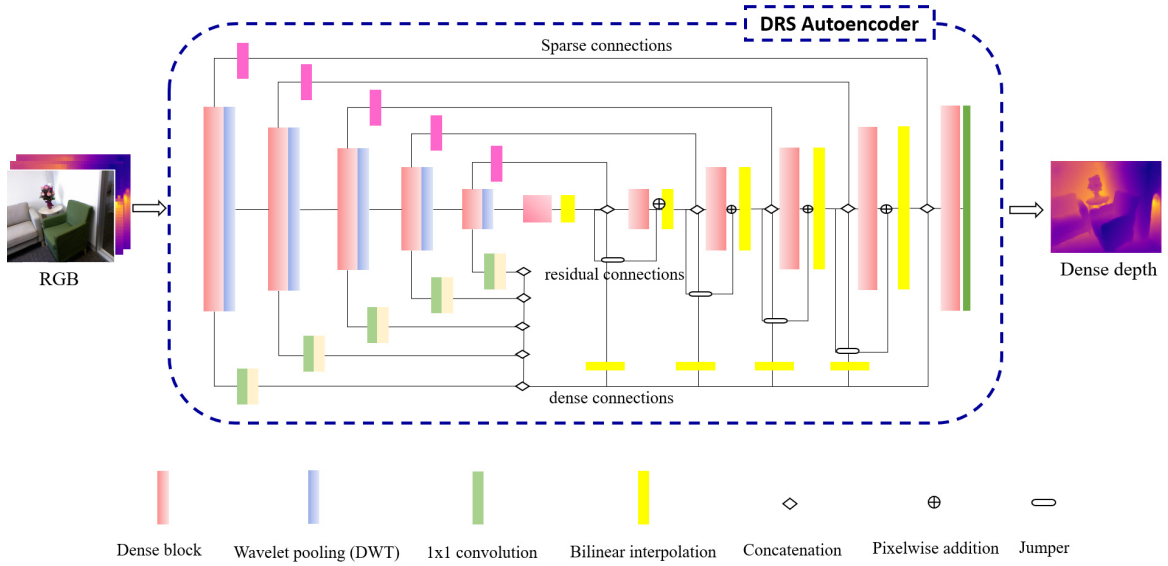

Figure 4. Proposed DRS auto-encoder: novel deep learning auto-encoder architecture

3.2. DRS autoencoder

In the previous Section, we discuss essential preprocessing module. In this Section, we introduce proposed DRS auto-encoder. Fig. 4 shows an overview of our DRS autoencoder network for depth densification. An ideal deep network produces accurate dense depth maps using sparse input. To produce these dense depth maps, the deepness of the network should be extensive. Towards this, works such as [17] introduce dense blocks where each layer is connected to every other layer in a feed-forward fashion. This alleviates the vanishing-gradient problem, strengthens feature propagation, encourages feature reuse, and substantially reduces the number of parameters. We employ these dense block as building blocks of our network. We introduce two types of skip connections to solve the problem of vanishing gradient problem, as shown in the Fig. 4.

In addition to solve vanishing gradient problem, these two skip connections aid the model to have foremost feature reuse which reduces deepness of the network. These two skip connections are explained in details in Section 3.2.1 and 3.2.2.

3.2.1 Dense skip connectivity

Depth completion task requires very deep network and introducing such network will introduce problems such as vanishing-gradient. DenseNet blocks and long skip connections, address this issue of vanishing gradient to some extent. However, vanishing gradient problem still remains a major concern to be solved. We address this issue by introducing new kind of connectivity between encoder and decoder called as Sparse connectivity. We also introduce a new type of direct connection with appropriate interpolation and channel reduction using 1x1 convolution from all encoder layers to every decoder layer. Introducing these dense skip connections ensures more feature re-usability and recovers the spatial information. Information is lost during down-sampling in encoder, and introducing these new types of skip connections helps the model to recover from this loss. These skip connections strengthens feature propagation, feature reuse and substantially reduce the number of parameters and employ faster convergence.

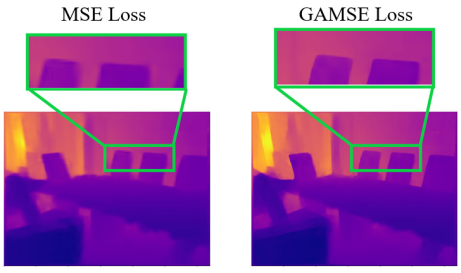

Figure 5. Comparison of MSE and GAMSE loss. Results of depth densification using DRS auto-encoder trained using MSE and GAMSE loss for input image shown in Fig. 6.

3.2.2 Residual connectivity

We introduce residual connectivity [18] as they preserve the gradients as they back propagate through identity function by doing vector addition of features. Back propagation with identity function makes the usage of residual blocks in tasks

Table 1. **DRS performance on NYU-V2 dataset:** The reported numbers are from the corresponding original papers. Note that lower values for RMSE and MRE are considered superior and higher values for $\delta 1(\%)$, $\delta 2(\%)$ and $\delta 3(\%)$ are considered superior. It is also to be noted that our method performs better in majority of the cases.

Model	% of points Sampled	Downsampling factor	RMSE	MRE	δ1(%)	δ2(%)	δ3(%)
D3	0.011	96X96	0.318	7.20	94.2	98.9	**99.8**
DRS(ours)	0.011	96X96	**0.309**	**6.98**	**94.8**	**99.04**	99.78
Ma et al.	0.029	~59X59	0.351	7.8	92.8	98.4	99.6
D3	0.043	48X48	**0.193**	3.21	98.31	**99.73**	**99.95**
DRS(ours)	0.043	48X48	0.196	**3.16**	**98.41**	99.71	99.94
Lu et al.	-	24X24	0.171	-	-	-	-
D3	0.174	24X24	0.118	**1.49**	99.45	99.92	99.98
Fu et al.	-	24X24	0.509	-	82.8	96.5	99.2
High quality Monocular Depth estimation.	-	24X24	0.390	-	89.5	98.0	99.6
DRS(ours)	0.174	24X24	**0.117**	1.56	**99.48**	**99.92**	**99.98**
Ma et al.	0.289	~19X19	0.23	4.4	97.1	99.4	99.8
Lu et al.	-	16X16	0.108	-	-	-	-
D3	0.391	16X16	0.087	**0.99**	**99.72**	99.97	99.99
DRS(ours)	0.391	16X16	**0.086**	1.03	99.74	**99.97**	**99.99**

such as optical flow estimation and semantic segmentation. We introduce residual connections in decoder in which feature maps from each decoder layer is added to its subsequent layer. Residual connections makes layers of decoder to learn from previously captured information of layers of the decoder.

Neighbourhood pooling techniques such as max pooling, average pooling [41] are simple and efficient. But these methods introduce edges halos, blurriness and aliasing. To address this,e we implement wavelet pooling [38]. We replace the max-pooling layers by a wavelet pooling layers that performs second level wavelet decomposition.

We apply bilinear interpolation in decoder as no learnable parameters are necessary. We use Random Relu (RRelu) [4] as activate function to our architecture. All convolutional layers in the network are batch normalized [20].

3.3. GAMSELoss

In this Section, we introduce Gradient Aware Mean Squared Error (GAMSE) loss function for the task of depth completion and present comparison of predictions made by models trained on MSE and GAMSE separately in Fig. 5. Mean Squared Error loss [2] calculates the loss for the entire image without having to cornerstone any specific features of the image. Based on the observations and extensive experiments, reconstruction problems have poor performance due to inaccurate depth prediction at edges. Inspired from works such as [29, 15, 31], and to give importance to edges, we propose new GAMSE loss. This new loss function helps the model with fast convergence, enhanced stability and edge preservation.

$$\mathbf{G} = \mathbf{E}\left[\left(\left(\frac{\partial p}{\partial x}\right) - \left(\frac{\partial g}{\partial x}\right)\right)^2 + \left(\left(\frac{\partial p}{\partial y}\right) - \left(\frac{\partial g}{\partial y}\right)\right)^2\right] \tag{1}$$

$$\mathbf{M} = \mathbf{E}\left[(p-g)^2\right] \tag{2}$$

$$\mathbf{LOSS} = \gamma\mathbf{G} + (1-\gamma)\mathbf{M} \tag{3}$$

where **p** is the prediction, **g** is ground truth or target, **E** is expectation, γ is hyperparameter $\gamma \in [0, 1]$

In equation (1) $\left(\frac{\partial p}{\partial x}\right) - \left(\frac{\partial g}{\partial x}\right)$ represents the difference of gradient of predicted depth and ground truth depth in horizontal direction and $\left(\frac{\partial p}{\partial y}\right) - \left(\frac{\partial g}{\partial y}\right)$ represents the difference of gradient of predicted depth and ground truth depth in vertical direction. And equation (2) represents Mean Squared Error between predicted depth and ground truth. The convolution operation of image with sobel [22] filter is performed while computing the gradients of the image considered.

The edges in an image are represented by set pixels which have higher gradient than nearby pixels. In the proposed loss function, the term **G** represents edge regions. It is determined by calculating the difference between ground truth and prediction, only at edges [22]. This enhances the propagation of gradients to kernels which are responsible for edge production. The G effects MSE loss [2] as edge regions are a part of MSE [2]. It ultimately results in fast convergence, enhanced stability and edge preservation. The hyperparameter gamma plays a crucial role in assigning the weightage to gradient regions in the image. Fig. 3.3. shows the comparison between predictions made by model trained on MSE [2] and GAMSE Loss function. As observed in

5 depth prediction at edges due to introduction of the proposed loss function are more accurate compared to MSE [2].

4. Results and Discussions

In Section 4.1 we provide the description of the dataset used. In Section 4.2 we provide implementations and training details. Section 4.3 present metrics used to evaluate performance of the model. In Section 4.4 we provide qualitative results of proposed architecture. In Section 4.5 we provide ablation study on various experiments performed.

4.1. Dataset Description

Our framework uses the NYU-Depth-v2 dataset [32] which consists of RGB and depth images collected from 464 different indoor scenes with a Microsoft Kinect. We use the official split of data, where 249 scenes are used for training and 215 for testing. The small labelled test dataset with 654 images is used for evaluating the final performance. For training, we sample spatially evenly from each raw video sequence from the training dataset, generating roughly 48,000 synchronized depth-RGB image pairs.

4.2. Implementation Details

We use batch size of 12 on 4 TeslaV100 GPUs on DGX-1 for training. The model is implemented in PyTorch 1.3.0 [3]. We train the model with a 24x24 downsampling rate for 15 epochs i.e. roughly 60000 iterations. We perform experiments with model trained on different downsampling rates. The model trained on 24x24 downsampling rate model is loaded and again trained for about 5 epochs i.e. roughly 20000 iterations. Initial learning rate of 0.01 is used and for every 4000 iterations learning rate is decayed by a factor of 10. Adam [23] is used as an optimizer and our custom GAMSELoss criterion is used.

4.3. Evaluation Metrics

Standard metrics are used to evaluate our depth estimation model against valid ground truth depth values. Let ŷ be the predicted depth and y the ground truth dense depth. We measure: (1) Root Mean Square Error (RMSE)[2]:

$$\sqrt{\frac{1}{N}\sum[\hat{y}-y]^2} \tag{4}$$

(2) Mean Absolute Relative Error (MRE)[1]:

$$\frac{100}{N}\sum\left(\frac{|\hat{y}-y|}{y}\right) \tag{5}$$

(3) Delta Thresholds (δ i):

$$\frac{\left|\left\{\hat{y}|max\left(\frac{y}{\hat{y}},\frac{\hat{y}}{y}\right)<1.25^i\right\}\right|}{\{\hat{y}\}} \tag{6}$$

, δ i is the percentage of pixels with relative error under a threshold controlled by the constant i.

4.4. Qualitative Analysis.

In this Section, we present analysis of DRS model with GAMSE loss function for NYUv2 dataset. We further demonstrate the results on multiple sparse input patterns. For downsampling factor of A x A, we take H*W / A^2 depth values as the sparse input. For example, for 24x24 downsampling on a 480x640 image, this would be 0.18 % of the total pixels.

In Fig. 6 we show the results on NYUv2 dataset with a downsampling factor of 48x48 in the first row, 12x12 in the second row, along with grid based sparse depth as input.

In Fig. 9 we show the results on NYUv2 dataset different rates of downsampling factors. We can observe that the presented network performs fairly better even with extreme sparse depth information as an input. With the increase in number points in sparse depth quality of dense depth map increases.

In Fig. 10 we show the results for dense map produced using the proposed network and Google ARCore depth API. The downsampling factor for the sparse input is 96x96. It can be observed that our method generates clearer, accurate and dense depth maps with higher quality compared to dense depth maps produced by Google ARCore depth API.

In Fig. 11 we show a gallery of depth estimation results that are predicated using our method along with a comparison to those generated by the two methods [11] and [5]. It is observed by the black bounding boxes for same regions for the scenes, that our approach produces depth estimations at higher quality where depth edges better match those of the ground truth and with significantly fewer artifacts.

4.5. Ablation Study

We provide ablation study for depth completion task in two different experimental conditions i.e

1. Dense depth map prediction with random sparse depth as input versus uniform grid based sparse depth as input, and

2. Dense depth map prediction without the introduced Gradient Aware MSE loss function versus Gradient Awarse MSE loss function.

In Fig. 7, the first column contains RGB image from NYUv2 dataset, second column contains predicted dense depth map from proposed DRS autoencoder with random sparse depth as input, third column contains dense depth map prediction from grid based sparse input and the fourth column contains the ground truth dense depth map. It can be observed that architecture provides better quality dense depth maps when the input is of uniform type. All the previous depth completion methods considered only uniform

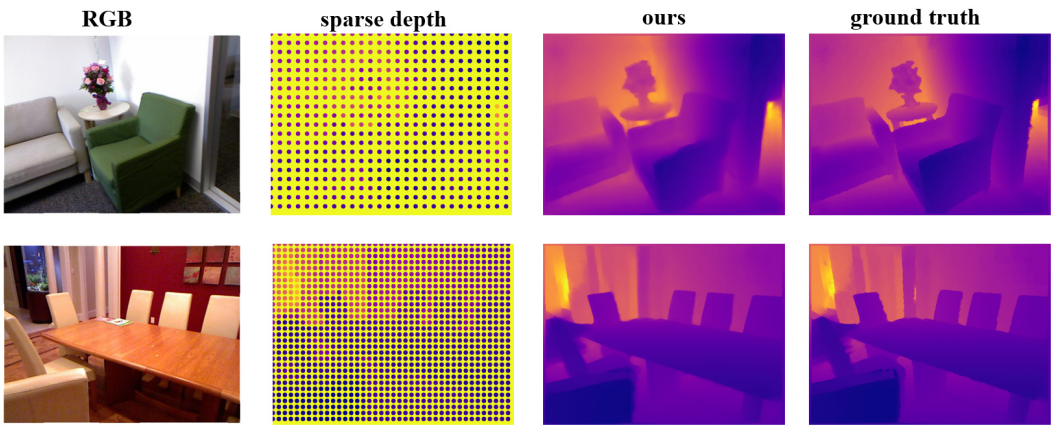

Figure 6. Visualisation of our results on NYUv2 dataset The first column consists of RGB image, second column consists of sparse depth. The sparse depth is obtained using a downsampling factor of 48x48, 24x24, 12x12.

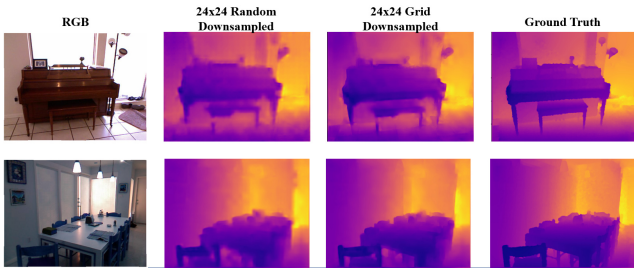

Figure 7. Results on DRS with unstructured random sparse depth as input and uniform grid sparse depth as input: Sparse depth input downsampling factor: 24x24.

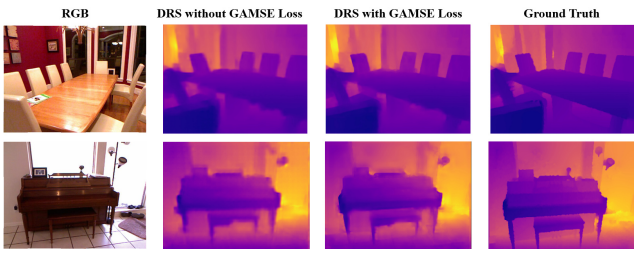

Figure 8. Results on DRS without GAMSE loss function and with GAMSE loss function: sparse depth input downsampling factor: 24x24.

grid based sparse depth as input. However, in practical applications, obtaining uniform data can act as a limitation which is addressed in this work as explained in 3.1.

In Fig. 8, the first column contains RGB image from NYUv2 dataset, second column contains predicted dense depth map from proposed DRS autoencoder without GAMSE loss function, third column contains dense depth map prediction with GAMSE loss function, and the fourth column contains the ground truth dense depth map. It is

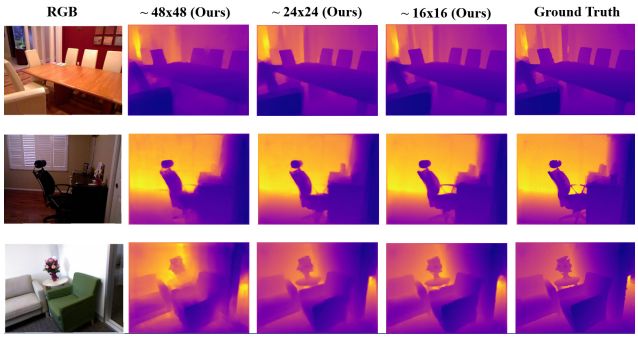

Figure 9. **Visualization of our results on different sampling Rate:** First column: input RGB images, second column: DRS (our) model predictions for downsampling factor: 48x48, Third column: DRS (our) model predictions for downsampling factor: 24x24, Fourth column: DRS (our) model predictions for downsampling factor: 12x12, Fifth column: ground truth.

observed that the dense depth predictions with the network trained with GAMSE loss function provides clear, smooth and better dense depth maps when compared to the network which is trained only on MSE. As the error at the edges decreases, the dense depth map predictions become more accurate.

5. Conclusions

In this paper we have proposed a novel framework for dense depth estimation using RGB and sparse sensing. We demonstrate that our novel DRS Auto Encoder network can operate on sparse depth information and RGB image to produce accurate dense depth maps similar to that of dedicated sensor hardware. Our model generalizes readily on diverse sparse input patterns and also introduce two new skip connections in the autoencoder. We have also proposed new

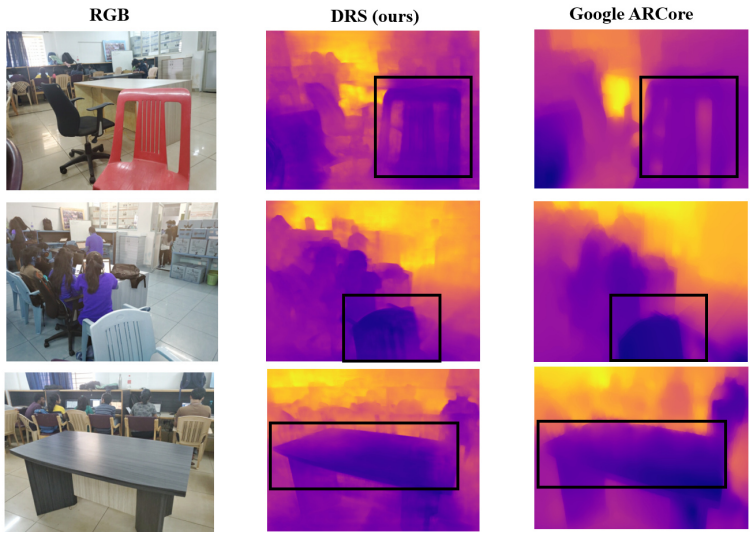

Figure 10. Visualization of our results versus ARCore dense depth map

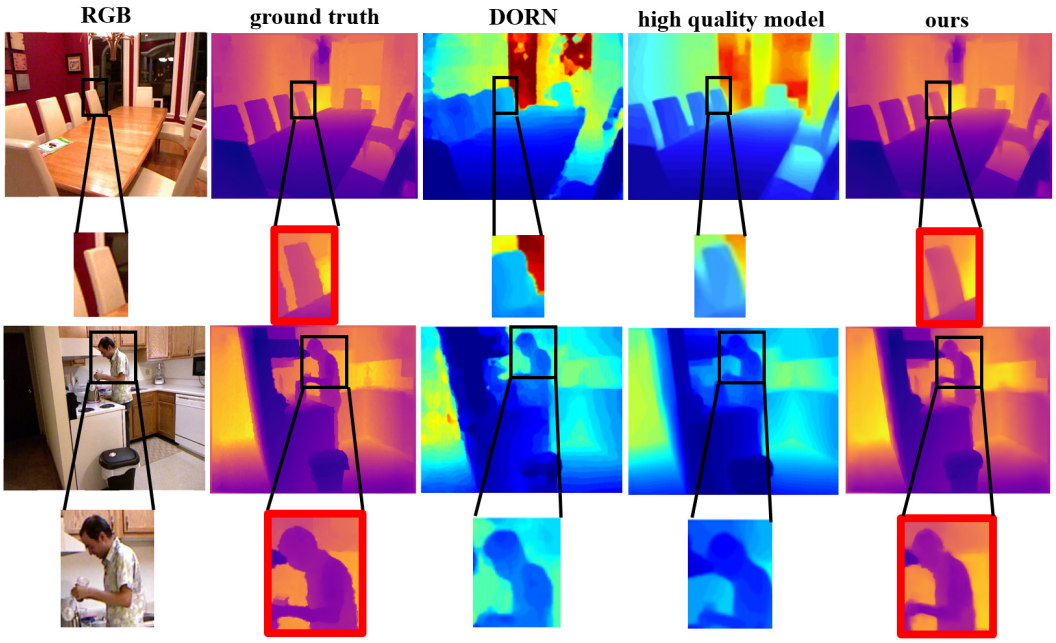

Figure 11. A gallery of estimated depth maps on the NYU depth v2 dataset: input RGB images, results of [11] (provided by the authors), results of [5] (provided by the authors), our estimated depth maps and ground truth depth maps. First column: input RGB image, Second column: ground truth dense depth map, Third column: results by DORN [11], Fourth column: results by High quality model [5], Fifth column: estimated dense depth map by proposed DRS autoencoder.

loss function for depth prediction at edges.

The next step would be to use dense depth for 3D reconstruction tasks, to reduce model parameters for on-device (mobile phones) applications and to show results on KITTI dataset [14]. We evaluate our DRS model for other vision tasks like super resolution. We hope that our work motivates additional research into use of DRS Autoencoder network.

6. Acknowledgement

This project is partly carried out under Department of Science and Technology (DST) through ICPS programme - Indian Heritage in Digital Space for the project “Crowd-Sourcing” (DST/ ICPS/ IHDS/ 2018 (General)) and “Digital Poompuhar” (DST/ ICPS/ Digital Poompuhar/ 2017 (General)).

References

[1] Mean absolute error. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, page 652. Springer, 2010. 6

[2] Mean squared error. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, page 808. Springer, 2017. 5, 6

[3] Pytorch: An imperative style, high-performance deep learning library. pages 8024–8035, 2019. 6

[4] Abien Fred Agarap. Deep learning using rectified linear units (relu). 03 2018. 5

[5] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *CoRR*, abs/1812.11941, 2018. 6, 8

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 2

[7] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. 2020. 3

[8] Zhao Chen, Vijay Badrinarayanan, Gilad Drozdov, and Andrew Rabinovich. Estimating depth from RGB and sparse sensing. *CoRR*, abs/1804.02771, 2018. 1, 2, 3

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014. 2

[10] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. 2018. 1

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *CoRR*, abs/1806.02446, 2018. 6, 8

[12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. 2

[13] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. 2016. 2

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 8

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. pages 6602–6611, 2017. 5

[16] Caner Hazirbas, Lingni Ma, Csaba Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. 2016. 1

[17] Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, and Kilian Weinberger. Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 4

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. 2017. 4

[19] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. 2016. 2

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. 5

[21] Maximilian Jaritz, Raoul de Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. 2018. 1

[22] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988. 2, 5

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2017. 6

[24] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2016. 1

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. 25, 2012. 1

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. pages 1106–1114, 2012. 1

[27] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. *CoRR*, abs/1702.02706, 2017. 2

[28] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. 1681:319, 1999. 3

[29] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. *CoRR*, abs/1804.00607, 2018. 5

[30] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, Oct 2016. 2

[31] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. *CoRR*, abs/1807.00275, 2018. 5

[32] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. 2012. 6

[33] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. 2020. 3

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 2016. 1

[35] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 2015. 1

[36] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. 18, 2006. 2

[37] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. 2017. 1, 2

[38] Travis Williams and Robert Li. Wavelet pooling for convolutional neural networks. 2018. 5

[39] W. Woo, Wonwoo Lee, and Nohyoung Park. Depth-assisted real-time 3d object detection for augmented reality. 2011. 1

[40] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. pages 161–169, 2017. 3

[41] Matthew Zeiler and Rob Fergus. Visualizing and understanding convolutional neural networks. *ECCV 2014, Part I, LNCS 8689*, 8689, 11 2013. 5