Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Monocular Depth Estimation using a Multi-grid Attention-based Model

Sangam Man Buddhacharya[1], Rabin Adhikari[2], Nischal Maharjan[3], Sanjeeb Prasad Panday[4]

[1-3]Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus, Tribhuvan University, Lalitpur, Nepal
[4]Associate Professor, Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus, Tribhuvan University, Lalitpur, Nepal

E-mail: [1]073bex438.sangam@pcampus.edu.np, [2]074bct529.rabin@pcampus.edu.np, [3]073bex421.nischal@pcampus.ed u.np, [4]sanjeeb@ioe.edu.np

Abstract

With the increased use of depth information in computer vision, monocular depth estimation has been an emerging field of study. It is a challenging task where many deep convolutional neural network-based methods have been used for depth prediction. The problem with most of these approaches is that they use a repeated combination of max-pooling and striding in an encoder, which reduces spatial resolution. In addition, these approaches use information from all the channels directly from the encoder, which is prone to noise. Addressing these issues, we present a multigrid attention-based densenet-161 model. It consists of a multigrid densenet-161 encoder that increases the spatial resolution and an attention-based decoder to select the important information from low-level features. We achieved absolute relative error (Absrel) of 0.109 and 0.0724 on NYU v2 and KITTI, dataset respectively. Our proposed method exceeded most evaluation measures with fewer parameters compared to the state-of-the-art on standard benchmark datasets. We produce a dense depth map from a single RGB image which can be used to create a dense point cloud. The anticipated depth map is accurate and smooth, which can be used in several applications.

Keywords: Convolutional Neural Network (CNN), depth estimation, dilation rate, multigrid, attention mechanism, depth map

1. Introduction

Depth estimation from a two-dimensional image is one of the most challenging tasks in computer vision. It has been studied for a long time and has a wide range of applications in computer vision like augmented reality [1], robotic navigation, autonomous driving car,

DOI: https://doi.org/10.36548/jiip.2022.3.001
Received: 29.06.2022, received in revised form: 17.07.2022, accepted: 02.08.2022, published: 12.08.2022

semantic segmentation [2], image refocusing [3], scene understanding, and 3D reconstruction. Different approaches have been used to estimate the depth of two-dimensional images, such as stereo vision and structure-from-motion [4]. Since it requires multiple views of the same scene or multiple moving frames, those approaches may not always be suitable. There might be situations where multiple scenes are not available or such a setup is not possible. The above constraints motivated the need for monocular depth estimations through computer vision models.

During the projection of 3D structure into the 2D image plane, some depth cues and 3D knowledge are preserved. Estimating depth accurately from a single 2D image is a difficult task because the multiple 3D points can project into the same 2D location. To estimate depth, humans use different local depth cues like shadowing, texture gradient, standard size, relative size, perspective, occlusion condition, and layout of the entire shape. Deep convolutional neural networks (DCNN) have recently shown promising results in different visual tasks.

Different DCNN-based supervised [5, 6, 7, 8, 9] and semi-supervised [10] learning models have attained splendid outcomes in monocular depth estimation. In recent works, the architecture is observed to be composed of two modules viz. dense feature extractor and depth predictor (decoder). High-performance classification models such as VGG, Resnet, and Densenet [11-13] are common choices for a dense feature extractor. The problem with this type of network is the reduced feature resolution caused due to repetitive pooling or stridden convolution operation. Reduction in feature resolution causes decimation of depth cues and spatial information. Additionally, these approaches take information from all channels without filtering relevant features, so the information is prone to noise.

To obtain a high-resolution dense feature map, several techniques such as multi-scale networks [8] and multi-layer deconvolutional networks [14, 15] are used, which require high computational and memory costs with complicated network architecture. Considering the higher-order 3D geometric constraints such as surface normal [16], plane coefficient, and ray plane intersection [17] has shown great improvement in depth estimation. However, the problems with this method are its complex network structure and numerous parameters, which make it ill-suited for real-time 3D reconstruction. In augmented reality, synthetic depth-of-field, and other image effects [18, 19, 20] require fast and highly accurate depth estimation. Addressing these issues, we propose a simple multi-grid attention-based network architecture that produces highly accurate and quality depth estimations. Experimental results

on a standard benchmark show that our proposed method has surpassed most of the evaluation measures with a few network parameters compared to the state-of-the-art. The resulting depth maps are smooth and on par, if not more promising, compared to those generated by existing methods with fewer network parameters and reduced inference time. During testing, we adopt a post-processing technique that improves the evaluation metrics.

Our contributions are as follows: a) Experiments with different CNN Architectures. b) Proposed a multigrid attention-based monocular depth estimation model. c) Study of our model performance on different depth ranges. d) Study of the effect on depth map with a scaling factor

2. Related Work

2.1 Supervised Depth Estimation

In general, the supervised approach takes images along with their depth data. The image is fed into the network and the corresponding output is tried to match that of the depth data. The depth data used as labels for this purpose are taken from multi-channel laser scanners or RGB-D cameras, or IR-based sensors. Saxena et al. [21] proposed a supervised approach to depth estimation from a single monocular image using a discriminatively-trained Markov Random Field (MRF) and extended it to use over-segmentation when introducing a 3D dataset for the specific task [5]. Eigen and Fergus [7] proposed a multi-scale Convolutional Neural Network (CNN) capable of outputting depth maps directly from the input image using a progressive refinement of predictions using a sequence of scales.

Due to the end-to-end nature of the method, many works improved upon this approach by incorporating several constraints and intermediate representations for surface normal estimation [22], using a post-processing refining step using Conditional Random Fields (CRFs) [23, 24, 25], or by discretizing the continuous depth values into multiple bins and labeling them with respect to their depth range [26]. Fu et al. [27] introduced a Spacing-Increasing Discretization (SID) strategy to discretize depth and formulated a regression problem as an ordinal regression problem using a multi-scale network structure. Gan et al. [28] modeled the relationships of different image locations with an affinity layer and combine absolute and relative features in an end-to-end network. Yin et al. [16] enforced high-order 3D geometric constraints by randomly sampling three points from the reconstructed 3D space to determine virtual normal directions.

2.2 Semi-Supervised Depth Estimation

Semi-supervised ways of estimating the depth of images include a secondary training objective along with the primary one, i.e., supervised training, to incorporate the lack of training data required for supervised learning. Chen et al. [29] introduced a dataset “Depth in the Wild" consisting of images in the wild annotated with relative depth between pairs of random points and proposed an algorithm that learns to estimate metric depth using annotations of that depth. Kuznietsov et al. [10] used sparse ground-truth depth from LiDAR sensors for supervised learning and enforced the network to produce photo-consistent dense depth maps in a stereo setup using a direct image alignment loss.

3. Proposed Work

In this section, we describe the implemented architecture, including the loss function, augmentation policy, and post-processing technique used during training and testing the network.

3.1 Network Architecture

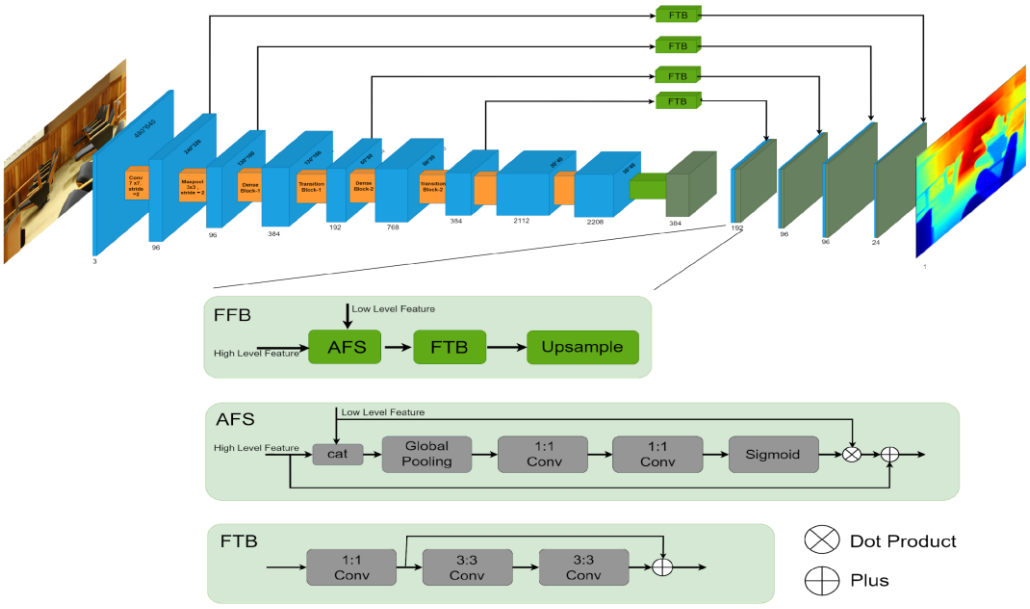


Figure 1. Proposed Network Architecture

The overview of the proposed architecture is illustrated in Figure 1. The network is composed of an encoder, a transitional block, and a decoder. For the encoder, to obtain a high-resolution dense feature map, we removed the pooling layers from the last block and introduced dilated convolutions with multiple dilation rates [30, 31] to build a multi-grid

densenet-161 network. The encoder encodes the input RGB image into a dense feature vector enriched with higher-level features. The feature vector is fed into a transition block which comprises a simple 1 × 1 convolutional layer followed by the batch normalization and ReLU activation function.

The purpose of the 1 × 1 convolutional layer is to reduce the channel of the feature map generated by densenet-161. Inspired by Li et al. [32], the decoder is composed of several Feature Fusion Blocks (FFBs). The FFB Block is comprised of an Attention-based Feature Selection block (AFS) followed by a Feature Transform Block (FTB) and an Upsampling Layer. The purpose of FFBs is to combine features from different levels. FTB transforms features from the encoder suitable for the depth estimation task. The lower-level features from the encoder are passed to the Attention-based Feature Selection block (AFS) which prioritizes important channels from the low-level features that are summed to the high-level features. The encoded feature vector has an output resolution of $\frac{H}{16} \times \frac{L}{16}$, so to get the original resolution, an upsampling layer and associated skip connections are used. Instead of using deconvolution ("transposed convolution"), we have used upsampling (nearest neighbor) followed by convolution as our decoding layer to reduce the checkerboard artifacts [33].

3.2 Multigrid

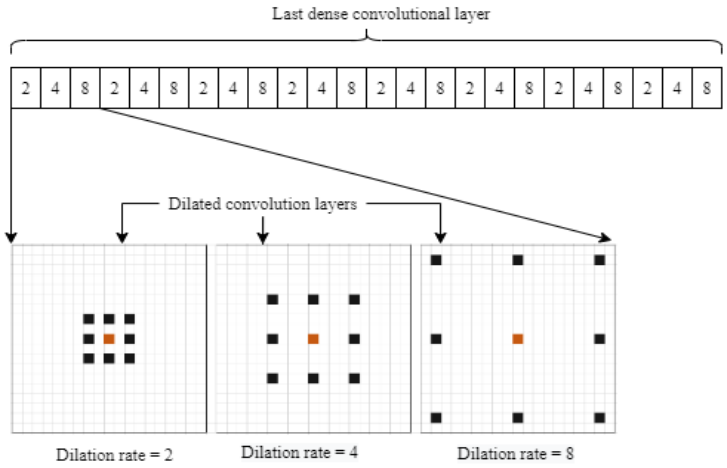


Figure 2. Multi Grid convolution

As aforementioned, a multigrid densenet-161 network is used as the encoder. So to overcome the decimation of depth cues and spatial information, the concept of multigrid has been used. The pooling layer from the last block of the densenet has been removed and dilated convolutions have been introduced to obtain a high-resolution dense feature map [30,

31]. Since using the same dilation rate causes the "gridding" problem [31], we used different dilation rates for each layer. Continuous layers within the last block are assumed to form a group to eliminate the issue. A layer within a group has dilation of two, four, and eight, as shown in Figure 2. The preceding groups repeat the same pattern to form saw-tooth-like waves. Therefore, 24 layers of the last block of the densenet-161 have eight groups that follow the same rising edge of increasing dilation rate.

3.3 Attention Mechanism

One of the primary blocks present in the decoder is the Attention-based Feature Selection (AFS) block. Generally, not all features have equal importance in generating the results. So focusing on the relevant features is a crucial task carried out by the AFS Block [32]. The structure of AFS is shown in Figure 1. The high-level feature being encoded by the encoder is concatenated with the low-level features obtained from the skip connection. Rather than using all channels from the low-level features, the AFS block selects channels representing relevant features using the attention mechanism as shown in Figure 3. The dense feature maps are squeezed into a single vector with a global max pooling layer and are denoted by F_{sq} and the corresponding weight parameters are $F_{ex}(W)$. These parameters are learned during the training process, which gives relevant features higher weights. Thus only important feature channels are extracted from input F_{sq} to generate a scaling vector of shape 1 × 1 × c, denoted by F_{scale}, that contains scaling weights for each input dense feature map. The scaling weights excite only the relevant channel required for a given task.

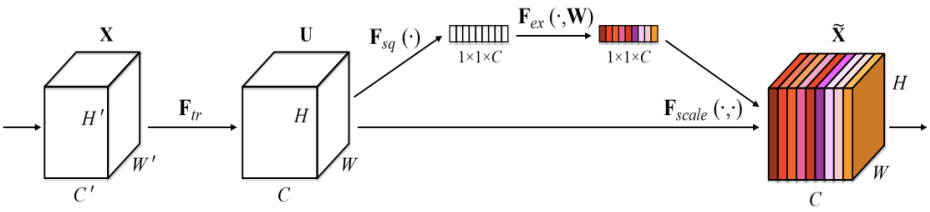


Figure 3. Selecting important features [34].

3.4 Training Loss

A larger distance has a large error contributing more to the loss function. As a result, models optimize themselves to reduce error for larger distances and neglect the error for smaller distances. To address this problem, we transfer the depth values in log space. Inspired by Eigen et al. [6], we used a scale-invariant loss function as shown in Equation 1.

$$l(d) = \frac{1}{T}\sum_i d_i^2 - \frac{\lambda}{T^2}(\sum_i d_i)^2 \quad (1)$$

Where, $d_i = \log\left(\frac{gt_i}{p_i}\right)$, gt_i is the ground truth depth, p_i is the predicted depth, λ is the mixing parameter, and T denotes the number of pixels consisting valid ground truth values.

From Equation 1, we get:

$$l(d) = \frac{1}{T}\sum_i d_i^2 - \frac{1}{T^2}(\sum_i d_i)^2 + \frac{1-\lambda}{T^2}(\sum_i d_i)^2 \quad (2)$$

From Equation 2, we can find that the loss is the sum of the variance and a weighted squared mean of the error in the log space. Hence, setting a higher λ enforces more focus on reducing the error variance, and we uncovered $\lambda = 0.85$ to best suit our work. Inspired by Lee et al. [17], properly scaling the range of the loss function betters the convergence and thus, the final training outcome. Our final training loss *L* is as follows:

$$L = \alpha\sqrt{l(d)} \quad (3)$$

As per Lee et al. [17], we experimented with different values of α and found $\alpha = 10$ provides better performance.

3.5 Augmentation Policy

To improve generalization and avoid over-fitting, we adopted a geometric and color augmentation policy before training our network. We only considered horizontal flipping, i.e., mirroring, of the input image with a 50% of chance. We did not flip the image vertically since it didn't increase the model performance and the flipped image can occur as a non-augmented example (the sky will not be at the bottom). For the KITTI [35] dataset, we rotated the input image randomly in a range of $[-1, 1]$ degrees. Inspired by Lee et al. [17], we added random contrast, brightness, and gamma to the input image in a range of [0.9, 1.1], with a chance of 50%. Before feeding the images to the model at training, we randomly cropped them to the size of 416 × 544 for NYU V2 [36] and 353 × 704 for KITTI [35] datasets.

3.6 Post-Processing

An additional post-processing technique has been implemented during testing, shown in Figure 4. The input RGB image is flipped horizontally and stacked over the original image. The stacked images are forwarded to the network to predict depth maps. The depth map

corresponding to flipped image denoted by d_f is reversed back and is averaged with the depth map corresponding to the original image d_0 to obtain the final depth map d as shown in Equation 4. From Table 3, we can observe that using the post-processing technique has improved overall evaluation metrics.

$$d = \frac{d_o + d_f}{2} \tag{4}$$

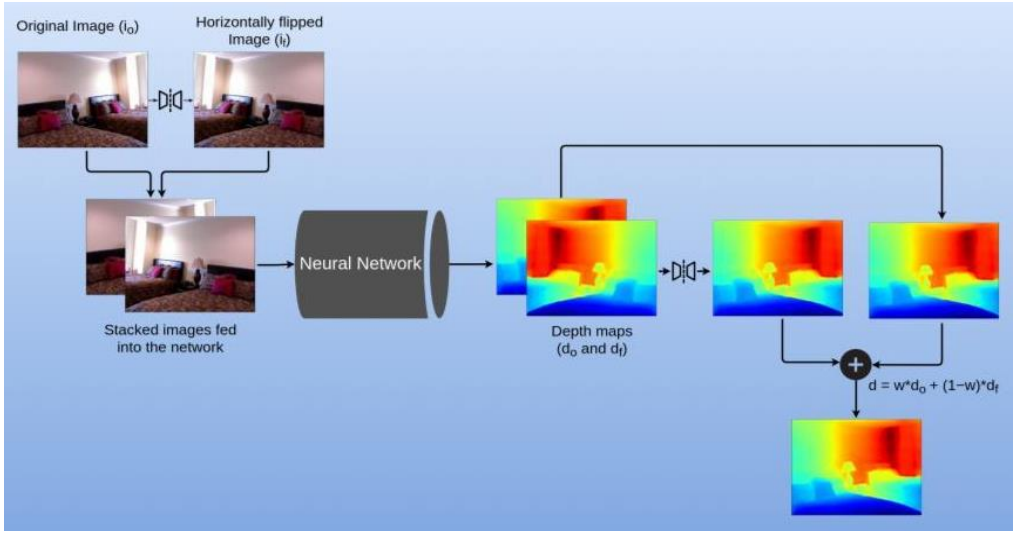


Figure 4. Post-Processing

4. Experiments

We conducted various experiments on two standard benchmarks to test and verify the effectiveness of our proposed method. We also compared our results with the state-of-the-art and found our method is on par, if not superior, compared to those generated by existing methods on a standard benchmark.

4.1 Datasets

4.1.1 NYUD-V2

The NYUD-V2 [36] dataset contains 464 different indoor scenes captured as video sequences using Microsoft Kinect at the resolution of 480 × 640. The dataset contains 120 thousand RGB-D pairs for training and 654 images for testing. Using the official train-test split, we train our method on a subset of 42,000 images taken from 249 scenes and tested with 654 RGB-D pairs from 215 scenes (654 images). We then align the raw RGB image and depth map for accurate pixel registrations, using the camera projection with the NYU toolbox in MATLAB.

4.1.2 KITTI

KITTI [35] dataset contains data from 61 different outdoor scenes having over 93,000 RGB and depth map pairs captured on driving cars with cameras and lidars. For training and testing, we follow the Eigen split [6] to compare with previous works.

4.2 Implementation Details

To implement our proposed network, we used TensorFlow [37] as our deep learning framework. We used the Adam optimizer [38] to train the model with $\beta_1 = 0.9,\ \beta_2 = 0.999$, and $\epsilon = 10^{-8}$ along with an exponential decaying learning rate of 0.96 for every 2,000 steps and starting learning rate of 10^{-4}. Each dataset was trained for 20 epochs. We performed our experiments using Google Colab with either Nvidia T4 or P-100 GPU. We use DenseNet-161 [13] as our base model for encoder pre-trained on ImageNet [39] classification datasets.

4.3 Evaluation Metrics

For evaluating our work, we use the following metrics.

$$Absrel: \frac{1}{T}\sum_{d \in T} \frac{|d'_i - d_i|}{d_i}$$

$$RMSE: \sqrt{\frac{1}{T}\sum_{d \in T} (d'_i - d_i)^2}$$

$$\log 10: \sqrt{\frac{1}{T}\sum_{d \in T} (\log d'_i - \log d_i)}$$

$$\text{Threshold: percentage of } d_i \text{ such that } \max\left(\frac{d_i}{d'_i}, \frac{d'_i}{d_i}\right) = \delta < threshold$$

where d'_i and d_i denote predicted depth and corresponding ground truth values of i^{th} pixel respectively and T is the total number of pixels for which there exist both valid ground truth and predicted depth.

4.4 Ablation study

4.4.1 Experiments with different Architectures

In this experiment, we started with a simple encoder-decoder architecture and tested with different encoders, i.e., densenet-121 (Simple 121), densenet-169 (Simple 169), and densenet161 (Simple 161) [13]. As in Table 1, densenet-161 performed best in most metrics,

so we selected densenet-161 as our encoder. After selecting the encoder, we added Atrous Spatial Pyramid Pooling (ASPP 161) and multigrid (multi 161) in the densenet-161 block. We found improvement in the model's performance after adding multigrid but no refinement with ASPP, so we continued our experiment with multigrid in the densenet-161 block. Further cascading several multigrid blocks in the last block of densenet-161 (Cascaded 161) slightly increased the performance in trade-off with parameters size and training time, so we discarded the cascaded block. To find the influence of the attention mechanism, we used the decoder, inspired by Li et al. [32]. As shown in Table 1, we observed a significant improvement in the performance of our models by adding an attention mechanism with multigrid (Multi attention 161).

Table 1. Evaluating Results on NYUv2 Dataset [36] with different architectures. Using a multigrid with densenet-161 [13] improved our performance. Enforcing attention mechanisms with multi-grid has shown the best result in all metrics. Using medians as a scalar factor improves our performance to a greater extent. The best results without scalar factor are bold and the results obtained after scaling are underlined.

Methods	**#Params**	**lower values are better**			**higher values are better**		
		AbsRel	RMSE	log10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Simple 121	8.8M	0.1235	0.415	0.052	0.849	0.9472	0.994
Simple 161	28M	0.1167	0.398	0.049	0.866	0.975	0.995
Simple 169	14M	0.1194	0.403	0.050	0.860	0.977	0.995
Multi 169	-	0.1145	0.386	0.048	0.873	0.978	0.995
Multi 161	29M	0.1129	0.386	0.047	0.876	0.979	0.996
ASPP 161	31M	0.1169	0.399	0.050	0.862	0.977	0.996
Cascaded 161	32M	0.1117	0.383	0.047	0.875	0.980	0.996
Multi attention 161	31M	**0.110**	**0.381**	**0.047**	**0.877**	**0.981**	**0.996**
With median scalar factor	31M	0.0909	0.341	0.038	0.913	0.983	0.997

4.4.2 Study of our model performance on different depth ranges

Our study shows that the regression loss function focuses more on the mean error; consequently, it tends to converge to the mean depth values, and this causes more significant errors in areas that are either too far from or too close to the camera. To examine this problem, we separated the testing depth map into three different ranges, a shorter range (0m - 3m), a middle range (3m - 7m), and a more extended range (7m - 10m). We obtained the

results as shown in Table 2. We can observe that the depth map in the middle range has performed better in all evaluation metrics. Therefore, the middle range distance is optimized more than nearer and farther distance.

Table 2. Result of NYU v2 [36] test dataset for different ranges. Error is minimum for range (3m - 7m) and maximum for range (7m - 10m). The best output is shown in bold.

Ranges	lower is better					higher is better		
	AbsRel	log10	RMSE	sqRel	logRMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
0m - 3m	0.1148	0.048	0.288	0.053	0.142	0.869	0.978	0.995
3m - 7m	**0.1105**	**0.046**	**0.510**	**0.106**	**0.130**	**0.892**	**0.974**	**0.993**
7m - 10m	0.1584	0.064	1.176	0.321	0.166	0.805	0.973	0.983

4.4.3 Effect of scaling factor

The predicted depth is a multiple of some scalar factors. Because if we multiply our predicted result with a constant such that its median matches the median of the ground truth, then it produces a highly accurate depth map.

$$Scalefactor = \frac{median\ of\ ground{-}truth\ depth}{median\ of\ predicted\ depth} \tag{5}$$

The result after scaling is shown in Table 1, we can observe that error has drastically reduced. With this fact, we can conclude that our predicted depth map is slightly wrongly scaled but has a very smooth and promising relative depth map, which can be confidently used in different applications such as augmented reality, 3D reconstruction, image refocusing, etc.

4.5 Comparison with state-of-the-art results on a standard dataset

In this section, we compare the results from our model with a few previous approaches on the standard dataset, NYUv2 [36], and KITTI [35]. Table 3 shows that our proposed method with post-processing exceeds other existing methods across the most evaluation metrics. Compared to Yin et al. [16] and Lee et al. [17], we have reduced the RMSE error by 3.6% and 1.2% respectively.

To test the generalization of our method, we also tested the proposed model on the outdoor scene KITTI dataset. Results in Table 4 show that our model has improved in

RMSElog and $\delta < 1.25$ metrics along with comparable results with previous state-of-the-art methods on other metrics.

Table 3. Evaluation results on NYU Depth v2 [36]. The best output is shown in bold.

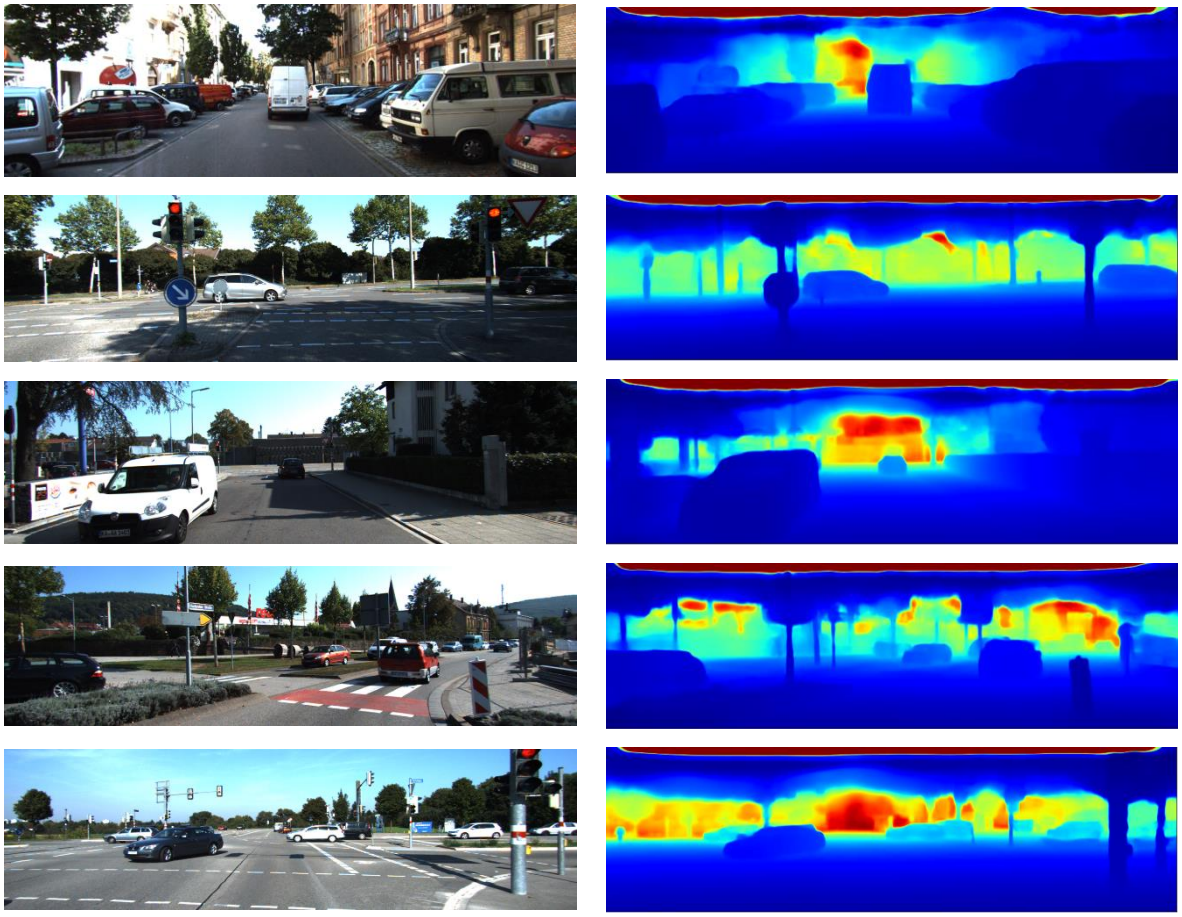
Methods	#Params	Lower values are better			Higher values are better		
		Absrel	RMSE	log10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena et al. [5]	-	0.349	1.214	-	0.447	0.745	0.897
Wang et al. [9]	-	0.220	0.824	-	0.605	0.890	0.970
Liu et al. [8]	-	0.213	0.759	0.087	0.650	0.906	0.976
Eigen and Fergus [7]	-	0.158	0.641	-	0.769	0.950	0.988
Chakrabarti et al. [43]	-	0.149	0.620	-	0.806	0.958	0.987
Li et al. [44]	-	0.152	0.611	0.064	0.789	0.955	0.988
Laina et al. [14]	-	0.127	0.573	0.055	0.811	0.953	0.988
Xu et al. [45]	-	0.121	0.586	0.052	0.811	0.954	0.987
Lee et al. [46]	-	0.139	0.572	-	0.815	0.963	0.991
Fu et al. [27]	110M	0.115	0.509	0.051	0.828	0.965	0.992
Qi et al. [47]	-	0.128	0.569	0.057	0.834	0.960	0.990
Yin et al. [16]	-	**0.108**	0.416	0.048	0.875	0.976	0.994
Lee et al. [17]	47.0M	0.110	0.392	0.047	**0.885**	0.978	0.994
Ours	31M	0.110	0.381	0.047	0.877	0.981	0.996
Ours with post-processing	31M	0.109	**0.380**	**0.047**	0.879	**0.982**	**0.996**

Table 4. Evaluation results on KITTI Eigen split [6] for the range of 0-80m. (CS+K) represents a model pre-trained on the Cityscapes dataset and fine-tuned with KITTI [35].

Methods	Lower is better				Higher is better		
	AbsRel	SqRel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena et al. [21]	0.280	3.012	8.734	0.361			
Eigen et al. [6]	0.203	1.548	6.307	0.282	0.702	0.898	0.967
Liu et al. [8]	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Godard et al. [48] (CS+K)	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Kuznietsov et al. [10]	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Gan et al. [28]	0.098	0.666	3.933	0.173	0.890	0.964	0.985
Fu et al. [27]	**0.072**	**0.307**	**2.727**	0.120	0.938	**0.990**	**0.998**
Yin et al. [16]	**0.072**	-	3.258	0.117	0.938	**0.990**	**0.998**
Ours with post-processing	**0.072**	0.331	3.169	**0.113**	**0.939**	**0.990**	**0.998**

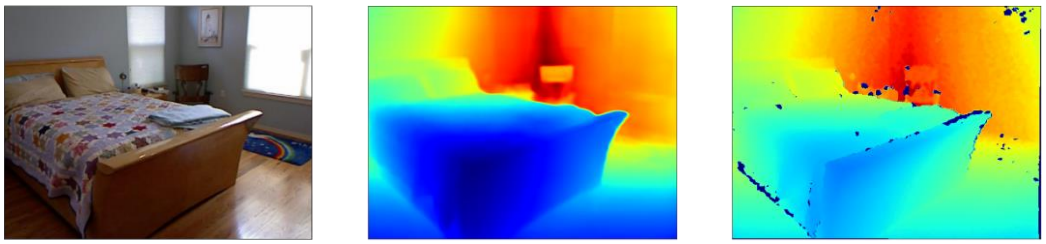
4.6 Outputs on NYU and KITTI datasets

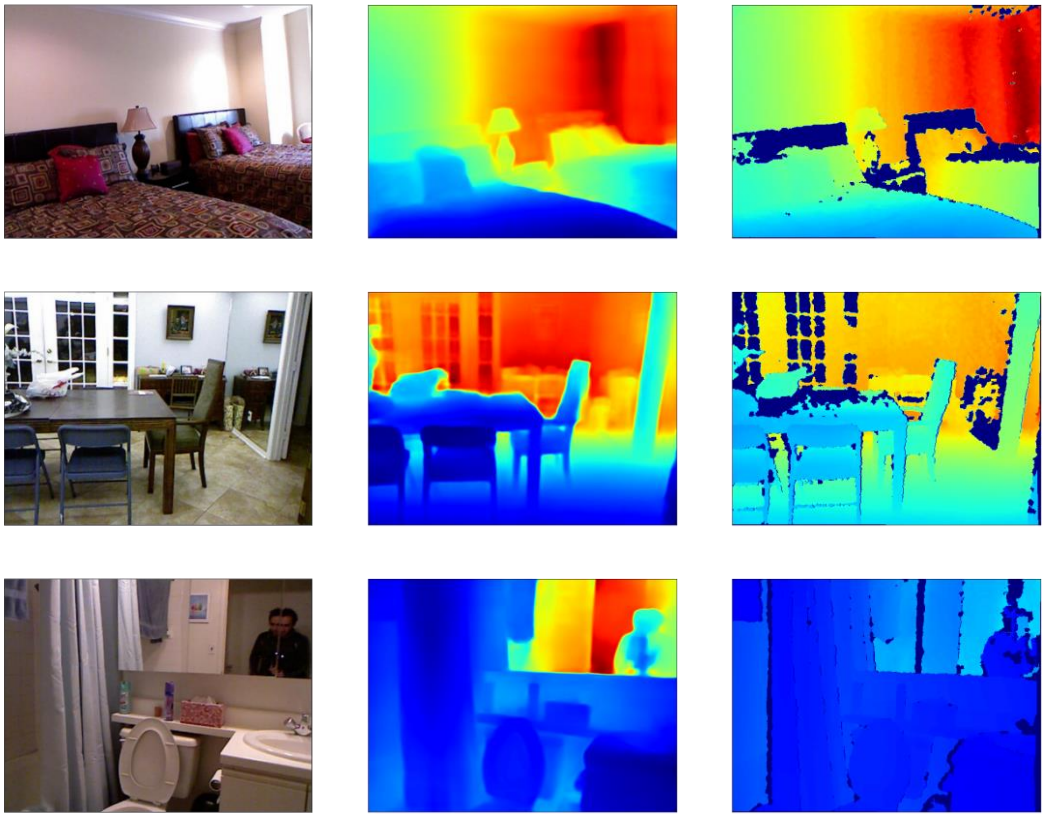
The predicted output from our proposed model on KITTI and NYU v2 datasets is shown in Figure 5 and Figure 6, respectively.



(a) RGB Image (b) Predicted output

Figure 5. Estimating depth by our proposed model on KITTI [35] dataset. a) and b) are the Input RGB images and their respective predicted depth map. Blue represents nearer, and red represents farther. Output from our model is continuous and smooth with fewer artifacts. Although the predictions don't contain any artifacts, the error seems to be large for larger distances, i.e., the sky seems to have different depths in different areas.





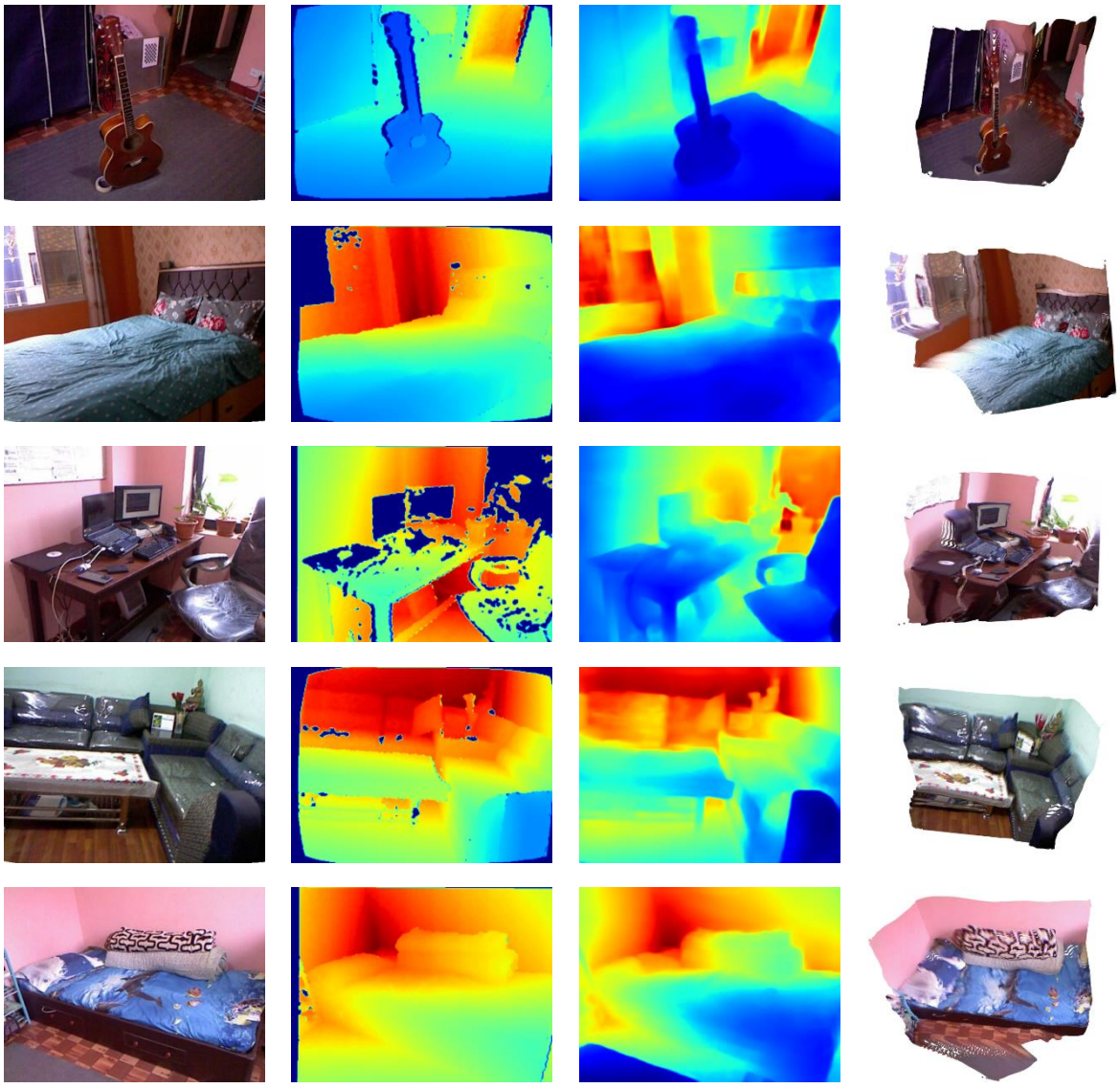
(a) RGB Image (b) Predicted depth map (c) Ground truth

Figure 6. The estimated depth by the proposed model on the NYUv2 dataset [36]. a) Input RGB image, b) predicted depth map and c) ground truth. Blue represents nearer, and red represents farther. Our prediction has a similar edge to that of the ground truth without any presence of artifacts. The predictions are smooth, almost resembling the ground truth. The occlusions created from Kinect due to its stereo effect from the sensor and IR projector are greatly minimized. We tested our model with an image containing a mirror and found that our model predicts a depth map as if there is not a mirror and the person is located at a farther distance.

4.7 Outputs on a Custom dataset

To better understand the performance of our proposed method, we have also tested our model on a custom dataset. For dataset preparation, XBOX 360 Kinect Sensor has been used to capture RGB images and the depth map of various objects in the room. The depth map from the Kinect has been used as ground truth to compare with the predicted output of our proposed model. The results can be seen in Figure 7. The outputs show that our method produces smooth and quality results with fewer artifacts in the diverse scene; therefore, it can be used for different applications in computer vision. We also generated the dense point

cloud from the predicted depth map and found that some flat objects are predicted to be curved.



(a) RGB Image (b) Ground truth (c) Predicted depth map (d) Dense point cloud

Figure 7. Images captured with Kinect V1

5. Conclusion

This paper proposes a monocular depth estimation network using multigrid densenet-161 as an encoder and attention-based decoder. By adding multigrid, we achieved a higher spatial resolution from the encoder, and the attention mechanism in the decoder further filtered the channels. Therefore, it improved the model performance significantly, which is verified by our experimental results. The proposed method surpasses most evaluation metrics on the NYU v2 dataset while achieving comparable results on the KITTI dataset, with

comparatively fewer parameters. Analyzing the output depth maps on NYU v2 and KITTI dataset, the model's predictions have smooth and pretty clear boundaries at the edges with continuous varying depth at its surface. Additional study on different ranges shows that our model works best for the range (3m -7m) for the indoor scene. Further study on the scaling factor confirms that the predicted depth is slightly scaled by a constant factor, which makes it suitable for the application that requires relative depth. These results of the scaling factor leave room for future research on finding the appropriate multiplying factor and improving the model's overall performance.

References

[1] João Paulo Silva do Monte Lima et al. "Depth-assisted rectification for real-time object detection and pose estimation". In: Machine Vision and Applications 27.2 (2016), pp. 193– 219.

[2] Caner Hazirbas et al. "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture". In: Asian conference on computer vision. Springer. 2016, pp. 213–228.

[3] Francesc Moreno-Noguer, Peter N Belhumeur, and Shree K Nayar. "Active refocusing of images and videos". In: ACM Transactions On Graphics (TOG) 26.3 (2007), 67–es.

[4] Rene Ranftl et al. "Dense monocular depth estimation in complex dynamic scenes". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 4058–4066.

[5] Ashutosh Saxena, Min Sun, and Andrew Y Ng. "Make3d: Learning 3d scene structure from a single still image". In: IEEE transactions on pattern analysis and machine intelligence 31.5 (2008), pp. 824–840.

[6] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network". In: Advances in neural information processing systems. 2014, pp. 2366–2374.

[7] David Eigen and Rob Fergus. "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture". In: Proceedings of the IEEE international conference on computer vision. 2015, pp. 2650–2658.

[8] Fayao Liu et al. "Learning depth from single monocular images using deep convolutional neural fields". In: IEEE transactions on pattern analysis and machine intelligence 38.10 (2015), pp. 2024–2039.

[9] Peng Wang et al. “Towards unified depth and semantic prediction from a single image”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 2800–2809.

[10] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. “Semi-supervised deep learning for monocular depth map prediction”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). July 2017, pp. 6647–6655.

[11] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for largescale image recognition”. In: arXiv preprint arXiv:1409.1556 (2014).

[12] Kaiming He et al. “Deep residual learning for image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778.

[13] Gao Huang et al. “Densely connected convolutional networks”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 4700–4708.

[14] Iro Laina et al. “Deeper depth prediction with fully convolutional residual networks”. In: 2016 Fourth international conference on 3D vision (3DV). IEEE. 2016, pp. 239–248.

[15] Ravi Garg et al. “Unsupervised cnn for single view depth estimation: Geometry to the rescue”. In: European conference on computer vision. Springer. 2016, pp. 740–756.

[16] Wei Yin et al. “Enforcing geometric constraints of virtual normal for depth prediction”. In: Proceedings of the IEEE International Conference on Computer Vision. 2019, pp. 5684– 5693.

[17] Jin Han Lee et al. “From big to small: Multi-scale local planar guidance for monocular depth estimation”. In: arXiv preprint arXiv:1907.10326 (2019).

[18] Peter Hedman and Johannes Kopf. “Instant 3D photography”. In: ACM Transactions on Graphics (TOG) 37.4 (2018), pp. 1–12.

[19] Shunsuke Saito et al. “3D hair synthesis using volumetric variational autoencoders”. In: ACM Transactions on Graphics (TOG) 37.6 (2018), pp. 1–12.

[20] Lijun Wang et al. “DeepLens: shallow depth of field from a single image”. In: arXiv preprint arXiv:1810.08100 (2018).

[21] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. “Learning depth from single monocular images”. In: Advances in neural information processing systems. 2006, pp. 1161–1168.

[22] Xiaolong Wang, David Fouhey, and Abhinav Gupta. “Designing deep networks for surface normal estimation”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 539–547.

[23] Bo Li et al. “Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 1119–1127.

[24] Alexander G Schwing and Raquel Urtasun. “Fully connected deep structured networks”. In: arXiv preprint arXiv:1503.02351 (2015).

[25] Patrick Knobelreiter et al. “End-to-end training of hybrid CNN-CRF models for stereo”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 2339–2348.

[26] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. “Estimating depth from monocular images as classification using deep fully convolutional residual networks”. In: IEEE Transactions on Circuits and Systems for Video Technology 28.11 (2017), pp. 3174–3182.

[27] Huan Fu et al. “Deep ordinal regression network for monocular depth estimation”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 2002–2011.

[28] Yukang Gan et al. “Monocular depth estimation with affinity, vertical pooling, and label enhancement”. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 224–239.

[29] Weifeng Chen et al. “Single-Image Depth Perception in the Wild”. In: Advances in Neural Information Processing Systems. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016, pp. 730–738.

[30] Liang-Chieh Chen et al. “Rethinking atrous convolution for semantic image segmentation”. In: arXiv preprint arXiv:1706.05587 (2017).

[31] Panqu Wang et al. “Understanding convolution for semantic segmentation”. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE. 2018, pp. 1451– 1460.

[32] Ruibo Li et al. “Deep attention-based classification network for robust depth prediction”. In: Asian Conference on Computer Vision. Springer. 2018, pp. 663–678.

[33] Augustus Odena, Vincent Dumoulin, and Chris Olah. “Deconvolution and checkerboard artifacts”. In: Distill 1.10 (2016), e3.

[34] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-excitation networks”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp. 7132–7141.

[35] Andreas Geiger et al. “Vision meets Robotics: The KITTI Dataset”. In: The International Journal of Robotics Research 32.11 (2013), pp. 1231–1237.

[36] Nathan Silberman et al. “Indoor segmentation and support inference from rgbd images”. In: European conference on computer vision. Springer. 2012, pp. 746–760.

[37] Martín Abadi et al. “Tensorflow: A system for large-scale machine learning”. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). 2016, pp. 265–283.

[38] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: arXiv preprint arXiv:1412.6980 (2014).

[39] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE. 2009, pp. 248–255.

[40] Ayan Chakrabarti, Jingyu Shao, and Greg Shakhnarovich. “Depth from a single image by harmonizing overcomplete local network predictions”. In: Advances in Neural Information Processing Systems. 2016, pp. 2658–2666.

[41] Jun Li, Reinhard Klein, and Angela Yao. “A two-streamed network for estimating fine-scaled depth maps from single RGB images”. In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 3372–3380.

[42] Dan Xu et al. “Multi-scale continuous CSRFs as sequential deep networks for monocular depth estimation”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 5354–5362.

[43] Jae-Han Lee et al. “Single-image depth estimation based on Fourier domain analysis”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 330–339.

[44] Xiaojuan Qi et al. “Geonet: Geometric neural network for joint depth and surface normal estimation”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 283–291.

[45] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. “Unsupervised monocular depth estimation with left-right consistency”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 270–279.

Author's biography

Sangam Man Buddhacharya received his Bachelor's degree in Electronics and Communication Engineering from Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal. He is currently working as a Machine Learning Engineer at Artlabs, New York. He has two years of working experience in research and industry. His current research interests include Computer Vision and Machine Learning. He has won three national

technological competitions and is a member of Sakura Science Club, Japan. More about him at https://www.linkedin.com/in/sangambuddhacharya/

Rabin Adhikari received his Bachelor's degree in Computer Engineering from Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal. He is enthusiastic about artificial intelligence and machine learning. His fields of interest are NLP, Computer Vision, and Medical Imaging. More about him at https://www.linkedin.com/in/rabinadk1/

Nischal Maharjan received his Bachelor's degree in Electronics and Communication Engineering from Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal. As an AI/ML enthusiast, he is keenly interested in Machine Learning and Computer Vision. He has been involved in Robotics Club, Pulchowk Campus, and participated in the international Robotic Competition ABU Robocon in 2019. More about him at https://www.linkedin.com/in/nischl-mhrjn/

Sanjeeb Prasad Panday received the Bachelors Degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan in 2001 A.D. and Masters Degree in Information and Communication Engineering from Tribhuvan University, Nepal in 2006 A.D.. He completed his Doctoral Degree in Information Systems Engineering at the graduate school of engineering, Osaka Sangyo University, Japan in 2011 A.D. He has been working as an associate professor in the Department of Electronics and Computer Engineering at Pulchowk Campus, Institute of Engineering, Tribhuvan University, Pulchowk, Lalitpur, Nepal since 2002 A.D. He is currently the Director of Information and Communication Technology Centre (ICTC). His research interests include Disaster Communications, RF and Microwave Techniques, Image Processing and Analysis, Computer Vision, Digital Holography and Optimization Algorithms.