Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

IMPROVED DEEP LEARNING ARCHITECTURE FOR DEPTH ESTIMATION FROM SINGLE IMAGE

Suhaila F. A. Abuowaida and Huah Yong Chan

(Received: 28-Jun.-2020, Revised: 3-Aug.-2020 and 20-Sep.-2020, Accepted: 27-Sep.-2020)

ABSTRACT

Numerous benefits of depth estimation from the single image field on medicine, robot video games and 3D reality applications have garnered attention in recent years. Closely related to the third dimension of depth, this operation can be accomplished using human vision, though considered challenging due to the various issues when using computer vision. The differences in the geometry, the texture of the scene, the occlusion scene boundaries and the inherent ambiguity exist because of the minimal information that could be gathered from a single image. This paper, therefore, proposes a novel depth estimation in the field of architecture, which includes the stages that can manage depth estimation from a single RGB image. An encoder-decoder architecture has been proposed, based on the improvement yielded from DenseNet that extracted the map of an image using skip connection technique. This paper also takes on the reverse Huber loss function that essentially suits our architecture hand driven by the value distributions that are commonly present in depth maps. Experimental results have indicated that the depth estimation architecture that employs the NYU Depth v2 dataset has a better performance than the other state-of-the-art methods that tend to have fewer parameters and require fewer training time.

KEYWORDS

Depth estimation, Single image, Deep learning, Encoder-decoder.

1. INTRODUCTION

Numerous benefits in the use of depth estimation from the single image field on medicine [33], robot video games [4] and 3D reality applications [15] have resulted in a spike of interest to the operations of 3D in recent years. Closely related to the third dimension of depth, this application can be accomplished using human vision, but is considered challenging through the computer vision. However, the differences in the geometry, the texture of a scene, the occlusion scene boundaries and the inherent ambiguity could not be captured in depth when employing a single image [26]. The estimation of depth is essential to obtain the ideal distance for each pixel in a single image between an observer and the visual detail [15]. Traditional algorithms that manage monocular images in recognising the dimension of depth, which include Structure from Motion (SfM) [27], as well as differences in shading and lighting of images [28] [1], require specific environmental assumptions. Besides, there are other issues related to finding the depth of the images, such as determining the heterogeneity of the depth [3] and the quality image processing after identifying the depth [22]. Initially, researchers have focused on the stereo vision to acquire depth estimation by using multi-view images [20] [32] [21]. Nevertheless, this method appears to have several setbacks, such as low efficiency of depth estimation due to blind region repetition and unmatching texture at areas of the same point. Since then, the use of a single image for depth estimation has been given much attention [7] [18] [9] due to the manageable cost, specialized equipment in use and flexibility in capturing the image. The devices, though light in weight, are reliable. By adapting Markov Random Field (MRF) [24] and Conditional Random Field (CRF) [19] [16] [29], the first algorithms; superpixels, are created to be used in discovering the depth from a single image.

Recently, deep learning has been used in computer vision tasks and proven to be useful in obtaining satisfactory results. The Convolutional Neural Network (CNN) of deep learning was receiving much attention for handling computer vision applications, such as object recognition [11] [13] [30] and segmentation [6] [17] [12], due to the self-learning feature. A study [7] has proposed a framework that would be the first to integrate CNN for depth estimation through multi-scale CNN. However, the framework took a long time to produce a depth estimation for each image. Since then, many methods

S. F. A. Abuowaida and H. Y. Chan are with School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia. Emails: suhilaowida@student.usm.my and hychan@usm.my

based on CNN had been proposed, as reported in a study by Eigen et al. [7]. Several researchers [16] [31] suggested a framework that merged CNN and CRF methods. CNN would extract features from the image and CRF would provide the final result of the prediction. However, these algorithms had high computation time and issues with the inferences.

The process of creating a CNN that contained multiple layers was also complicated due to the emergence of gradient vanishing problem in the training process. Many studies were affected by the increase in the number of these layers in CNN, which demonstrated a positive effect of obtaining high performance [14] [10] [2] [8]. The ResNet architecture designed by He et al. [11] was adopted in a study by Laina et al. [14] to observe up-sampling blocks for depth estimation, which found that the property related to translating invariance that existed in deep CNN may adversely affect the process of depth estimation. This issue, however, could be overcome by a skip connection technique [5] [2] [23]. The researcher in [2] utilizes the DenseNet [13] to determine the depth of a single image. Despite that, this algorithm was needing a long computation time, due to adopting DenseNet [13].

Hence, this paper intends to use a novel depth estimation architecture that can discern depth estimation from a single RGB image to improve the accuracy of depth estimation and reduce the number of parameters, which affects reduced computation time and is efficient even with the use of a huge dataset. This architecture consists of many stages, whereby firstly an encoder-decoder architecture has been improved from DenseNet [13] to deal with some implied problems in DenseNet that still exist, such as recognition of layers that have failed to have enough training, recognition of layers that have had more focused training and the use of a large filter size for the first convolution layer. Then, the standard loss function can be observed and adopted. The evaluation of depth estimation tends to be carried out by employing four types of measurements, which are average relative error (rel), root mean squared error (rms), average (log10) error and threshold accuracy. The algorithms in the proposed architecture of this study were also compared to algorithms used in other studies, such as Eigen et al. [7], Laina et al. [14], Alhashim et al. [2], Hao et al. [10], Wang et al. [29], Ren et al. [23] and Carvalho et al. [5]. The results from the four types of algorithms in measurement are described before concluding the observations of this study.

2. METHODOLOGY

This paper deals with investigating depth estimation of a single RGB image using an end-to-end learning architecture that produces a direct mapping of RGB in depth, as shown in Figure 1.

2.1 Encoder-Decoder Architecture

Figure 1 shows the proposed encoder-decoder architecture for depth estimation from a single RGB image. Many researchers have argued that the performance of the CNN architecture may increase with the depth of the CNN architecture. Nevertheless, stacking many layers on the CNN architecture cannot guarantee improved performance of the network and may alternatively lead to a significant decrease in performance. This issue exists because of the gradient vanishing problem during the training phase [24], which happens when the CNN architecture is stacked with too many layers. By using the DenseNet, the vanishing problem can be avoided through a connection between the layers. However, the DenseNet has been found to disregard the activation layer during the backpropagation process, as there is no formula within the parameters of DenseNet that describes the changing process, which leads to reduced accuracy in the gradient formula. The formula used in the DenseNet may not ascertain the layers that need more training than others. The novel architecture proposed in this study has improved the DenseNet [13], as shown in Figure 2, by simplifying and analyzing forward and backward propagation. The new rules of the different parameters in the DenseNet are obtained based on the new gradient formula in determining the layer that needs more or reduced training. A filter size more suitable than DenseNet is also selected to extract high and low levels of the features from the input image and the reduction parameters requirement, which leads to a reduced computation time based on the Formula 2.

- **DenseNet Analysis**

The connection between the layers through the gradient formula [13] is the key to solve the gradient vanishing problem. However, there are challenges when directly inferring forward and backward

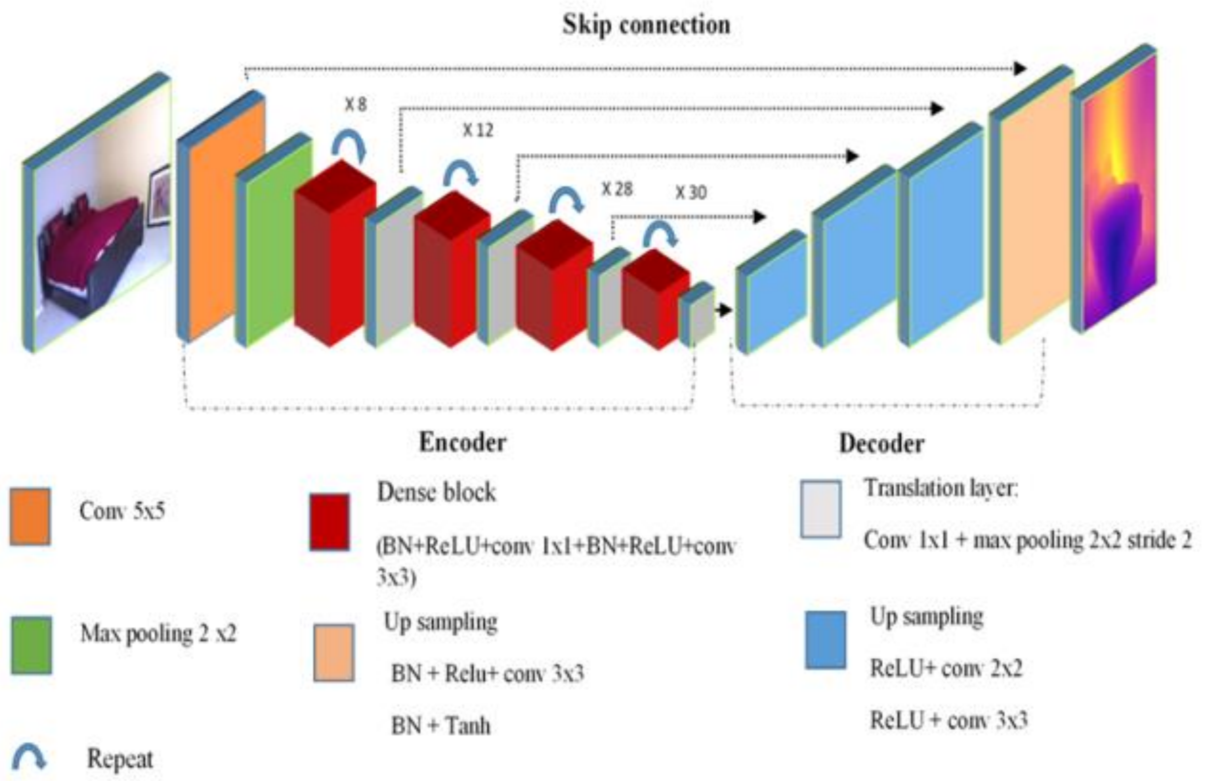

Figure 1. Encoder-decoder architecture for depth estimation from a single RGB image.

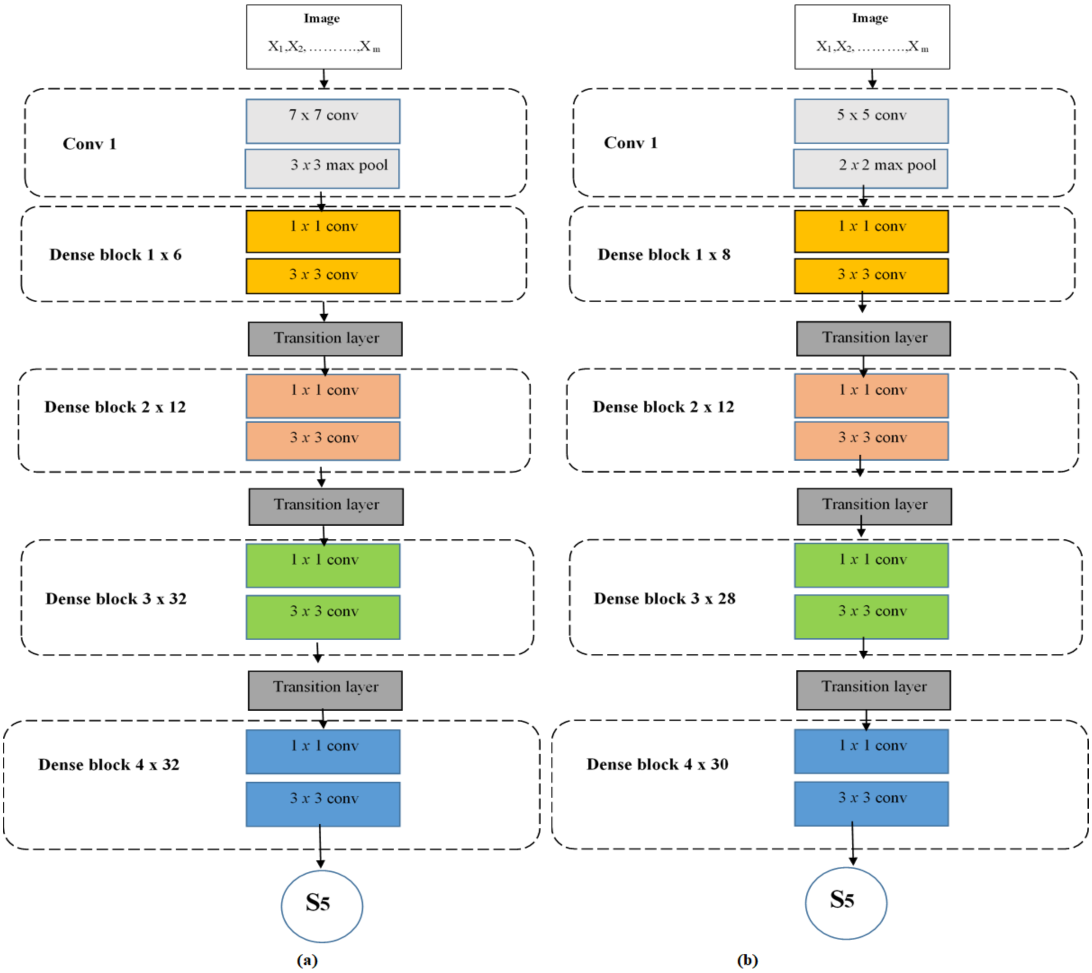

Figure 2. (a) DenseNet architecture 169 and (b) Improved DenseNet architecture.

propagation of DenseNet through gradient formula. Therefore, forward propagation and backward propagation of the DenseNet within a network addressing gradient vanishing are analyzed.

- **Forward Propagation Analysis**

The total loss function in the DenseNet is calculated using the square of the difference between the predicted output and the ground truth, as represented in the following formula:

$$L = \frac{1}{N}\sum_{i=1}^{c} |\,\hat{y}_i - y_i|^2 = \frac{1}{2}\sum_{i=1}^{c} |\,e_i|^2 \tag{1}$$

where N = The normalization term, y_i= The ground truth value, $\hat{y}_i$ = Predicted value, e_i= $y_i - \hat{y}_I$ and c = Number of classification layers.

Forward propagation, which is the first convolution layer the DenseNet, is represented by the following formula:

$$s_0 = \sum_{i=0}^{n} x_i . w_i \tag{2}$$

The dense block (DenseB) contains the h(.) function that has three operations: Batch normalization (BN), Rule layer and convolution kernel, which is a set of the first input layer [s_0, s_1, ..., s_{i-1}], where each layer receives the maps of the feature from all previous layers as input, as demonstrated by Formula 3.

$$DenseB = w_1 h_1(s_0) + w_2 h_2(s_0, s_1) + w_3 h_3(s_0, s_1, s_2) + w_4 h_4(s_0, s_1, s_2, s_3) \tag{3}$$

$$DenseB_i = \sum_{j=1}^{R}\sum_{i=1}^{4} w_i\, h_i(s) \tag{4}$$

where R is represents the repeated number of the dense block.
Then, the feature maps from the dense block input to the transaction layer are connected to the different dimensions through the following formula:

$$y_0 = \sum_{i=0}^{n} w_i \ .\theta(DenseB_i) \tag{5}$$

where θ(.) is the activation function and y_0 is the output from the first transaction layer. Forward propagation for the encoder uses the following formula:

$$y_j = \sum_{j=0}^{3}\sum_{i=1}^{N} w_i\, .\theta(DenseB_i) \tag{6}$$

where j= 1...3 (number of dense block in the architecture).

- **Back Propagation Analysis**

The predicted output of the simplified encoder is obtained from the weight of the last layer that employed backpropagation. The gradient, L, is represented in the following formulae.

$$\frac{\partial L}{\partial \omega_B} = \frac{\partial L}{\partial \hat{y}_B}\frac{\partial \hat{y}_B}{\partial \omega_B}$$

$$= \delta_B \frac{\partial\left(\Sigma_{i=1}^{4} \sum_{j=1}^{N} W_{B.}\theta(S_{Bi})\right)}{\partial \omega_B} = \delta_B . \theta(S_{Bi}) \tag{7}$$

where

$$\delta_{B_4} = \frac{\partial L}{\partial S_{B_4}}$$

$$= \frac{\partial \frac{1}{m}\Sigma_{\mathrm{i}=1}^{c}(\hat{y}_i - y_i)}{\partial S_{B_4}}$$

$$= e_i.\theta'\left(S_{B_4}\right) \tag{8}$$

The hidden layers of the encoder of this study have the same gradient formula:

$$\delta_i = \theta'\left(y_i.\sum_{i=0}^{3}\delta_{i+1}.w_{i+1}\right) \tag{9}$$

The gradient of the first connected weight fades as the number of layers increases in the network. This study defines $\Delta\delta^n$ as the gradient that increases based on the number of layers (n) in the encoder, as shown in the following formula.

$$\triangle\,\delta_i^n = \theta'(y_i^n)\sum_{i=0}^{c}\delta_i^{n+1}, \qquad n = 1,\ldots,4 \tag{10}$$

$\triangle\,\delta_i^n$ solved the vanishing problem in a deep network.

Skip connection technique is used to connect the encoder and decoder transferring the features of the maps to the decoder during the up-sampling process for depth estimation, which tends to speed up the learning of context awareness and overcome the translation invariance. The decoder in this study uses bi-linear up-sampling, as shown in Figure 1, where the up-sampling block utilizes ReLU for activation and convolution of the layers.

2.2 Loss Function

In this study, the encoder-decoder architecture has adopted various loss functions as represented in the following formulae:

$$L_{meanabsolute}(L_1) = \frac{1}{N}\sum_{i=1}^{s}|\,\hat{y}_i - y_i|\tag{11}$$

$$L_{meansquare}(L_2) = \frac{1}{N}\sum_{i=1}^{s}(\hat{y}_i - y_i)^2\tag{12}$$

$$L_{huber} = \begin{cases} L_1(l_i) & L_1(l_i) \geq c, \\ \dfrac{L_2(l_i) + c^2}{2c} & else \end{cases}\tag{13}$$

$$L_{berhub} = \begin{cases} L_1(l_i) & L_1(l_i) \leq c, \\ \dfrac{L_2(l_i) + c^2}{2c} & else \end{cases}\tag{14}$$

where yi= The ground truth value, yˆi = Predicted value, s = Number of classification layers, N = Normalization term, $c = \frac{1}{5}\max(|\hat{y}_i - y_i|)$ and i = Index value of pixel for each depth image in the current batch.

3. RESULTS AND DISCUSSION

The experimental results and evaluation of the algorithms presented have been analysed and interpreted. The evaluation of the encoder-decoder architecture has employed the following four measurements:

1) Average relative error (rel) =

$$\frac{1}{n}\sum_{p}^{n}\frac{|y_i - \hat{y}_i|}{y}\tag{15}$$

2) Root mean squared error (rms) =

$$\sqrt{\frac{1}{n}\sum_{p}^{n}(y_i - \hat{y}_i)^2} \qquad (16)$$

3) Average ($\log_{10}$) error =

$$\frac{1}{n}\sum_{p}^{n} |\log_{10} y_i - \log_{10} \hat{y}_i| \qquad (17)$$

4) Threshold accuracy =

$$\max\left(\frac{y_i}{\hat{y}_i}, \frac{\hat{y}_i}{y_i}\right) = \delta < threshold = 125{,}125^2, 125^3 \qquad (18)$$

where y_i= The ground truth value, $\hat{y}_i$ = Predicted value and n= Total value of pixel for each depth image.

3.1 Experimental Specifications

The architecture implemented in this study is built using TensorFlow [11], Amazon Web Services (AWS) and Amazon Machine Image (AMI), whereby the GPU-Us-Tesla V100 was at 16GB, while the VCPUs-8 cores had 61GB.

The algorithm of optimization used in is Stochastic Gradient Descent (SGD) [13]. The weight decay is 0.0001, with the learning momentum at 0.9 and the learning rate at 0.001 for 20 epochs.

3.2 Dataset

The quality of depth estimation has been evaluated using the NYU Depth v2 benchmark [25], which is considered one of the most well-known datasets for RGB single-image depth estimation. This dataset contains 1449 densely labeled pairs of images from indoor scenes with depth, 464 new scenes and 407,024 new unlabeled images that have been captured using Microsoft Kinect. Based on previous works that employed the NYU Depth v2 benchmark in examining depth estimation [7] [14] [2], the standard training and testing split was used to evaluate 654 image-depth pairs from the set.

3.3 Backbone Result

The proposed network has been chosen through the specific number of duplicates for each dense block based on the acquired results, as shown in Table 1. A selection of suitable duplicates for each layer of the dense block is carried out to improve the performance of the backbone network after the filter size is decreased.

Table 1. Number of duplicates for each dense block.

Block of dense layers	Number of duplicates	↓ *rel*
1	**1**,1,1,1,1	0.6630
	2,1,1,1,1	0.6656
	3,1,1,1,1	0.6705
2	1,2,1,1,1	0.5410
	1,4,1,1,1	0.5403
	1,6,1,1,1	0.5333
	1,**8**,1,1,1	0.5200
	1,10,1,1,1	0.5170
3	1,8,2,1,1	0.4801
	1,8,4,1,1	0.4711
	1,8,6,1,1	0.4287
	1,8,8,1,1	0.4122
	1,8,**12**,1,1	0.4036
	1,8,12,2,1	0.3885

4	1,8,12,4,1	0.3806
	1,8,12,6,1	0.3705
	1,8,12,8,1	0.3674
	1,8,12,10,1	0.3506
	1,8,12,12,1	0.3302
	1,8,12,14,1	0.3089
	1,8,12,16,1	0.2883
	1,8,12,18,1	0.2615
	1,8,12,20,1	0.2505
	1,8,12,22,1	0.2366
	1,8,12,24,1	0.2307
	1,8,12,26,1	0.2277
	1,8,12,**28**,1	0.2186
	1,8,12,30,1	0.2185
5	1,8,12,28,2	0.2092
	1,8,12,28,4	0.2025
	1,8,12,28,6	0.2003
	1,8,12,28,8	0.1945
	1,8,12,28,12	0.1920
	1,8,12,28,14	0.1858
	1,8,12,28,16	0.1789
	1,8,12,28,18	0.1603
	1,8,12,28,20	0.1552
	1,8,12,28,22	0.1501
	1,8,12,28,24	0.1483
	1,8,12,28,26	0.1382
	1,8,12,28,28	0.1241
	1,8,1228,,**30**	0.1220
	1,8,12,28,32	0.1220

Based on Table 1, the most suitable duplicates for each dense block are as follows:

1) One-time repetition of the first convolution layer.
2) Eight-time repetition of the second dense block.
3) Repetition of the third dense block 12 times.
4) Repetition of the fourth dense block 28 times.
5) Repetition of the fifth dense block 30 times.

As shown in Table 1, it is found that some dense blocks need to be repeated more to obtain better results (second dense block), while some dense blocks do not need duplicates of training. So we have reduced the number of repetition of training for these dense blocks (fourth dense block and fifth dense block), because it not needed to repeat the training, which leads to reduce parameters and then leads to reduce computation time.

3.4 Loss Function

The various loss functions of the encoder-decoder architecture tend to be compared using mean absolute, mean square, Huber and BerHub. Table 2 shows the results of this comparison.

As shown in Table 2, the performance of the loss function using BerHub is found to be the best for the different measurements used, which are rel error, rms error, $\log_{10}$ error, $\theta < 1.25$ accuracy, $\theta < 1.25^2$ accuracy and $\theta < 1.25^3$ accuracy. The results for the architecture are 0.1220, 0.4584, 0.0531, 0.8525, 0.9735 and 0.9946, respectively per sequence of measurements, as previously mentioned.

The loss function using Berhub is also found to be balanced between ground truth depth map values. When the differences between the ground truth depth map values are small, there will not be big differences in the weights and the dependence in this case is on L_1, but when the differences between

Table 2. Performance of various loss functions.

Loss function	↓ rel	↓ rms	↓ $\log_{10}$	↑ $\theta < 1.25$	↑ $\theta < 1.25^2$	↑ $\theta < 1.25^3$
Mean absolute	0.1367	0.5809	0.0582	0.8214	0.9670	0.9926
Mean square	0.1308	0.4820	0.0557	0.8408	0.9708	0.9934
Huber	0.1357	0.4949	0.0563	0.8363	0.9724	0.9942
BerHub	**0.1220**	**0.4584**	**0.0531**	**0.8525**	**0.9735**	**0.9946**

the ground truth depth map values are large, we tend to choose the equation $\frac{L_2(l_i)+c^2}{2c}$ and this equation is meant to reduce the loss in weights, which leads to a convergence between $(\hat{y}_i, y_i)$ so that we can get the best distribution of the ground truth depth map values. Hence, the loss function using Berhub is also found to be more appropriate for the architecture proposed in this study because of the small residuals that are utilized in the training stage, which has resulted in a better weight adjustment to achieve a better result.

3.5 Encoder-Decoder Comparison with Various Backbones

Table 3 presents the results of a comparison between various implementations to the backbone.

Table 3. Performance of various implementations of backbone.

Backbone	↓ rel	↓ rms	↓ $\log_{10}$	Parameters	Computation time per hour
DenseNet 121	0.1312	0.4970	0.0571	**21.2M**	**16.28**
DenseNet 169	0.1281	0.4740	0.0551	47.0M	21.13
DenseNet 201	0.1289	0.5515	0.0537	55.9M	26.34
ResNet50	0.1571	0.5590	0.0672	49.5M	30.55
ResNet101	0.1441	0.5559	0.0687	68.5M	40.23
Ours	**0.1220**	**0.4584**	**0.0531**	44.3M	19.31

The performance of the proposed backbone was found to exhibit better results. The rel error, rms error and $\log_{10}$ error accuracy of our backbone network amounted to 0.1220, 0.4584 and 0.0531. These values were significantly higher compared to other backbone values. The proposed backbone network solved the issue of vanishing gradient through identity shortcut based on a new gradient formula while taking the efficiency training for each dense block into account, as shown in Table 1. This aspect was considered through a specific duplicate increase or decrease in training and the reduction in the filter size to extract a certain amount of features from the input image and transfer this amount to other layers. The benefit obtained from the features led to positive results compared to other algorithms. The proposed backbone has consumed 44.3M parameters and 19.31 computation time per hour, making it the second in terms of the parameters and computation time per hour after DenseNet 121 backbone due to the increased repetition for some dense blocks to obtain better results, despite the high-speed characterization of the DenseNet 121 backbone algorithms in this process. Simultaneously, this is due to accuracy in terms of rel error, rms error and $\log_{10}$ error.

3.6 Comparison with the State of the Art Architecture

The results from this study are compared to those of studies conducted by Eigen et al. [7], Laina et al. [14], Al- hashim et al. [2], Hao et al. [10], Wang et al. [29], Ren et al. [23] and Carvalho et al. [5]. Table 4 shows the results of these comparisons. As shown in Table 4, the performance of the proposed architecture in this study obtained better results compared to other types of architectures when calculated using the different measurements; rms error, $\log_{10}$ error, $\theta < 1.25$ accuracy, $\theta < 1.25^2$ accuracy and $\theta < 1.25^3$ accuracy. The results of the proposed architecture are 0.4584, 0.0531, 0.8525, 0.9946, respectively, per the sequence of measurement previously mentioned. This result has suggested that the framework of the new backbone proposed in this study is extremely crucial in achieving desirable results. The proposed network has taken into account the efficiency of training for each dense block by increasing specific duplicates or decreasing training. Moreover, the filter size is decreased to obtain the number of features extracted from the input image so as to be fed to other layers. These steps have resulted in a better performance compared to other algorithms. Besides, this study has adapted the various loss functions and chosen the most suitable loss function for the proposed architecture, as shown in Table 2. The visual experimental results obtained from this study are clearly shown in Figure 3. The visuals illustrated in sequence are: the original RGB image, the depth prediction and the map of ground truth

Table 4. Performance of state-of-the-art architectures on NYU Depth v2 dataset.

Architecture	↓ *rel*	↓ *rms*	↓ $\log_{10}$	↑ $\theta < 1.25$	↑ $\theta < 1.25^2$	↑ $\theta < 1.25^3$
Eigen et al.[7]	0.158	0.641	-	0.769	0.950	0.988
Laina et al. [14]	0.127	0.573	0.055	0.811	0.953	0.988
Alhashim et al.[2]	0.123	0.465	0.053	0.846	**0.974**	0.994
Hao et al. [10]	0.127	0.555	0.053	0.841	0.966	0.991
Wang et al. [29]	0.220	0.745	0.094	0.605	0.890	0.970
Carvalho et al. [5]	0.135	0.600	0.059	0.819	0.957	0.987
Ren et al. [23]	**0.113**	0.501	-	0.833	0.968	0.993
Ours	0.122	**0.458**	**0.052**	**0.853**	**0.974**	**0.995**

.

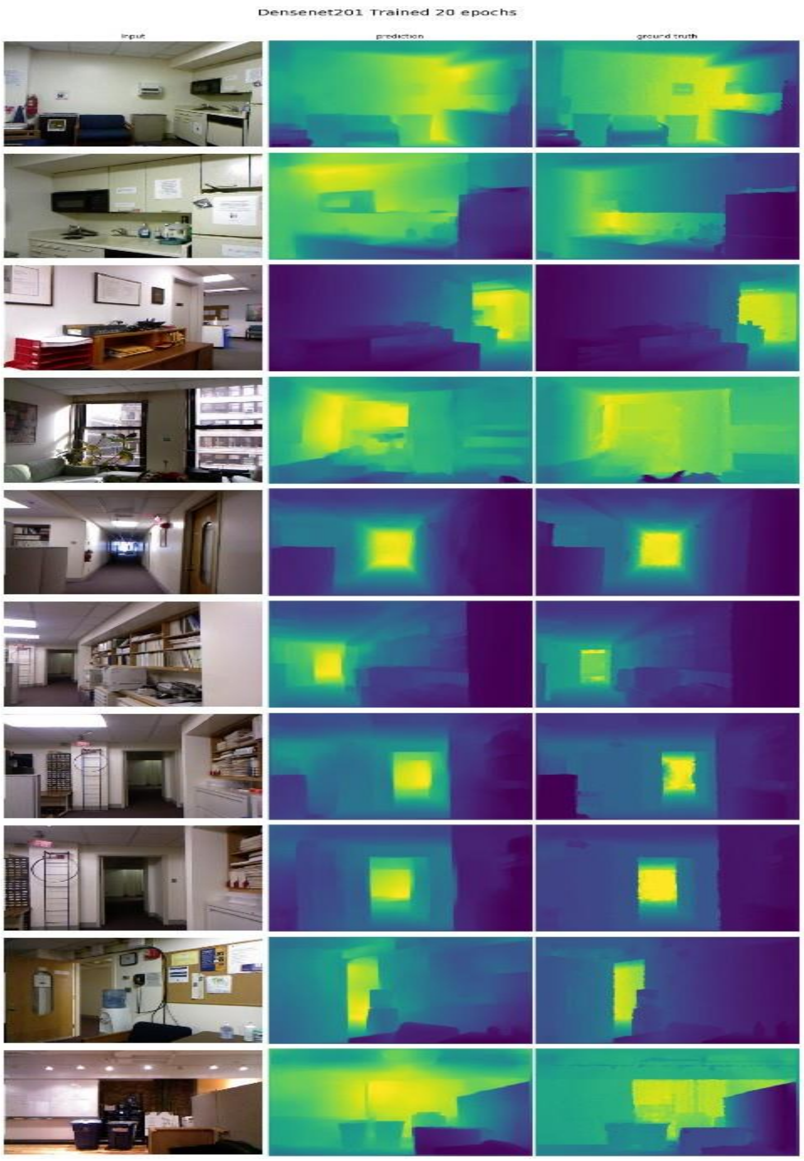

Figure 3. The visual experimental results of our encoder- decoder architecture from NYU Depth v2 dataset.

4. Conclusion

In this study, a novel encoder-decoder architecture has been proposed to investigate depth estimation from a single RGB image. There are several stages in this architecture, whereby firstly the encoder-decoder architecture has been simplified from the DenseNet and analyzed using forward and backward propagation. Then, the new rules for the different parameters of the DenseNet are obtained based on the new gradient formula to search for the layer that needs more or less training. The filter size that is selected to extract high and low feature levels from the input image for solving the gradient vanishing problem should be more suitable than DenseNet. Results from this study on the NYU Depth v2 dataset have also shown that the loss function using Berhub has produced the best performance. Experiments on the NYU Depth v2 dataset further demonstrated that the encoder-decoder architecture in this study has achieved a state-of- the-art performance based on the consistent performance in obtaining depth estimation from a single RGB image.

References

[1] A. Abrams, C. Hawley and R. Pless, "Heliometric Stereo: Shape from Sun Position," European Conference on Computer Vision, Part of the Lecture Notes in Computer Science Book Series (LNCS), Vol. 7573, pp. 357–370, Springer, 2012.

[2] I. Alhashim and P. Wonka, "High Quality Monocular Depth Estimation *via* Transfer Learning," arXiv: 1812.11941v2, [Online], Available: https://arxiv.org/pdf/1812.11941.pdf, 2018.

[3] A. Atapour-Abarghouei and T. P. Breckon, "Real-time Monocular Depth Estimation Using Synthetic Data with Domain Adaptation *via* Image Style Transfer," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE/CVF), pp. 2800–2810, Salt Lake City, UT, USA, 2018.

[4] T. Bebie and H. Bieri, "A Video-based 3D-Reconstruction of Soccer Games," Computer Graphics Forum, vol. 19, no. 3, pp. 391–400, DOI: 10.1111/1467-8659.00431, 2000.

[5] M. Carvalho et al., "On Regression Losses for Deep Depth Estimation," Proc. of the 25th IEEE International Conference on Image Processing (ICIP), pp. 2915–2919, Athens, Greece, 2018.

[6] J. Dai, K. He and J. Sun, "Instance-aware Semantic Segmentation *via* Multi-task Network Cascades," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3150–3158, Las Vegas, NV, USA, 2016.

[7] D. Eigen, C. Puhrsch and R. Fergus, "Depth Map Prediction from a Single Image Using a Multi-scale Deep Network," Advances in Neural Information Processing Systems, arXiv: 1406.2283v1, pp. 2366–2374, [Online], Available: https://arxiv.org/pdf/1406.2283.pdf, 2014.

[8] H. Fu et al., "Deep Ordinal Regression Network for Monocular Depth Estimation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE/CVF), pp. 2002–2011, Salt Lake City, UT, USA, 2018.

[9] A. Grigorev et al., "Depth Estimation from Single Monocular Images Using Deep Hybrid Network," Multimedia Tools and Applications, vol. 76, no. 18, pp. 18585–18604, 2017.

[10] Z. Hao et al., "Detail Preserving Depth Estimation from a Single Image Using Attention Guided Networks," Proc. of the IEEE International Conference on 3D Vision (3DV), pp. 304–313, Verona, Italy, 2018.

[11] K. He et al., "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, Las Vegas, NV, USA, 2016.

[12] K. He et al., "Mask R-CNN," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2961–2969, Venice, Italy, 2017.

[13] G. Huang et al., "Densely Connected Convolutional Networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708, Honolulu, HI, USA, 2017.

[14] I. Laina et al., "Deeper Depth Prediction with Fully Convolutional Residual Networks," Proc. of the 4th IEEE International Conference on 3D Vision (3DV), pp. 239–248, Stanford, CA, USA, 2016.

[15] W. Lee, N. Park and W. Woo, "Depth-assisted Real-time 3D Object Detection for Augmented Reality," Proc. of the 21st International Conference on Artificial Reality and Telexistence, (ICAT), vol. 11, no. 2, pp. 126–132, Osaka, Japan, 2011.

[16] B. Li et al., "Depth and Surface Normal Estimation from Monocular Images Using Regression on Deep Features and Hierarchical CRFs," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVRP), pp. 1119–1127, Boston, MA, USA, 2015.

[17] Y. Li et al., "Fully Convolutional Instance-aware Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVRP), pp. 2359–2367, Honolulu, HI, USA, 2017.

[18] F. Liu et al., "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 10, pp. 2024–2039, 2015.

[19] M. Liu, M. Salzmann and X. He, "Discrete-continuous Depth Estimation from a Single Image," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 716–723, Columbus, OH, USA, 2014.

[20] Y. Liu et al., "Continuous Depth Estimation for Multi-view Stereo," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2121–2128, Miami, FL, USA, 2009.

[21] P. K. Martin et al., "Improved Depth Map Estimation from Stereo Images Based on Hybrid Method," RadioEngineering Journal, vol. 21, no. 1, pp. 70-78, 2012.

[22] F. Qi et al., "Structure Guided Fusion for Depth Map Inpainting," Pattern Recognition Letters, vol. 34, no. 1, pp. 70–76, 2013.

[23] H. Ren, M. El-Khamy and J. Lee, "Deep Robust Single Image Depth Estimation Neural Network Using Scene Understanding," Computer Vision and Pattern Recognition Workshops, arXiv: 1906.03279v1, [Online], Available: https://arxiv.org/pdf/1906.03279.pdf, pp. 37–45, 2019.

[24] A. Saxena, M. Sun and A. Y. Ng, "Make3D: Learning 3D Scene Structure from a Single Still Image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 824–840, 2008.

[25] N. Silberman et al., "Indoor Segmentation and Support Inference from RGBD Images," Proc. of European Conference on Computer Vision, Part of the Lecture Notes in Computer Science Book Series (LNCS), vol. 7576, pp. 746–760, Springer, 2012.

[26] F. Simões et al., "Challenges in 3D Reconstruction from Images for Difficult Large-scale Objects: A Study on the Modeling of Electrical Substations," Proc. of the 14th IEEE Symposium on Virtual and Augmented Reality, pp. 74–83, Rio de Janiero, Brazil, 2012.

[27] R. Szeliski, Computer Vision: Algorithms and Applications (Texts in Computer Science), 2011 Edition, Springer, 2011.

[28] M. W. Tao et al., "Depth from Shading, Defocus and Correspondence Using Light-field Angular Coherence," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1940–1948, Boston, MA, USA, 2015.

[29] P. Wang et al., "Towards Unified Depth and Semantic Prediction from a Single Image," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2800–2809, Boston, MA, USA, 2015.

[30] Y. C. Wong et al., "Deep Learning-based Racing Bib Number Detection and Recognition," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 181-194, 2019.

[31] D. Xu et al., "Multi-scale Continuous CRFs As Sequential Deep Networks for Monocular Depth Estimation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5354–5362, Honolulu, HI, USA, 2017.

[32] H. Xu, Y. Cai and R. Wang, "Depth Estimation in Multi-view Stereo Based on Image Pyramid," Proceedings of the 2nd International Conference on Computer Science and Artificial Intelligence, pp. 345–349, [Online], Available: https://doi.org/10.1145/3297156.3297238, 2018.

[33] S. Zachow, M. Zilske and H.-C. Hege, "3D Reconstruction of Individual Anatomy from Medical Image Data: Segmentation and Geometry Processing," Proc. of the 25th ANSYS Conference & CADFEM Users' Meeting, Proc. CD 2.12.15, ZIB-Report, pp. 7-41, ISSN: 1438-0064, Dresden, Germany, 2007.

ملخص البحث:

جلبت الفوائد العديدة لتقدير العُمق من مجال صورةٍ واحدةٍ في الطّبّ وألعاب الفيديو الخاصة بالروبوت وتطبيقات الواقع ثلاثية الأبعاد الكثير من الاهتمام في السنوات الأخيرة. ولعلاقتها الوثيقة بالبعد الثالث المتمثل في العُمق، يمكن إنجاز هذه العملية باستخدام رؤية الإنسان؛ نظراً لاعتبارها محفوفةً بالتحديات بسبب قضايا متنوعة عند استخدام رؤية الحاسوب. فالاختلافات في الهندسة، ونسيج المشهد، والحدود الخاصة بانسداد المشهد، والغموض المتأصّل؛ كلها قضايا موجودة بفعْل ضآلة المعلومات التي يمكن الحصول عليها من صورةٍ مفردةٍ.

لذلك، فإنّ هذه الورقة تقترح طريقةً مبتكرةً لتقدير العُمق في مجال المعمارية، تتضمن المراحل التي يمكنها أن تقوم بتقدير العُمق من صورةٍ مفردةٍ بالأحمر والأخضر والأزرق (RGB). ولقد تمّ اقتراح معمارية مؤلفة من وحدة ترميز وأخرى لفكّ الترميز، بناءً على التحسين الذي تم الحصول عليه من (DenseNet)؛ إذ جرى استخلاص خريطة للصورة باستخدام تقنية التوصيل القائمة على التّخطّي. ومن ناحيةٍ أخرى، تعتمد هذه الورقة دالّة الفقْد العكسية لهوبر (Huber) التي تلائم بصورة أساسية المعمارية المقترحة من خلال توزيعات القيم التي تُوجد بشكلٍ عام في خرائط العُمق.

وقد أشارت النتائج التجريبية الى أنّ معمارية تقدير العُمق التي توظِّف مجموعة البيانات (NYU Depth v2) كانت ذات أداءٍ أفضل مقارنة بالطرق الأخرى المستخدمة في مجال تقدير العُمق التي تميل الى امتلاك عددٍ أقلّ من المتغيرات وتتطلب زمن تدريبٍ أقصر.