CNN Based Monocular Depth Estimation

K. Swaraja*, K. Naga Siva Pavan, S. Suryakanth Reddy, K.Ajay, P.Uday Kiran Reddy, Padmavathi Kora, K. Meenakshi, Duggineni Chaitanya and Himabindu.Valiveti

GRIET, ECE Department, Hyderabad, India

Abstract. In several applications, such as scene interpretation and reconstruction, precise depth measurement from images is a significant challenge. Current depth estimate techniques frequently provide fuzzy, low-resolution estimates. With the use of transfer learning, this research executes a convolutional neural network for generating a high-resolution depth map from a single RGB image. With a typical encoder-decoder architecture, when initializing the encoder, we use features extracted from high-performing pre-trained networks, as well as augmentation and training procedures that lead to more accurate outcomes. We demonstrate how, even with a very basic decoder, our approach can provide complete high-resolution depth maps. A wide number of deep learning approaches have recently been presented, and they have showed significant promise in dealing with the classical ill-posed issue. The studies are carried out using KITTI and NYU Depth v2, two widely utilized public datasets. We also examine the errors created by various models in order to expose the shortcomings of present approaches which accomplishes viable performance on KITTI besides NYU Depth v2.

1 Introduction

Commercial depth sensors, such as different LiDAR systems including Kinects, are often used to acquire depth information. Besides the expensive cost and high level of operating expertise required, they also have the drawbacks of low resolution and limited perception distance, which restrict their application scope. Because RGB images are widely available, interpreting a depth map from monocular images is gaining popularity. However, because it is intrinsically scale ambiguous, an endless number of potential depth maps may be connected with one image, as it is a difficult ill-posed issue. Monocular image depth cues have been used as clues during depth estimate. Despite the fact that these strategies have been made known to be excellent in experiments, they are not frequently used because depth cues are not often present in natural images. Several researchers are trying to solve monocular depth prediction through implementing deep learning approaches, motivated by the remarkable performance of deep learning on image categorization, object identification, including semantic segmentation.

Early techniques depended solely on hand-crafted features including probabilistic models, whereas current approaches have largely relied on convolutional neural networks (CNNs) because of their superior performance. Many CNN based approaches are proposed by various authors to evaluate the depth of a monocular image. Saxena et al. [1] initially projected a Markov Random Field to envisage depth from a monocular horizontally-associated image, which then far ahead progressed into the Make-3D scheme. When the settings are out of control

this method fails to work, particularly when the horizontal arrangement state does not embrace. Eigen et al. [2] are the first to use CNN to guess the depth of one image. They trained a multi scale convolutional neural network (CNN) to measure depth using Deep Learning techniques for this type of challenge. Xiaobai Ma et al.[3] proposed that CNN based depth estimation methods began to dominate recently. Since the task is closely related to semantic labeling, most of the works are built upon ImageNet Large Scale Visual Recognition Challenge (ILSVRC), often initializing their networks with AlexNet or the deeper VGG. The authors addressed the task by employing two deep network stacks. The first network is used to predict the depth of the whole image, and the second network defines this prediction locally. This idea is later extended, where three stacks of CNN are used to additionally predict surface normals and labels together with depth.

Zhao Chen et al.[4] suggested a method for overcoming the limits of monocular depth estimation by giving a depth model a sparse amount of recorded depth and an RGB image to predict the entire depth map. The resultant model developed can provide comparable performance to a modern depth sensor, despite only observing a small fraction of the depth map. The design of smaller and more energy-efficient depth sensor hardware is preferred. Because of this challenge, many computer vision systems use RGB images obtained from video stream for camera pose estimation. Subsequently that, Liu et al.[5] presented a number of alternative CNN-based techniques. Roy et al.[6] have suggested a Neural Regression Forest-based depth estimation technique. The approaches outlined above, on the other hand, are particular to the

* Corresponding author: kswaraja@gmail.com

context in which they were trained and so are not domain independent. Monocular depth estimation relies on image processing techniques [7-25] and is helpful during communication [26-30].

The rest of the work is prearranged as follows. Section 2 illustrates the proposed methodology associated with the monocular depth estimation. Simulation results and analysis are provided in section 3 with different performance metrics for two publicly available datasets. Conclusion is given in section 4.

2 Proposed Methodology for Monocular Depth Estimation

We take a set of images from dataset of size 640x480 and resize them to 256x256. When these input images are passed through the encoder, it performs down sampling along with a few operations to get an image of size 8x8. Encoder output is then passed to the decoder. Now decoder up samples these images along with few operations to obtain the final predicted depth map images of size 256x256. In this project ResNet-50 is used as an encoder and bilinear-up sampling method as the decoder. Experiments were also conducted with EfficientNet and DenseNet architectures as the encoders, and the results are compared with ResNet. The following flowchart demonstrates the process in detail.

Fig 1. Flow Chart

- The database is the first block of encoder. It's where the input data is saved. For this project we used a database that comprises of 5000 images under the training set and 500 images under the testing set.
- Image Normalization is performed to convert input images into a range of pixel values.
- Data-Augmentation is performed on the dataset to increase diversity. Auto Augmentation is the technique that is used in the current project. Auto Augmentation is pre-learned with sub-policies on ImageNet.
- Later we resize the image to a double vector matrix. These operations are performed in the second block.
- The third block contains the neural network model. ResNet-50 or EfficientNet-B0 are the models used in the project.
- EfficientNet-B0 or ResNet-50 models are defined using TensorFlow and Keras. The transfer learning

approach is used in this project with pretraining performed on ImageNet.

- These are the blocks in encoder which down sample the input images and produce an image of size 8x8.
- Later, in decoder a custom Up sampling block is defined, this block consists of Bilinear up sampling layer along with convolutional and activation layers. The convolutional layers consist of skip connections from the encoder. These connections provide feature-rich correspondences.
- Here the images are up sampled and we get the depth map images.
- The model is trained in Google Colab with GPU support for 25 epochs. Each epoch takes around 5-6 minutes.
- Finally, Depth maps are predicted and the output is obtained.
- For performance and Mean Actual Error (MAE) evaluation, we used Structural Similarity Index Measurement, Jaccard, and F1-score metrics. For validation loss measurement, the test dataset is taken as the validation dataset.
- These are the algorithmic steps followed to obtain the required results.

3. Simulation Results and Analysis

3.1 DATASETS

3.1.1 KITTI Dataset

The KITTI dataset was introduced in IJRR in 2013. This dataset comprises outdoor scenes captured by cameras besides depth sensors in a driving car, with a dimension of 375×1242 image. This dataset has two versions encompassing RGB stereo sets and equivalent ground truth depth maps. The KITTI dataset contains genuine photos from the "city," "residential," and "road" categories. The 56 image scenes in the KITTI dataset are separated into 2 portions: 28 for training and 28 for testing. Stereo image pairs having a resolution of 1224 x368 are used to create each scene. We resize the Images to 256x256 during training. This dataset is moreover alienated into RD: KITTI Raw Depth; CD: KITTI Continuous Depth; SD: KITTI Semi-Dense Depth; ES: Eigen Split; ID: KITTI Improved Depth. It is customary to use the Eigen split instead of using the complete KITTI dataset. This split contains 23,488 photos for training and 697 images for testing from 32 scenes.

A revolving LIDAR sensor samples the corresponding depth of every RGB image in a sparse manner. We only evaluate the bottom half of the picture space since the lidar cannot give depth information for the upper half of the image space. The dataset is also commonly used to assess deep learning-based visual odometry (VO) algorithms because it includes the ground truth of posture for 11 odometry sequences. This dataset is frequently used in computer vision subtasks such as optical flow, visual odometry, depth, object detection, semantic segmentation, and tracking, among others. As this Dataset

is more complicated for the use in supervised and semi-supervised algorithms, we have used NYU-Dataset.

Fig 2. KITTI Dataset

3.1.2 NYU-Depth V2

Different from the KITTI dataset, The NYU-Depth V2 data set is composed of monocular video sequences from a variety of indoor scenes as taken by both the RGB and Depth cameras from the Microsoft Kinect. Nathan Silberman and Fergus introduced the NYU Depth dataset at ECCV 2012. This dataset consists of 1449 RGB images compactly labelled with depth images. The datasets comprise of 407K frames of 464 scenes captured from 3 different cities. This dataset focuses on indoor situations, and there are 464 indoor image scenes in this dataset. There are 249 training sequences and 215 testing sessions in the official split.

Fig 3 NYU Depth v2 dataset

It is the common benchmark and the main training dataset in the supervised monocular depth estimation. We randomly selected frames from each training sequence, yielding about 12k distinct images. Subsequent to offline augmentations, our dataset includes around 59k RGB-D image pairs. The original image resolution is 480×640 . We down sampled the images to 256×256 as our network input. Because of the varied frame rates of the RGB and depth cameras, there is no one-to-one correlation amongst depth maps and RGB images. To line up the depth maps and RGB images, each depth map is first paired with the RGB image that is nearest to it. The camera projections are then utilized to bring into line depth and RGB pairings using the geometrical connection supplied by the dataset. Because the projection is discrete, not all pixels have a depth value, hence the pixels with no depth value are

masked off during the trials. We conducted studies using these down sampled photos as input images.

3.2 Evaluation metrics

Evaluation metrics comprise error besides accuracy metrics which are proposed in five evaluation indicators such as absolute relative error (ARE), root mean square error (RMSE), logarithm root mean square error (RMSE log) and square relative error (SRE).

These parameters are defined as follows:

$$ARE = \frac{1}{k} \sum \frac{|a_i - a_i^*|}{a_i} \quad (1)$$

$$RMSE = \sqrt{\frac{1}{k} \sum |a_i - a_i^*|^2} \quad (2)$$

$$RMSE(\log) = \sqrt{\frac{1}{k} \sum |\log a_i - \log a_i^*|^2} \quad (3)$$

$$SRE = \frac{1}{k} \sum \frac{|a_i - a_i^*|^2}{a_i} \quad (4)$$

Accuracy with threshold ($\delta < th$): % of a_i such that $\max(\frac{a_i}{a_i^*}, \frac{a_i^*}{a_i}) < th$; where $th = 1.25, 1.25^2, 1.25^3$. Where a_i

and a_i^* are the ground truth and foreseen depth at pixel i , and k is the entire number of pixels with real-depth values, and th signifies the threshold. These evaluation metrics were used to measure the quality of our model after the learning process.

3.3 Discussions and comparative analysis

Table 1 depicts the model parameters and their respective values during training the proposed framework. Table 2 provides the Mean Absolute error (MAE) for different epochs with 400 homogeneous input images only, while Table 3 provides the Mean Absolute error (MAE) for different epochs with 4600 heterogeneous input images. The average error metrics over the entire test dataset for architectures is provided in Table 4. Thus from these tables it is found that EfficientNet architecture has minimum loss values and is more accurate compared to ResNet architecture.

Table 1 Model parameters and their corresponding values

Model Parameters	Value
Learning rate	0.0002
No.of Epochs	50
Loss function	customized
Optimizer	Adam
Batch size	10

Table 2. Mean Absolute Error comparison (400 homogeneous images)

Epochs	ResNet-50	EfficientNet-B0
5	0.5339	0.2779
10	0.4093	0.2675
15	0.3353	0.2009
20	0.2789	0.1872
25	0.2581	0.1234
50	0.2044	0.1075

Fig 4. MAE plot for EfficientNet-B0 (400 homogeneous input images)

Table 3. Mean Absolute Error (heterogeneous input images)

Epochs	ResNet-50	EfficientNet-B0
5	0.3365	0.3587
10	0.2366	0.2465
15	0.1842	0.1985
20	0.1597	0.1635
25	0.1413	0.1553

Fig.5 MAE plot for EfficientNet-B0 (4600 heterogeneous input images)

Fig.6 Training and validation loss for EfficientNet-B0 (400 homogeneous input images)

Fig.7 Training and validation loss for EfficientNet-B0 (4600 heterogeneous input images)

Fig 4 shows the MAE values for different epochs when executed with EfficientNet-B0 for 400 homogeneous input images. Fig 5 shows the MAE values for different epochs when executed with EfficientNet-B0 for 4600 heterogeneous input images. Fig.6 shows the Training and validation loss for EfficientNet-B0 for 400 homogeneous input images, whereas Fig.7 depicts the Training and validation loss for EfficientNet-B0 for 4600 heterogeneous input images. It is found that EfficientNet architecture has minimum loss values and is more accurate compared to ResNet architecture.

Table 4. The average error metrics over the entire test dataset for architectures

Architectures	REL	RMSE _{lin}	RMSE _{log}
ResNet-50	0.168	0.638	0.069
EfficientNet-B0	0.156	0.625	0.061

3.4 Results Analysis

During execution, we used 400 homogeneous images as input and also 4600 heterogeneous images as input for the encoder and compared with both the ResNet encoder and EfficientNet encoder to know which is the best. Fig. 8 shows the original images and their corresponding ground

truth and depth maps results obtained with the EfficientNet model for 400 images while Fig 9 shows the same for **ResNet model**. Fig. 10 shows the original images and their corresponding ground truth and depth maps results obtained with the EfficientNet model for 4600 images while Fig 11 shows the same for **ResNet model**. On the basis of training and validation losses and MAE, we compared the outcomes and observed that though the input is 400 homogenous or 4600 heterogeneous images, the EfficientNet design produces more accurate results.

Fig. 8 shows the (a)original images and their corresponding (b)ground truth and (c)depth maps results obtained with the EfficientNet model for 400 images

Fig.9 shows the (a)original images and their corresponding (b)ground truth and (c)depth maps results obtained with the ResNet model for 400 images

Fig.10 shows the (a)original images their corresponding (b)ground truth and (c)depth maps results obtained with the EfficientNet model for 4600 images

Fig.11 shows the (a)original images their corresponding (b)ground truth and (c)depth maps results obtained with the ResNet model for 4600 images

4. Conclusion

Monocular depth estimation plays a critical role in scene understanding. From training approaches to task categories, this work introduces relevant deep learning frameworks. In this work monocular depth estimation utilizing encoder and decoder architecture is proposed.

The encoder, EfficientNet-B0 or ResNet-50, is a low-weight network, and Bilinear Interpolation is a decoder network. This decoder helps in the enhancement of model parameters. Since resolution scaling is one of the essential features of EfficientNet-B0, the results confirm that EfficientNet-B0 would be more competent for high resolution images. When tested on 5000 images, we found that EfficientNet-B0 yields reduced validation loss in contrast to other frameworks due to varying scaling. When the input resolution is higher, EfficientNet has a lower rate of loss than ResNet50.

References

1. Saxena, A., Schulte, J. and Ng, A.Y., *IJCAI*, **7**, pp. 2197-2203 (2007).
2. Wan, L., Eigen, D. and Fergus, R., *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 851-859) (2015).
3. Ma, X., Geng, Z. and Bie, Z., *Semantic Scholar* (2017).
4. Chen, Z., Badrinarayanan, V., Drozdov, G. and Rabinovich, A., *European Conference on Computer Vision (ECCV)* (pp. 167-182) (2018).
5. Liu, F., Shen, C. and Lin, G., *IEEE conference on computer vision and pattern recognition* (pp. 5162-5170) (2015).
6. Roy, A. and Todorovic, S., *IEEE conference on computer vision and pattern recognition* (pp. 5506-5514) (2016).
7. Kuraparthi, S., Kollati, M., Kora, P. *Traitement du Signal*, **36(6)**, 565-573 (2019).
8. Swaraja, K., *J. Adv. Res. Dyn. Control Syst.(JARDCS)*, Special, (11) (2017).
9. Swaraja, K., Madhaveelatha, Y. and Reddy, V.S.K., *Int J Appl Eng Res*, **11(1)**, 216-225 (2016).
10. Swaraja, K., Latha, Y.M., Reddy, V.S.K. and Paramkusam, A.V., *2011 Annual IEEE India Conference*, (pp. 1-4) (2011).
11. Swaraja, K., Madhaveelatha, Y. and Reddy, V.S.K., *IEEE International Conference on Computational Intelligence and Computing Research*, (pp. 1-4) (2014).
12. Swaraja, K., Madhaveelatha, Y. and Reddy, V.S.K., *ARNP Journal of Engineering and Applied Sciences*, **10(4)**, pp.1822-1827 (2015).
13. Kora, P., Meenakshi, K., Swaraja, K., Rajani, A., & Islam, M. K., *Informatics in Medicine Unlocked*, **17**, 100257 (2019).
14. Kora, Padmavathi, and K. Sri Rama Krishna, *Information systems design and intelligent applications*, Springer, pp. 309-317, (2016).
15. Kumari, C.U., Kora, P., Meenakshi, K., Swaraja, K., Padma, T., Panigrahy, A.K. and Vignesh, N.A., *International Conference on Innovative Computing and Communications*, Springer (pp. 425-433), (2020).
16. Kumari, C.U., Kora, P., Meenakshi, K. and Swaraja, K., *3rd International Conference on Computing Methodologies and Communication (ICCMC)*, (pp. 114-117), (2019).
17. Swaraja, K., Meenakshi, K. and Kora, P., *Biomedical Signal Processing and Control*, **68**, p.102688, (2021).
18. Sravan, V., Swaraja, K., Meenakshi, K., Kora, P. and Samson, M., *4th International Conference on Trends in Electronics and Informatics (ICOEI)*, (48184), (pp. 1063-1068), (2020)
19. Meenakshi, K., and GN Beena Bethel. *First International Conference on Networks & Soft Computing (ICNSC2014)*, (2014).
20. Meenakshi, K., Swaraja, K. and Kora, P., *Soft computing and signal processing*, Springer, Singapore, (pp. 477-485), (2019).
21. Kora, P. and Kalva, S.R., *SpringerPlus*, **4(1)**, pp.1-19 (2015).
22. Kumar, S.K., Reddy, P.D.K., Ramesh, G. and Maddumala, V.R., *Traitement du Signal*, **36**, pp.233-237 (2019).
23. Kumari, C.U., Prasad, S.J. and Mounika, G., *3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1095-1098) (2019).
24. Dhanalaxmi, B., Naidu, G.A. and Anuradha, K., *Procedia Computer Science*, **46**, pp.432-442 (2015).
25. Jayahari, L., Sasidhar, P.V., Reddy, P.P., BaluNaik, B., Gupta, A.K. and Singh, S.K., *Journal of King Saud University-Engineering Sciences*, **26(1)**, pp.21-31 (2014).
26. T. Pavani, A. Hemanth, I. S. Narayana and Y. R. Rao, *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 54-57 (2021).
27. Valiveti, H.B. and Polipalli, T.R., *International Journal of Computer Aided Engineering and Technology*, **10(5)**, pp.599-608, (2018).
28. Valiveti, H.B. and Kumar, B.A., *International Journal of Communication Systems*, **34(6)**, p.e4285, (2021).
29. H. B. Valiveti, S. Ganesh, B. A. Kumar and D. K. Yadav, *Computers, Materials & Continua*, **67(2)**, pp. 2049–2059, (2021).
30. Koduru, A., Valiveti, H.B. and Budati, A.K., *International Journal of Speech Technology*, **23(1)**, pp.45-55 (2020).