Monocular Depth Estimation from Single Image

Mahdi Abdollah Chalaki
University of Alberta
Edmonton, AB, CA
Abdolla2@ualberta.ca

Masoud Jafaripour
University of Alberta
Edmonton, AB, CA
jafaripo@ualberta.ca

Abstract

In this report, we will discuss what we have achieved in our project, including a discussion about our literature review, related works in this research area, various datasets, a baseline model implementation, improved models implementation and discussion.

1. Introduction

Depth estimation plays a vital role in today's computer vision tasks, such as self-driving cars, 3D reconstruction, augmented reality etc.[1] There are two mainstreams in depth estimation: active and passive approaches. The most frequently implemented approach is active methods which are using sensors and structured light cameras. This equipment basically belongs to illuminating coded signals technique. The passive approach is often based on multi-view geometry, such as stereo and binocular depth estimation methods. Stereo depth estimation is usually achieved by using images which come from two camera to triangulate and estimate distances. Outstanding stereo vision-based systems have been developed by researchers during last few decades. Despite stereo vision-based systems performance in many environments, they suffer from limitation on baseline distance between two cameras to remain accurate. Otherwise, very small error on camera calibration cause very large error in depth. So, they always depend on lots of calibrated equipment. Furthermore, stereo vision always fails in regions without any texture, since they cannot find reliable correspondence between pairs of images.[2]

On the other hand, any frame of an image contains many monocular visual cues including texture gradients, de-focus, color/haze, etc. that have not been used in stereo vision methods. Humans understand depth by combing many of stereo and monocular cues. Monocular depth estimation is difficult since it needs to take account global structure of image. Although there are many works on stereo depth estimation, monocular side is an open problem with many challenges including better understanding numerous images around us in both indoor and outdoor scenes.

There are many reasons that stereo vision has been more investigated than monocular case. Given accurate image correspondences, stereo vision can recover depth deterministically. So, stereo vision can be summarized just to developing robust image point correspondences. Contrary, monocular depth estimation require to use cues such as perspective, line angle, object size, image position, and atmospheric effects.[3] But the monocular depth estimation does not require rectified images and additional equipment (for example multiple calibrated camera), So, demand for monocular depth estimation increased significantly in recent years.[2]

In this paper, we present existing approaches for monocular depth estimation from a single image with focus on deep learning methods. We define basic tools for monocular depth estimation such as metric for evaluation, loss functions, most used datasets, and a general network architecture of convolutional neural network. Then, more advanced algorithms including transfer learning and transformers are elaborated. Hereafter, results of each algorithm on famous datasets are provided. Finally, depth map results of implementing two most advanced method on photos are shown.

2. Related works

Many reasons exist for less development of depth estimation using single RGB image in comparison to stereo case. Lack of scene coverage, scale ambiguities, translucent, and reflective materials increase complexity of extracting geometry from appearance. At present time, most of successful dept estimator methods rely on hardware assisted method (active approaches) such as depth sensors which are more expensive than passive approaches. In recent years, depth estimation methods based on deep neural networks learning are capable of extracting proper depth map from a single RGB image even in real-time. There are many approaches which are related to our research on monocular depth estimation.[2]

Alhashim et al. [4] has suggested a straightforward transfer learning-based network architecture that generates depth estimations with greater precision and quality. Compared to depth maps produced by approaches that use more

parameters and fewer training iterations, the output depth maps more accurately represent object boundaries. For quicker learning, create a comparable loss function, a learning method, and a straightforward data augmentation policy.

An innovative method for measuring depth from a single picture was introduced by Eigen et al. in [5]. Utilizing a neural network with two components, one of which assesses the scene's overall structure and the other of which refines it using local input, they directly regressed on the depth. In addition to pointwise error, their network was trained using a loss that explicitly takes into consideration the depth relations between pixel positions.

Ladicky et al. [6] demonstrated the method of monocular depth estimation by combining features and semantic object labels to enhance performance. However, they developed picture segmentation using handmade features and superpixels. To better estimate moving foreground topics in movies, Karsch et al. [7] added motion information to their transfer mechanism using k-Nearest Neighbors on SIFT Flow [8] to estimate the depths of static backdrops from single RBG pictures.

Several machine learning algorithms have been used in the stereo vision instance, generally yielding superior results while reducing the requirement for meticulous camera alignment. [9]. The study by Konda et al. [10] is most pertinent to our work because they trained a factored auto-encoder to estimate depth from stereo sequences using batch images. However, they depended on the lotion difference that stereo gave.

Nassir et al. used depth sensors, such as RGBD cameras and LIDAR, to get the pixel-level dense depth map directly, but their approach suffered from a limitation in measuring range and outside sunlight sensitivity [11]. LIDAR is often employed in the unmanned vehicle sector for depth measuring [12], but it only produced a sparse 3D map owing to the monocular cameras' inexpensive price, compact size, and variety of uses.

Another recent mainstream in monocular depth estimation is using transformers for extracting depth from single RGB image. The transformer which firstly utilized in Natural Language Processing has promising success in computer vision, in particular depth estimation.[13] To conclude, end-to-end deep learning research has been extensively conducted on the topic of predicting the dense depth map from a single picture.

3. Methods

3.1. Baseline model

3.1.1 Network Architecture

As we have recently started learning about Neural Networks and the convolutional networks, we tried to implement a very simple model so we can have a baseline model. We have borrowed our model from [14], which uses a simple U-net [15] as the backbone. In this model, we used the indoor image section of the DIODE dataset which we discussed before. There is roughly 81GB of data in the training set, compared to 2.6GB in the validation set. Due to the fact that this is not the final model, and training on the training data requires a lot of computational resources, we just used the validation set as our training set. Other datasets also could be used for training. In figure 1, a few samples of this dataset are plotted.

3.1.2 U-Net

U-net architecture is symmetric and consists of two major parts. A general convolutional process constitutes the first part, which is called the contracting path. The second part layers (expansive path) are formed by transposed 2d convolutional. In Figure 2, the original architecture of a U-Net is displayed.

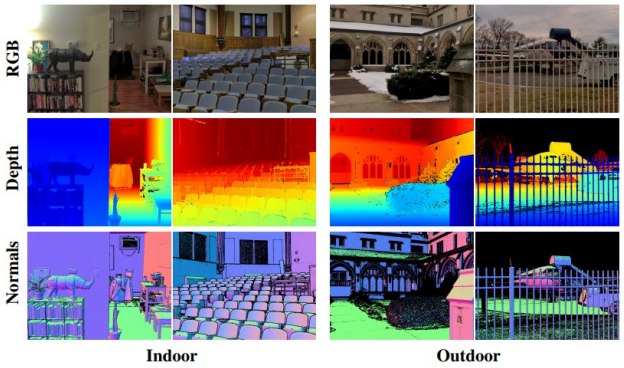

Figure 1: Random examples of indoor and outdoor pictures sampled from the DIODE dataset

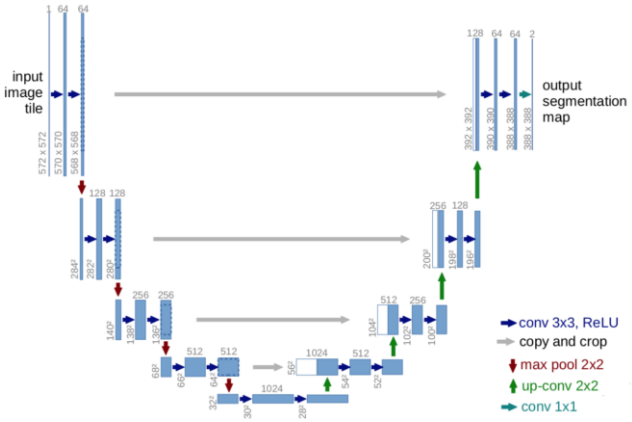

Figure 2: U-Net architecture

3.1.3 Our Model

Our model is a bit different from this one and it is a smaller version. Input files include RGB images, depth images, and depth mask images. The images are first

Dataset	Sensors	Annotation	Type	Scenario	Images	Resolution	Year
KITTI	LIDAR	Sparse	Real	Driving	44K	1024×320	2013
NYU-Depth	Kinect V1	Dense	Real	Indoor	1449	640×480	2012
DIODE	Laser Scanner	Dense	Real	In/Outdoor	25.5K	768×1024	2014
Make3D	Laser Scanner	Dense	Real	Outdoor	534	1024×2048	2016
Cityscapes	Stereo Camera	Disparity	Real	Driving	5K	2272×1704	2008
Driving Stereo	LIDAR	Sparse	Real	Driving	182K	1762×800	2019

Table 1: The most famous datasets in MDE

resized to 256*256, and then fed into the network. There are 8 convolutional layers in the downscaling part, with 16,32,64, and 128 filters per pair. In the upscaling part, it is just a reversed version. After each conv-layer, we implemented a batch-normalization layer, and after each pair, there exists a dropout as large as 0.25. The activation functions used in these blocks are Leaky ReLU, and the last layer has a Tanh activation function. We also have to mention that the kernel sizes are 3*3, padding is set to be the same, and stride is 1.

3.1.4 Loss Function

The loss function also consists of three parts. The first one in our case, is a point-wise depth loss, or L1-loss. The second part is a depth smoothness loss and the third part (which is more important) is the Structural similarity index (SSIM), which is a commonly used parameter for comparing two images.

3.2. Datasets

Datasets play a vital role in training and assessing monocular depth estimation. Quite a few datasets can be classified based on the scenario, sensors, annotation, and type of data stored in them. Here, some of the most frequent and popular ones are highlighted. In table 1, a summary of these datasets is provided.[2][3]

3.2.1 KITTI

The KITTI [16] is not only the most commonly implemented dataset in computer vision but is the most used dataset for unsupervised and semi-supervised MDE as well. The KITTI includes 44K images and their associated depth maps collected in the driving scenario from outdoor scenes in the city, rural areas, and roads. It is common to use Eigen split method for dividing this dataset into two sections for DL use: 32 scenes for training and 29 scenes for testing [1][2][17]

3.2.2 NYU Depth

The NYU Depth [18] is one the most popular dataset for depth estimation and the primary source in the training stage of supervised MDE. This dataset includes 1449 densely annotated images divided into two sections for testing and training. The NYU Depth dataset focuses on indoor environments and scenes, including indoor and depth images corresponding to them. Some studies reduced its image resolution to speed up training. [1][2]

3.2.3 DIODE

The DIODE (the Dense Indoor/Outdoor Depth) dataset [19] is a monocular depth dataset that includes diverse indoor and outdoor scenes collected with the same hardware setup. In this dataset, 8574 indoor and 16,884 outdoor samples were collected from 20 scans for training, and 325 indoor and 446 outdoor samples were collected from 10 scans for validation. In the Middlebury 2014 Middlebury dataset, 33 images have a resolution of 6 megapixels, with indoor and outdoor ranges of 50 m and 300 m, respectively. A stereo DSLR camera and two point-and-shoot cameras were used to capture the images. With a 6 megapixels resolution, the disparity ranges from 200 to 800 pixels.

3.2.4 Make3D

The Make3D dataset[20] includes 534 images taken by laser scanners during an outdoor scenario. The Make3D contains monocular RGB and depth images but does not include stereo images. Therefore, it is suitable for training steps in the supervised method. It usually is not applied to semi-supervised and unsupervised learning training and is mainly used for evaluating unsupervised learning networks over disparate datasets as a testing set. Like some of the other datasets with high-resolution images, down-sampling its images can speed up the training step.

3.2.5 Cityscapes

The Cityscapes datasets [21] possess 25K of fine- and course-annotation images mainly focused on semantic

segmentation.

3.2.6 Driving Stereo

Among the new large-scale stereo driving datasets, driving stereo contains 182k images. In addition to disparity images, LIDAR can also capture them. MDE stereo matching is evaluated using two new metrics, a distance-aware metric, and a semantic-aware metric. 1762*800 pixels are the dimensions of this dataset.[22]

3.3. Training using transfer learning

As the baseline model did not generate an acceptable depth map, we tried to do a literature review and use other models and architectures. A very good model is proposed by [4], where a transfer learning method along with some other modifications have been done.

3.3.1 Data augmentation

In deep learning tasks, we usually try to generalize our model by training it on a large dataset, and augmentation is a way towards this goal. In depth estimation tasks, the location of the floor and ceiling are important. Hence, we just can use horizontal flip. Moreover, changing the picture channels has been proved to improve the performance of our model. We have used these two methods here.

3.3.2 Loss function

In addition to the loss function we used for our baseline model that primarily concerns image reconstruction and point-wise error, we add a new term that penalizes high-frequency distortions. This is useful for having sharper boundaries for objects. So, our final loss function will become:

$$L(y,\hat{y}) = \lambda L_{\text{depth}}(y,\hat{y}) + L_{\text{grad}}(y,\hat{y}) + L_{SSIM}(y,\hat{y})$$

Where the newly added term is:

$$L_{\text{grad}}(y,\hat{y}) = \frac{1}{n}\sum_{p}^{n} |g_{\mathbf{x}}(y_p,\hat{y}_p)| + |g_{\mathbf{y}}(y_p,\hat{y}_p)|$$

3.3.3 Training

Due to hardware limitations, we could just use the pretrained weights published by the authors. They have used a powerful 12GB GPU to train the model on the NYU Depth V2 for 20 hours. The training dataset was a subset of 50k images from that dataset. The network outputs prediction depth map of 320*240 pixels, and then were upsampled. The network hyperparameters are as follows. Adam optimizer was used with a learning parameter of 0.0001, number of training parameters are 42.6M, and the training was done for 1M iterations.

3.3.4 Network Architecture

A DenseNet-169 which is trained on ImageNet dataset, has the responsibility to encode the RGB image into a feature vector as our encoder. The final depth map is constructed by combining this vector with a series of up-sampling layers. In this structure, we also use skip-connections in order to take advantage of feature reusability. In Figure 3, a picture is shown that compares the validation loss in case of random weights and transfer learned parameters.

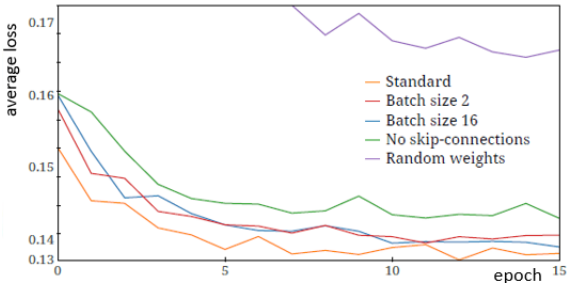

Figure 3: Average loss for different setups[4]

3.4. Transformers

Transformer networks are receiving more attention as a practical building block for problems involving computer vision in addition to its usual usage in natural language processing[23]. They recently have been used in depth estimators' architecture. We propose to use a Transformer encoder as a building block for non-local processing on a CNN's output, following the success of previous trends that combine CNNs with Transformers[13].

One of the recent application of transformers in depth estimation, is using vision transformers for dense depth prediction[24]. In this method, transformers are used in encoder section. The decoder section contains CNNs layer which named Reassemble and Fusion layers. This algorithm uses scale- and shift-invariant trimmed loss together with the gradient-matching loss for loss function definition. The algorithm architecture is shown in Figure 4.

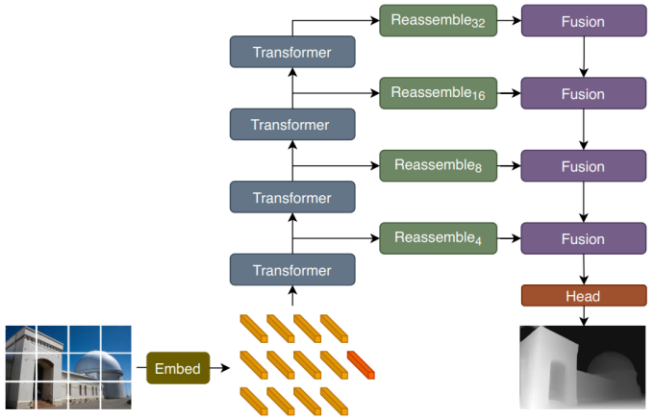

Figure 4: Architecture overview

The input image is transformed into tokens (orange) either by extracting non-overlapping patches followed by a linear projection of their flattened representation (DPT-Base and DPT-Large) or by applying a ResNet-50 feature extractor (DPT-Hybrid). The image embedding is augmented with a positional embedding and a patch independent readout token (red) is added. The tokens are passed through multiple transformer stages. We reassemble tokens from different stages into an image-like representation at multiple resolutions (green). Fusion modules (purple) progressively fuse and up sample the representations to generate a fine-grained prediction.

3.5. AdaBins

AdaBins method is another most recent technique[25]. It basically transforming monocular depth estimation to a classification task by discretizing depth range using some bins which changing adaptively with respect to datasets. Its network architecture contains two main blocks: a standard encoder/decoder and an Adaptive Bins module. Its loss function uses Scale-Invariant loss (SI) & bi-directional Chamfer Loss to not only decrease prediction error, but also to encourage the bin centers to be close to the actual ground truth depth values. Two key elements make up our architecture: an encoder-decoder block and AdaBins, a suggested adaptive bin-width estimator block. A single channel h*w depth picture is produced from an RGB image with spatial dimensions H and W as the input to our network (e.g., half the spatial resolution).[25]

4. Experimental Results

In 2014, Eigen et al.[5] proposed six metrics to evaluate the quality of depth map estimation. Since then, these parameters become popular and are used as the main metrics in the most papers in the related research area. These parameters are:

- Average relative error:

$$\frac{1}{n}\sum_{p}^{n}\frac{|y_p-\hat{y}_p|}{y}$$

- Root mean squared error (rms):

$$\sqrt{\frac{1}{n}\sum_{p}^{n}(y_p-\hat{y}_p)^2}$$

- Average (log10) error:

$$\frac{1}{n}\sum_{p}^{n}|\log_{10}(y_p)-\log_{10}(\hat{y}_p)|$$

- Threshold accuracy (δi):

$$\%\ \text{of}\ y_p\ \text{s.t.}\ max\left(\frac{y_p}{\hat{y}_p},\frac{\hat{y}_p}{y_p}\right)=\delta<thr$$
$$\text{for}\ thr=1.25,1.25^2,1.25^3$$

In the above equations, we have:
$\boldsymbol{y_p}$: a pixel in depth image y
$\widehat{\boldsymbol{y_p}}$: a pixel in the predicted depth image $\hat{y}$
n: total number of pixels for each depth image

4.1. Baseline model results

After defining the model structure and loss function, we trained our model in 50 epochs with a learning rate of 0.0001 and a batch size of 32. We could reach a validation loss of 0.1949 and test loss of 0.034 during this training process. In Figure 5, a picture from the dataset is shown along the actual depth map and the corresponding predicted map.

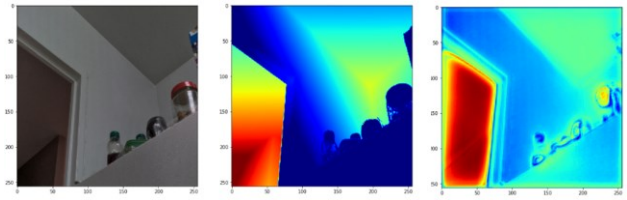
Figure 5: Results from baseline model

As it could be seen from the results, we still have not reached a good prediction. Our model completely relies on the amount of light and shadows. Also, the edges of each part are not clear enough. Actually, we somehow expected this model not to perform very well, because we did not use complicated structures (such as residual blocks), or transfer-learning (using pre-trained networks and fine-tuning them).

4.2. Training with transfer learning results

To evaluate the performance of the model, we implemented the model and tested it on some images from the indoor environment. Figure 6 depicts some examples.

Figure 6: Results from using transfer learning

As it could be seen from the pictures, the predicted depth maps are much better than our baseline model, and the boundaries of objects are visible. Regarding the metrics we introduced before, the results are shown in Table 2.

Table 2: Transfer learning method results

Method	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	rel↓	sq. rel↓	rms↓	$log_{10} \downarrow$
Using transfer learning	0.886	0.965	0.986	0.093	0.589	4.170	0.171

Although this model was quite good, some improvements could still be made, by leveraging the power of transformers.

4.3. Transformer network results

Here are its results in comparison to the state-of-the-art methods which show improvement in depth estimation performance.

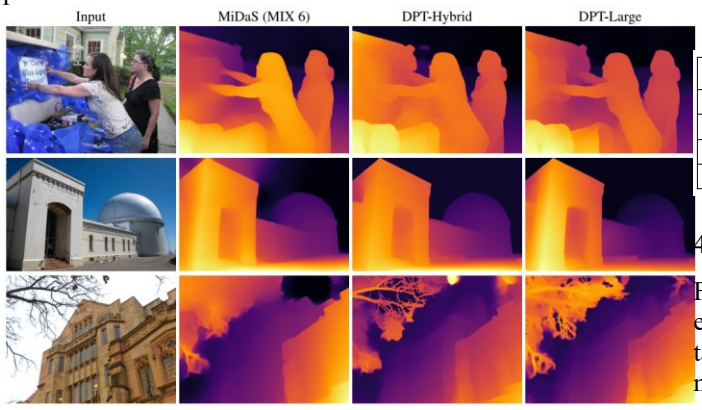

Figure 7: Sample results for monocular depth estimation. Compared to the fully convolutional network used by MiDaS, DPT shows better global coherence (e.g., sky, second row) and finer-grained details (e.g., tree branches, last row)

Compared to the fully-convolutional network used by MiDaS, DPT shows better global coherence (e.g., sky, second row) and finer-grained details (e.g., tree branches, last row) [R3].

4.4. AdaBins results

Results of proposed method on the NYU Depth dataset in comparison to the state-of-the-art methods are shown in figure 8 which outperform them in all metrics.

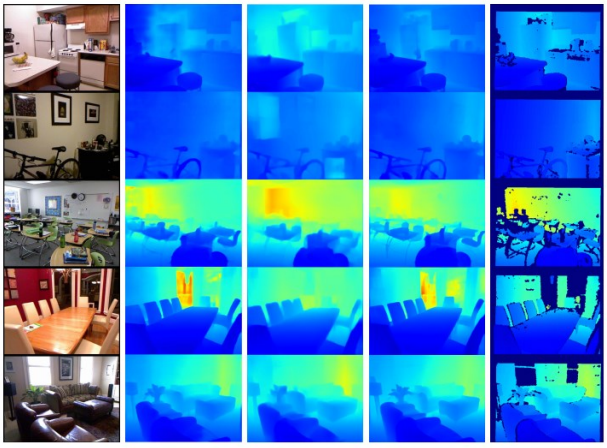

Figure 8: A comparison of the Adabins and other methods

4.5. Comparing Results

An overall comparison of results of last three methods (Transfer Learning, DPT, and AdaBins) are presented in table 3 for NYU-Depth-v2 and KITTI datasets respectively. The results are shown based on five evaluation metrics against one state-of-the-art method called BTS (Big to Small [R5]). The results show that at least one of these three methods outperform BTS.

Table 3: Comparison of results on the NYU depth dataset

Method	δ_1	δ_2	δ_3	REL	RMS
BTS (*Lee et all, arXiv preprint 2019*)	0.885	0.978	0.994	0.110	0.392
Transfer Learning (Dense-Net 169)	0.846	0.974	0.994	0.123	0.465
Dense Prediction Transformer (DPT)	0.904	0.988	0.998	0.110	0.357
AdaBins	0.903	0.984	0.997	0.103	0.364

4.6. Implementation on real world images

Finally, in order to test implemented algorithms on our environment, we produce depth map from photos were taken from indoor space in university campus by pretrained models. The depth maps are shown in figure 9. As we can see in figure below, the quality of depth map, DPT is able to generate high quality maps in near distance while AdaBins

is better in far distance. The underlying reason is datasets used for their training.

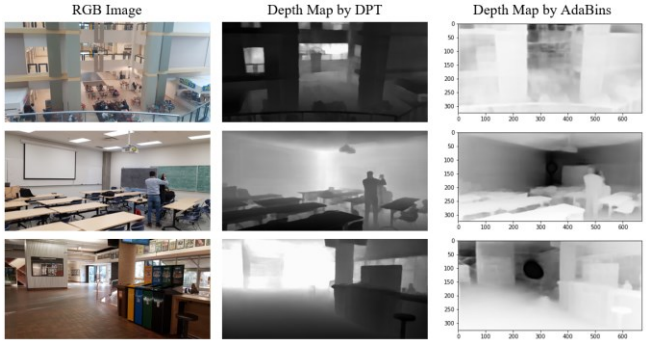

Figure 9: Depth maps resulted from implementation of Dense Prediction Transformer[24] and AdaBins[24] methods on images captured from university campus.

5. Conclusion

In this project, we chose the topic of depth estimation and tried to go through literature and learn more about it. During the past months, we first gathered information about different kinds of depth estimation (e.g. monocular depth estimation, binocular depth estimation, using stereo images, etc.). Comparing these methods and their applications and limitations, we chose Monocular depth estimation (MDE) from a single image as the topic of our project. Further, we studied classic and modern solutions to the MDE problem and found out that deep learning methods are popular tools in this area of research. Consequently, we took deep learning courses from MOOC websites (Coursera) to have a better insight on neural networks and the convolutional variation of them. These courses helped us to program a baseline encoder-decoder model that predicted depth map images from a single input image. We also had to review the most important datasets to decide which one to choose. A review of our work is written in the previous parts. Unfortunately, the baseline model was not good enough and the results were not satisfying. However, this could be predicted, as we did not use complex structures for our model.This made us go further and implement more complicated models from different papers and compare them to our baseline model. In the first step, we leveraged transfer learning methods alongside data augmentation techniques to train our model. A pretrained DenseNet169 was used as the encoder and by introducing a new term to our previous loss function, a better performance was achieved. Then, we implemented pretrained models of more advanced algorithms, vision transformer, for depth estimation. Since vision transformer utilize a transformer box as their backbone, we managed to produce more accurate depth map on KITTI and NYU-Depthv2 datasets. A compression based on standard metric has been done between transfer learning and vision transformers. Finally, we used vision transformer and AdaBins algorithms to extract depth map from images which were taken from indoor space in university campus. The produced depth map by our implementation has good accuracy given our limited hardware and GPUs.

6. Contribution:

During this project, in one hand we tried to split every one's job to work with more focus and on the other hand we actively share information and idea with each other and always consulted with each other. Despite teamworking, in summary our duties were separated as following: Mahdi implemented almost all of needed algorithm (from baseline to improved ones), write more than half of report and created half of presentation. Masoud searched on Introduction, Datasets, Literature Review, a bit try on running codes, created his half of presentation and rest of report (less than half). In conclusion, Mahdi's contribution was more than Masoud, although they continuously contacted with each other.

7. Codes

We put all of needed files including codes, reports, data, etc, in a Google drive folder: ECE 740 Course Project - Mahdi & Masoud - MDE vid DL
https://drive.google.com/drive/folders/1nRTKDEneOv3CQMbRp77fUw9fTQnSQmNV?usp=sharing

References

[1] A. Bhoi, "Monocular Depth Estimation: A Survey," 2019, [Online]. Available: http://arxiv.org/abs/1901.09402.

[2] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular Depth Estimation Using Deep Learning: A Review," *Sensors*, vol. 22, no. 14, pp. 1–24, 2022, doi: 10.3390/s22145353.

[3] X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass, "Towards Real-Time Monocular Depth Estimation for Robotics: A Survey[-5pt]," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–22, 2022, doi: 10.1109/TITS.2022.3160741.

[4] I. Alhashim and P. Wonka, "High Quality Monocular Depth Estimation via Transfer Learning," 2018, [Online]. Available: http://arxiv.org/abs/1812.11941.

[5] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Adv. Neural Inf. Process. Syst.*, vol. 3, no. January, pp. 2366–2374, 2014.

[6] L. Ladický, J. Shi, and M. Pollefeys, "Pulling things out of perspective," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 89–96, 2014, doi: 10.1109/CVPR.2014.19.

[7] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer:

Depth extraction from video using non-parametric sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2144–2158, 2014, doi: 10.1109/TPAMI.2014.2316835.

[8] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *Dense Image Corresp. Comput. Vis.*, pp. 15–49, 2015, doi: 10.1007/978-3-319-23048-1_2.

[9] F. H. Sinz, J. Q. Candela, G. H. Bakir, C. E. Rasmussen, and M. O. Franz, "Learning depth from stereo," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3175, pp. 245–252, 2004, doi: 10.1007/978-3-540-28649-3_30.

[10] K. Konda and R. Memisevic, "Unsupervised learning of depth and motion," 2013, [Online]. Available: http://arxiv.org/abs/1312.3429.

[11] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6565–6574, 2017, doi: 10.1109/CVPR.2017.695.

[12] K. Yoneda, H. T. Niknejad, T. Ogawa, N. Hukuyama, and S. Mita, "Lidar scan feature for localization with highly precise 3-D map," *2014 IEEE Intell. Veh. Symp. Proc.*, pp. 1345–1350, 2014.

[13] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12346 LNCS, pp. 213–229, 2020, doi: 10.1007/978-3-030-58452-8_13.

[14] V. Basu, "Monocular depth estimation," 2021. https://keras.io/examples/vision/depth_estimation/.

[15] W. Weng and X. Zhu, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, vol. 9, pp. 16591–16603, May 2015, doi: 10.48550/arxiv.1505.04597.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research," *Int. J. Rob. Res.*, no. October, pp. 1–6, 2013.

[17] A. Mertan, D. J. Duff, and G. Unal, "Single image depth estimation: An overview," *Digit. Signal Process. A Rev. J.*, vol. 123, pp. 1–18, 2022, doi: 10.1016/j.dsp.2022.103441.

[18] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7576 LNCS, no. PART 5, pp. 746–760, 2012, doi: 10.1007/978-3-642-33715-4_54.

[19] I. Vasiljevic *et al.*, "DIODE: A Dense Indoor and Outdoor DEpth Dataset," Aug. 2019, doi: 10.48550/arxiv.1908.00463.

[20] A. Saxena, M. Sun, and A. Ng, "Make3D: Learning 3D Scene Structure from a Single Still Image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 824–840, 2009, doi: 10.1109/TPAMI.2008.132.

[21] M. Cordts *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 3213–3223, 2016, doi: 10.1109/CVPR.2016.350.

[22] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 899–908, 2019, doi: 10.1109/CVPR.2019.00099.

[23] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2020, [Online]. Available: http://arxiv.org/abs/2010.11929.

[24] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision Transformers for Dense Prediction," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 12159–12168, 2021, doi: 10.1109/ICCV48922.2021.01196.

[25] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth Estimation Using Adaptive Bins," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4008–4017, 2021, doi: 10.1109/CVPR46437.2021.00400.