Monocular Depth Estimation using Deep learning

Sai Karthik Vyas Akondi
University at Buffalo
50488251
sakondi@buffalo.edu

1 Overview

Monocular depth estimation is a task that uses image data and estimates the depth or distance of the image from the camera. My idea is to use deep learning techniques like Convolution Neural Networks, Neural Style Transfer and Generative Adversarial Networks and compare them with most widely used state of the art MIDAS model. Monocular Depth Estimation is widely used in Autonomous Vehicles, UAV's and other robotic applications. Analysing different types of Deep Learning techniques to get depth maps would be very useful in these applications.

The main idea of this project is to analyze different types of deep learning techniques for Monocular Depth Estimation task. The main challenge with Monocular depth estimation is use of sequential data from a single camera which makes it challenging to abstract depth maps using simple computer vision techniques. Whereas in the case of Stereo depth estimation, where left and right sequences and images are given is much simpler.

Inputs : This Deep learning application takes RGB images from sequential data of indoor scenes Outputs : Depth maps, the outputs will be Depth matrix which will have depth matrix is a matrix with depth of each pixel from the camera.

1.1 Dataset

The data-set selected for this project is the NYU V2 data-set, consisting of 1449 indoor scenes with corresponding RGB images and ground truth depth maps. The data-set provides various indoor environments essential for the depth estimation task. Some of the most significant scenes are listed below:

- Scene Type: Playroom, Number of Images: 1330
- Scene Type: Living Room, Number of Images: 12242
- Scene Type: Dining Room, Number of Images: 7778
- Scene Type: Bedroom, Number of Images: 15466
- Scene Type: Office, Number of Images: 3270

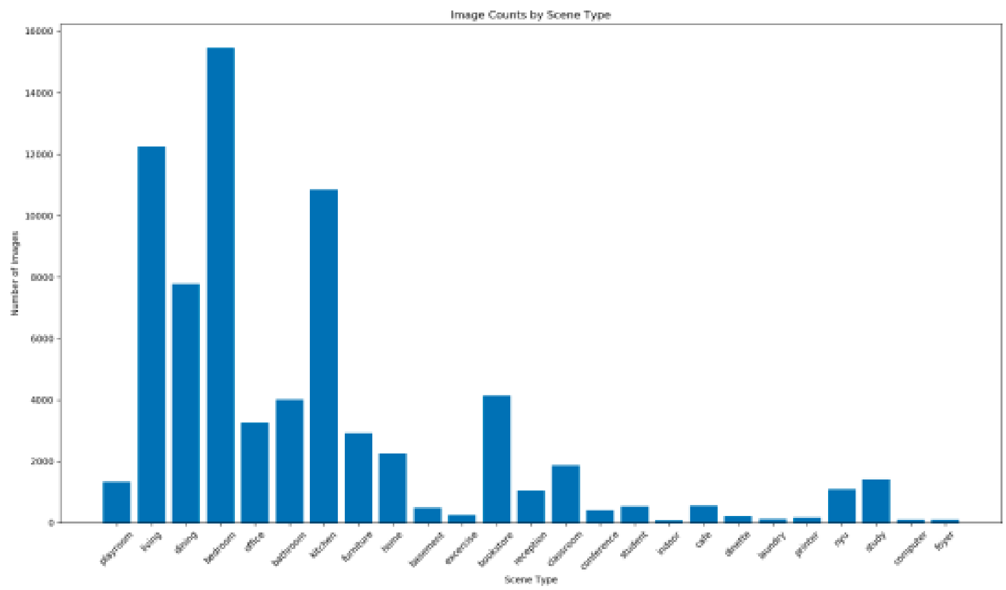

Figure 1: NYU V2 Scene distribution

Figure-1 shows the distribution of all scenes

1.2 State of the Art

Monocular depth estimation has seen significant advances in recent years, with deep learning techniques at the forefront of these developments. Current state-of-the-art solutions employ a variety of approaches, each offering unique insights and advancements in depth estimation from single images. Key contributions include:

- The **MIDAS model** represents a breakthrough in depth estimation, offering robust and accurate predictions. This model's strength lies in its ability to generalize across different scenes and lighting conditions, making it highly versatile for various applications.
- The work by Eigen et al. [1] entitled *"Depth Map Prediction from a Single Image using a Multi-Scale Deep Network"* introduced a novel approach that leverages multi-scale neural networks. This methodology significantly improved the accuracy of depth predictions from single images, setting a new benchmark for subsequent research.
- In *"Indoor Segmentation and Support Inference from RGBD Images"* [11], a combination of image segmentation and the RANSAC algorithm is employed. This study enhanced understanding of the indoor scenes' structural elements, providing a foundational technique for depth estimation.
- Gatys et al. [2] in their work on *"Image Style Transfer Using Convolutional Neural Networks"* provided a unique perspective on the use of neural networks, influencing approaches in depth estimation.

2 Approach

2.1 Deep Learning Algorithms

Deep learning techniques have always been pretty useful for images tasks. Currently the following four techniques have been explored in Dept estimation task :

- **CNN with Batch normalization**
- **Pre-trained MIDAS model**
- **Neural Style transfer using VGG-19 weights**
- **Generative Adversarial Networks**

In the above mentioned algorithms: the CNN model with Batch normalization implemented using Pytorch was implemented without using any existing applications. The Neural Style transfer and GAN haven't been used for depth estimation till now, hence making this project novel.

2.2 CNN with Batch normalization

Contribution : Dataset class, Model, Training loop was written and implemented with minimal use online repositories and pre-trained models.

2.2.1 Model

The CNN model uses the following layers:

- Conv2d Layers : Conv2D layers are the most important part of any CNN model, they take the input and perform Convolution operation on given input. When the convolution operation is performed, important features of input data is extracted into a generic way.
- Batch normalization layers : Batch normalization layers normalize (standardize) the data, and in turn the model's performance increases.
- Transposed Convolution Layers : These layers take the convolution outputs and then bring back the extracted features in higher dimensions. This is helpful in getting back the features that we require for getting depth map outputs.
- Activation : RELU activation function removes negative values and only keeps the positive one's. Acting as a filtering mechanism
- Max-pooling : Max pooling is used to take the max value in a given filter window, this skips rest of the unnecessary values and helps in making the training faster.

2.2.2 Training

Hardware used for training : NVIDIA RTX 3050 , 4GB graphics card.

The model is trained on the whole dataset. The dataset is loaded using Pytorch Dataset and Dataloader classes. The shuffle parameter is set to false, due to a CUDA error that was observed during initial training. If shuffle was set to true, the model would have performed a bit better.

Training is done on epoch sizes of 3 and 5, loss function Mean Squared Error and Adam optimizer. During initial runs, the model was outputting a blank image and the loss was very high. After changing the loss function to L1 loss, the model performance improved and the depths of some features was captured in the predicted image. The model performance might improve by increasing the number of epochs, but computationally it's not that easy with a 4GB graphics card. Model is saved as a pth file, which can be useful for further training and fine tuning.

2.3 Pre-trained MIDAS model

Contribution : Code for loading, transforming, and passing image through pretrained model was written. Pre-trained MIDAS model was imported from Torch hub.

Hardware used for training : NVIDIA RTX 3050 , 4GB graphics card.

MIDAS model is a part of PyTorch, a tool for building AI models, and is very good at estimating depths. This part of the project involved loading the MIDAS DPT_Large model using torch load. The process involved loading an image using OpenCV, converting it from BGR to RGB format, and aligning it with the MiDaS model's format.After applying the necessary transformations to the image, it was fed into the MiDaS model to predict depth. To accurately represent the depth information, the model's output was resized to match the original image dimensions.

Result : MIDAS model had the best depth map outputs, which were close to the ground truth images.

2.4 Neural Style transfer using VGG-19 weights

Contribution: This was directly taken from Torch implementation of Neural Style transfer for a different application. Modifications were done to fit our application into the model, i.e changing the input features, output features, and conversion of output RGB to Grayscale to showcase depth images.

Hardware used for training : NVIDIA RTX 3050 , 4GB graphics card.

In this section, the VGG19 model was utilized with its pre-trained weights. The process began with loading and configuring the VGG19 model from PyTorch's library of pre-trained models. For image processing, content and style images were prepared using PyTorch's transform function.

Style Image: Existing depth images Content Image: RGB image

The neural style transfer implementation involved defining specific layers within the VGG19 model to track content and style losses. Content loss was computed using the Mean Squared Error between the feature maps from the content and generated images, while the style loss tracked with the Gram matrix.

Result : Neural Style transfer with VGG-19 model weights, did not output expected depth images. This might be due to the face that, the main aim for neural style transfer is aimed for RGB to RGB conversion.

2.5 Generative Adversarial Networks

Hardware used for training : NVIDIA RTX 3050 , 4GB graphics card.

GAN for depth images is also a new research area, that hasn't been done yet. Currently this part of the application is still in progress. I'm going through some papers and some code online on GAN, to apply it to Depth estimation tasks.

3 Experimental Protocol

3.1 Dataset

3.1.1 NYU V2 Dataset

The NYU Depth Dataset V2 was used for depth estimation, which is a comprehensive collection of RGB and depth images from indoor scenes.

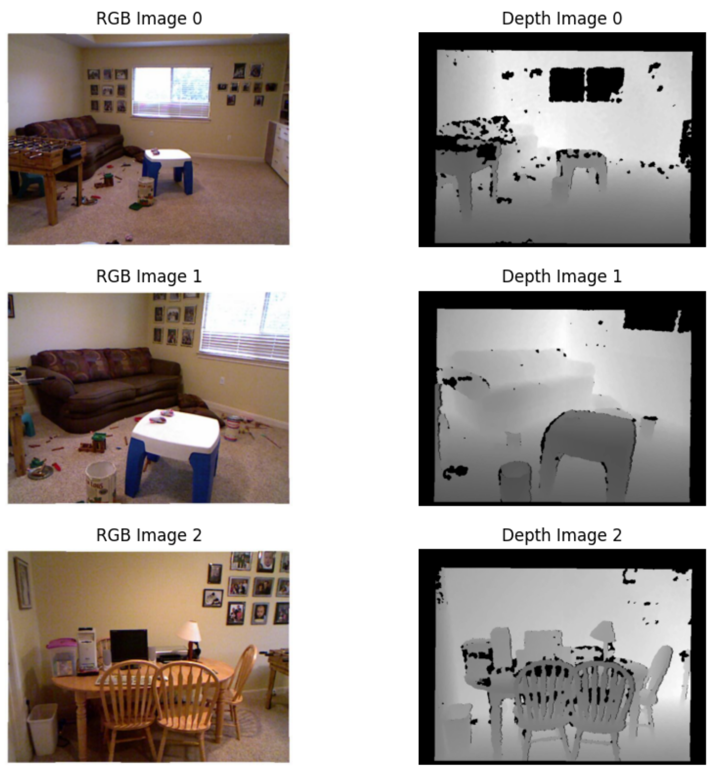

Figure 2: NYU V2 Dataset

3.1.2 Relevance to the Depth Estimation task

This dataset is particularly selected due to its diverse environments and scenarios, providing a better chances of diversity for training and evaluating the depth estimation model. For the neural style transfer, custom images were selected from NYU V2 dataset, with RGB images serving as content and depth images as style sources.

3.2 Evaluation of Success

The most simplest qualitative result for any depth estimation task is obtained by just comparing the ground truth depth images with that of the predicted images.

3.3 Computational Resources

3.3.1 Hardware :

NVIDIA RTX 3050, 4GB graphics card was used to train the models for Depth Estimation task.

3.3.2 Frameworks and Libraries:

PyTorch's library for pre-trained models (e.g., VGG19, MIDAS)

4 Results

4.1 CNN with Batch Normalization

The CNN model's performance was not as effective as state-of-the-art CNN models, primarily due to its less complex architecture. This is evident from the loss graph shown below:

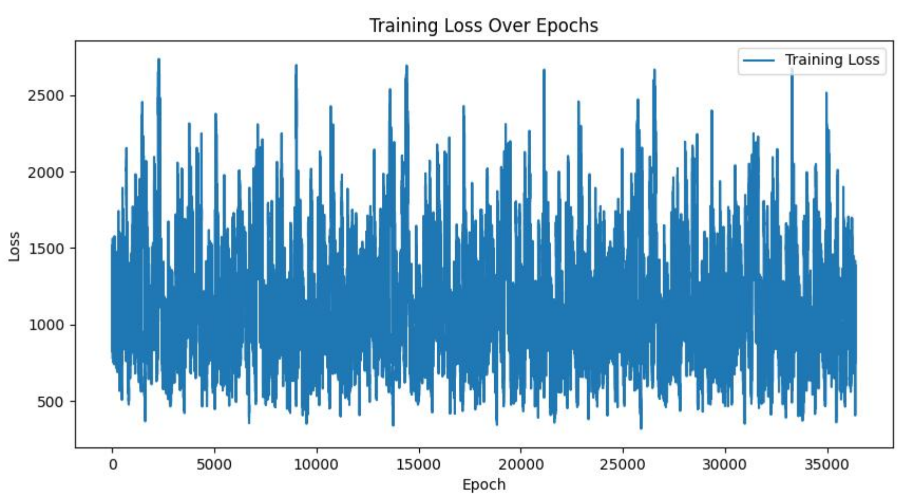

Figure 3: Loss graph of the CNN model

Figure 4: RGB Image

Figure 5: Ground truth depth image

Although the output images from the CNN model contained a significant amount of noise, some aspects of the depth maps were recognizable and corresponded reasonably well to the expected output.

Figure 2 shows the fluctuation in loss during the training. In the graph it's clearly observed that the training cross is in the range of 300 and 2300, which suggests that

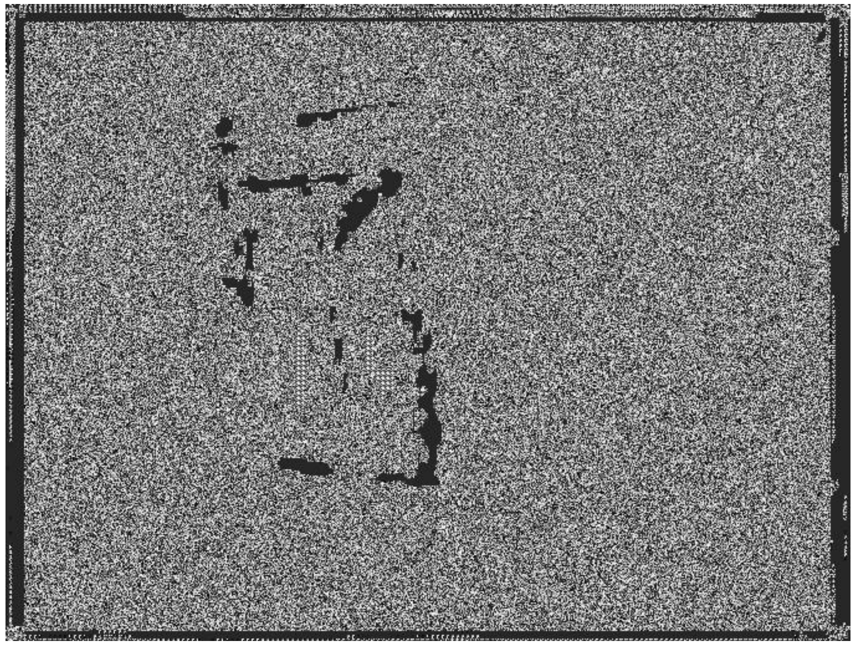

Figure 6: Predicted Image - CNN with Batch normalization

the Deep learning model should have more layers and complexity to fit the task of Depth estimation.

The output of the Deep learning model is shown in Figure-3:

Comparing the RGB (Figure 3), Ground truth depth (Figure 4) and CNN predicted depth images(Figure 5)

4.2 Pre-trained MIDAS Model

The Pre-trained MIDAS model showed a marked improvement in depth estimation accuracy. The depth maps generated by this model were closer to the ground truth, as illustrated in the image below:

The superior performance of the MIDAS model is attributed to its advanced architecture and the extensive training it has undergone, which is reflected in the quality of the depth maps it produces.

The pretrained MIDAS model has the best depth outputs out of all three

4.3 Neural Style transfer using VGG-19 weights

Neural style transfer involves fusing two images: a content image (such as a photograph) and a style image (usually an artwork), to produce a result that maintains the content of the first image but is stylistically similar to the second. VGG-19 weights were used, as the model is a bench mark in Neural Style transfer.

The novel idea was to implement Neural style transfer for our Depth estimation task. This method sought to integrate the depth information with the RGB content. However, the results indicated that this novel approach did not give fruitful results.

Figure 7: Depth map generated by the Pre-trained MIDAS model

Figure 8: Content Image

Considering Style Image as the Ground Truth Depth and Content image as it's corresponding RGB image. Neural style transfer was implemented, the result was an empty image, monochromatic, lacking any meaningful synthesis of the depth data with the RGB content.

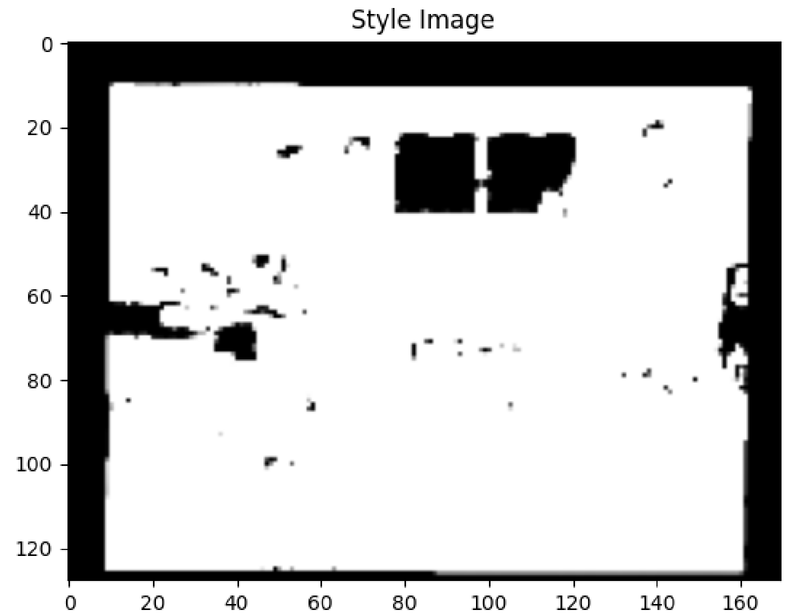

Figure 9: Style Image

4.4 GAN for Depth Estimation task

In this experiment to use Generative Adversarial Networks (GANs) for the task of estimating depth in images. The complexities involved in adapting GANs for monocular depth estimation task required more time and couldn't be completed in this time frame. Hence this section of the project was not completed as expected and therefore no results.

5 Analysis

5.1 CNN with Batch normalization

Improving the complexity of the

5.1.1 Advantages

- The advantage of using a CNN-based approach with batch normalization is its simplicity and ease of implementation.
- It is computationally less intensive compared to more complex models, making it feasible for training on hardware with limited resources.

5.1.2 Limitations

- The main limitation of this approach is its limited capacity to capture complex depth information. The simple architecture may not be sufficient to learn depth patterns.

Figure 10: Output of Neural Style transfer - blank image

5.1.3 Improvements

- Although CNN with Batch normalization did not give expected results, this technique can be improved by augmenting data and increasing the number of epochs for training.

5.2 Pre-Trained MIDAS model

5.2.1 Advantages

- The depth maps generated by the MIDAS model closely resemble ground truth depth images, indicating high-quality results.

- Pre-trained MIDAS was the best model for Depth estimation task, it gave out best results both in terms of accuracy and the capturing of depth features.

5.2.2 Limitations

- One of the most important limitation to consider is that the Pre-trained MIDAS model was trained on the same NYU-V2 dataset. So it might be possible that the features might have over-fitted.

5.2.3 Improvements

- Training on different types of data will be the most important improvement for this model.

5.3 Neural Style Transfer

5.3.1 Advantages

- The main advantage about this technique is it's novelty. Neural Style Transfer is a creative approach to depth estimation, attempting to combine style (depth) with content (RGB) images to generate unique results.
- It uses pre-trained VGG-19 weights, which are known for their effectiveness in image processing tasks.

5.3.2 Limitations

- Neural Style transfer was mainly developed for converting images to paintings, here both the content and style images are RGB.
- In our application the style image is gray-scale and content image is RGB, research has to be done in this are for effectively applying Neural Style transfer for Depth task.

5.3.3 Improvements

- Instead of using VGG-19 pre-trained model, other models specifically designed for depth task can be used.

6 Discussion

In summary, this project aimed to explore different deep learning methods for monocular depth estimation, an important in computer vision with applications in robotics, autonomous vehicles, Four techniques were researched and three of them were implemented :

- CNN with Batch normalization
- MIDAS pre-trained model
- Neural Style transfer using VGG-19 weights
- GANs –> this was not implemented

The pre-trained MIDAS model produced the most promising results, almost resembling ground truth depth maps. However, the custom CNN model had limitations due to its simplicity. Adapting neural style transfer for depth estimation faced challenges related to image style differences between RGB and depth images. The application of GANs for depth estimation is still an ongoing research. To improve accuracy, future work should explore more complex CNN architectures, diverse data sets, and innovative techniques bridging RGB and depth information

References

[1] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Monodepth2: Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[6] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Proceedings of the 4th International Conference on 3D Vision (3DV)*, 2016.

[7] F. Liu, C. Shen, and G. Lin. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2015.

[8] F. Liu, C. Shen, G. Lin, and I. Reid. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and J.-M. Frahm. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[10] 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*, 31(5):824–840, 2009.

[11] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, 2012.

Online Resources

a. PyTorch. (n.d.). *Neural Style Transfer Tutorial*. Retrieved from `https://pytorch.org/tutorials/advanced/neural_style_tutorial.html`

b. PyTorch. (n.d.). *MiDaS Model for PyTorch*. Retrieved from https://pytorch.org/hub/intelisl_midas_v2/

c. The Computer Vision Foundation. (n.d.). *Open Access to Computer Vision Research*. Provides access to a wide range of research papers and resources in computer vision. Retrieved from https://www.thecvf.com/

d. NYU Depth Dataset V2. (n.d.). *NYU Depth Dataset V2*. A dataset of indoor scenes for depth estimation. Retrieved from https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html