Analysing the impact of loss functions on the accuracy of monocular depth estimation

Rishabh Singh Kushwaha
Stanford University
rish94@stanford.edu

Abstract

Monocular depth estimation is an important problem in computer vision having applications in autonomous systems, augmented reality, robotics and biological imaging. There are multiple approaches to estimate the depth map of an image. Implementation of a deep neural network is a popular approach and there have been numerous studies on monocular depth estimation using deep networks. In this project, a simple U-Net architecture network is implemented to predict depth map of images. Training and prediction performance is analysed against a combined loss function which is a weighted average of individual loss functions. Depth maps generated by the network are then compared with ground truth using RMSE as validation metrics to quantify model performance. Further, batch normalization and data augmentation techniques are also implemented to check if it results in any improvement.

1. Introduction

When an image of a scene is captured using a camera, the 3-dimensional information of the scene is lost. But it is possible to understand the spatial variation in images due to different depths of objects present in the image. This understanding could be useful in prediction of different objects in an image and hence recovering the 3-dimensional information.

The recovery of the 3D information of the scene, in particular constructing a depth map, has wide ranging applications including autonomous vehicles, biological imaging, robotics, augmented reality and industrial manufacturing. Recently, semiconductor manufacturers have also started working towards recovering the depth information of etched holes. Understanding the depth variation across the semiconductor wafer is essential towards improving their lithographic process and build next generation chips. [7]

Different approaches have been explored for depth estimation, which can be broadly divided into active techniques and passive techniques. Active techniques usually employ specialized light sources and scanning systems, for example in lidar systems and structured light imaging. On the other hand, passive depth estimation relies instead of the ambient light illumination of the scene. Examples include stereo vision and light field cameras. [4] [6] Supervised and unsupervised learning based have also gained popularity due to rapidly increasing compute capability. [1] Unsupervised learning methods tend to offer more generality whereas supervised learning methods are simple to implement and provide better accuracy.

Accuracy of supervised learning method usually depend upon two factors: 1) Loss function ; 2) Model architecture. This project report contains experimental results when supervised learning method is used for depth estimation task. My main contributions in this project can therefore be categorized into three parts:

- Training model with different weights for respective loss functions and reporting the accuracy.
- Analysing the impact on data augmentation.
- Tuning the model architecture for better learning.

2. Background

In the past decade, deep learning-based depth estimation methods have made significant advancements. Eigen et al., (2014) introduced an early deep learning model that used a convolutional neural network (CNN) to estimate depth from single RGB images. [2] More recent supervised methods have introduced advanced architectures such as U-Net [9] for improved depth prediction.

Traditional loss functions play a crucial role in training depth estimation models. Common loss functions include Mean Squared Error (MSE) and Mean Absolute Error (MAE), which measure the squared and absolute differences between predicted and true depth values, respectively. Other loss functions, such as structural similarity index (SSIM) [8], focus on perceptual quality, promoting visually enhanced depth maps. SILog loss is another function which addresses the issue of varying scales in depth data These losses can be used in combined functions, like MAE-SSIM, Edge-Depth, SILog and , enhancing the overall accuracy and perceptual quality of depth predictions.

3. Approach

[3]

3.1. Dataset

DIODE: A Dense Indoor and Outdoor Depth Dataset containing 300 images with resolution of 1024 x 768 for both RGB and depth map is used in this study. However, input image size is reduced to 256 x 256 pixels to save training time.

3.2. Loss Functions

- **MAE Loss** The L1 loss, also known as the Mean Absolute Error (MAE), is given by the formula:

  $$\mathcal{L}_{L_1} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{1}$$

  where:

  - n is the number of samples.
  - y_i is the true value for the i-th sample.
  - $\hat{y}_i$ is the predicted value for the i-th sample.

- **SSIM Loss** The Structural Similarity Index Measure (SSIM) is often used for evaluating the similarity between two images. When used as a loss function, the goal is typically to maximize SSIM, but for a loss function, we might use (1-SSIM) so that lower values correspond to better quality.

  $$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

  where:

  - x and y are the two images being compared.
  - μ_x and μ_y are the mean intensities of x and y, respectively.
  - σ_x^2 and σ_y^2 are the variances of x and y, respectively.
  - σ_{xy} is the covariance of x and y.
  - C_1 and C_2 are constants to stabilize the division with weak denominator.

  The SSIM loss function can be defined as:

  $$\mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(d, \hat{d}) \tag{3}$$

- **Edge depth Loss**

  The edge-aware smoothness loss is commonly used in depth estimation tasks to enforce the smoothness of the depth map while preserving edges.

  $$\mathcal{L}_{\text{edge}} = \sum_{i,j}\left(|\partial_x d_{i,j}|e^{-\|\partial_x I_{i,j}\|} + |\partial_y d_{i,j}|e^{-\|\partial_y I_{i,j}\|}\right) \tag{4}$$

  where:

  - $d_{i,j}$ is the predicted depth value at pixel (i, j).
  - $I_{i,j}$ is the image intensity at pixel (i, j).
  - ∂_x and ∂_y denote the gradients in the x and y directions, respectively.
  - $|\partial_x d_{i,j}|$ and $|\partial_y d_{i,j}|$ are the absolute values of the depth gradients in the x and y directions.
  - $e^{-\|\partial_x I_{i,j}\|}$ and $e^{-\|\partial_y I_{i,j}\|}$ are the exponential terms that weight the depth gradients based on the image gradients, emphasizing smoothness in regions with less texture and preserving edges where image gradients are high.

- **Scale invariant log(SILog) loss** type of loss function used in depth estimation tasks that is designed to handle the scale ambiguity problem inherent in depth prediction. The scale ambiguity problem arises because the depth map can be scaled by an arbitrary positive constant without changing the quality of the prediction.

  $$\mathcal{L}_{\text{SILog}} = \frac{1}{n}\sum_i \left(\log d_i - \log \hat{d}_i\right)^2 - \frac{\alpha}{n^2}\left(\sum_i \log d_i - \log \hat{d}_i\right)^2 \tag{5}$$

  where:

  - d_i is the ground truth depth value at pixel i.
  - $\hat{d}_i$ is the predicted depth value at pixel i.
  - n is the total number of pixels.
  - α is a constant term, typically set to balance the scale-invariant and scale-dependent components.

- **Combined Loss function**

  $$\mathcal{L}_{\text{combined}} = \lambda_1\mathcal{L}_{\text{SILog}} + \lambda_2\mathcal{L}_{\text{SSIM}} + \lambda_3\mathcal{L}_{\text{Edge}} + \lambda_4\mathcal{L}_{\text{L1}} \tag{6}$$

  where:

  $$\mathcal{L}_{\text{SILog}} = \frac{1}{n}\sum_i \left(\log d_i - \log \hat{d}_i\right)^2 - \frac{\alpha}{n^2}\left(\sum_i \log d_i - \log \hat{d}_i\right)^2 \tag{7}$$

$$\mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(d, \hat{d}) \tag{8}$$

$$\mathcal{L}_{\text{Edge}} = \frac{1}{n} \sum_{i} \left(\left| \nabla_x d_i - \nabla_x \hat{d}_i \right| + \left| \nabla_y d_i - \nabla_y \hat{d}_i \right| \right) \tag{9}$$

$$\mathcal{L}_{\text{L1}} = \frac{1}{n} \sum_{i} |d_i - \hat{d}_i| \tag{10}$$

In the equations above:

- d_i is the ground truth depth value at pixel i.
- $\hat{d}_i$ is the predicted depth value at pixel i.
- n is the total number of pixels.
- α is a constant term, typically set to balance the scale-invariant and scale-dependent components in the SILog loss.
- SSIM$(d, \hat{d})$ is the Structural Similarity Index between the ground truth depth map d and the predicted depth map $\hat{d}$.
- ∇_x and ∇_y represent the image gradients in the x and y directions, respectively, used to calculate the edge depth loss.
- $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are weighting factors that balance the contributions of each loss term.

3.3. Model Architecture and Data Augmentation

Model Architecture The U-Net architecture is a popular model for image-to-image tasks, including depth estimation and this architecure is implemented in this project. UNet is a type of convolutional neural network (CNN) that consists of an encoder (contracting path) and a decoder (expanding path). Given below are key elements of U-Net architecture:

- **Input Image:** The model takes an input image of a specified size (e.g., 256x256x3).
- **Encoder:** The encoder consists of three convolutional blocks followed by max pooling layers. It extracts features at multiple levels of abstraction.
- **Bottleneck:** The bottleneck layer is a convolutional block with 512 filters, which represents the deepest part of the U-Net, where the features are highly abstracted.
- **Decoder:** The decoder consists of upsampling layers and convolutional blocks. It gradually reconstructs the spatial dimensions using the features from the encoder to maintain detail and accuracy in the output.
- **Output Map:** The final layer is a convolutional layer that maps the features back to the desired output space, typically using a 1x1 convolution to reduce the depth to the number of desired output channels.

Data Augmentation is a technique used to artificially increase the size of a dataset by applying various transformations to the existing data samples. [5] In this study, we apply the following transformations to our dataset:

- **Image Flipping:** Flipping images horizontally or vertically.
- **Brightness Adjustment:** Adjusting the brightness of images.
- **Contrast Adjustment:** Adjusting the contrast of images.

4. Experiments and Results

Training parameters used in the UNet architecture for depth estimation are given in Table1 Experiments can be broadly divided into three categories.

Table 1. Training parameters

Training Parameters	Value
Input image size	256 x 256
Learning Rate	0.0002
Epoch	10
Batch Size	24

4.1. Tuning weights of the combined loss function

- **Tuning SILog and SSIM loss weights**: Training and validation losses are analysed for different weight w of SILog and SSIM loss. Here we plot

$$\mathcal{L}_{\text{combined}} = w\mathcal{L}_{\text{SSIM}} + (0.9 - w)\mathcal{L}_{\text{SILog}} + + \lambda_3 \mathcal{L}_{\text{Edge}} + \lambda_4 \mathcal{L}_{\text{L1}} \tag{11}$$

- **Tuning SSIM and Edge Depth loss weights**: Based on previous result, SILog loss function doesn't help in improving the minimizing the loss. Therefore, SILog loss is assigned zero weight for this experiment. Training and validation losses are now analysed for different weight w of SSIM and Edge depth loss.

$$\mathcal{L}_{\text{combined}} = (0.9 - w)\mathcal{L}_{\text{SSIM}} + w\mathcal{L}_{\text{Edge}} + \lambda_4 \mathcal{L}_{\text{L1}} \tag{12}$$

It was observed that training becomes unstable if edge weights are increased beyond 0.5. 2 records the weights assigned to each loss function in combined

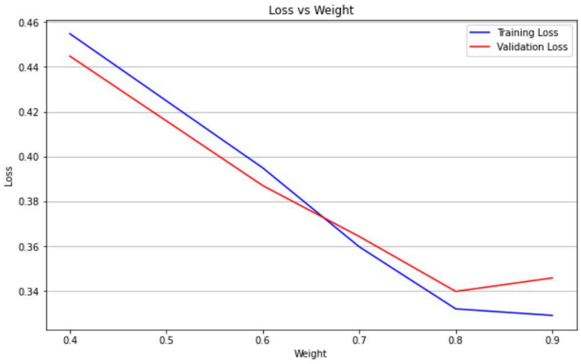

Figure 1. Combined loss after training is concluded $\mathcal{L}_{\text{combined}}$ against w

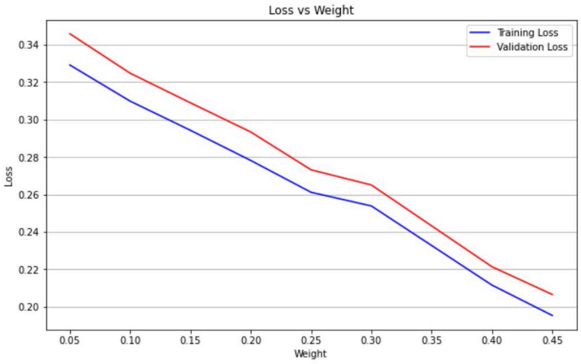

Figure 2. Combined loss after training is concluded $\mathcal{L}_{\text{combined}}$ against w

Table 2. Table of Loss Functions and Corresponding Data

Loss Function	Weights
$\mathcal{L}_{\text{SSIM}}$	0.55
$\mathcal{L}_{\text{Edge}}$	0.45
$\mathcal{L}_{\text{SILog}}$	0.0
$\mathcal{L}_{\text{L1}}$	0.05

loss function. 2 give the training and validation loss with epochs for the weight configuration given in table 2

4.2. Batch Normalization in Model Architecture

Training performance remained similar after using batch normalization. 5 High loss in validation set can be attributed to smaller dataset used for validation.

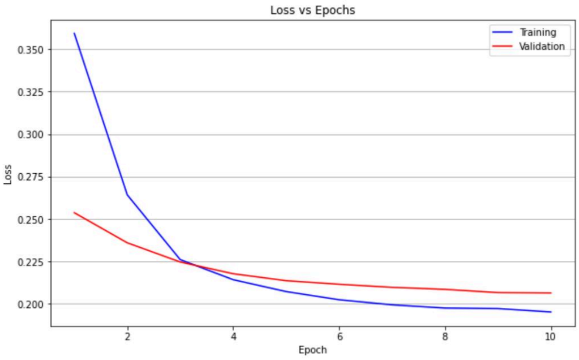

Figure 3. Loss vs Epochs for combined loss function

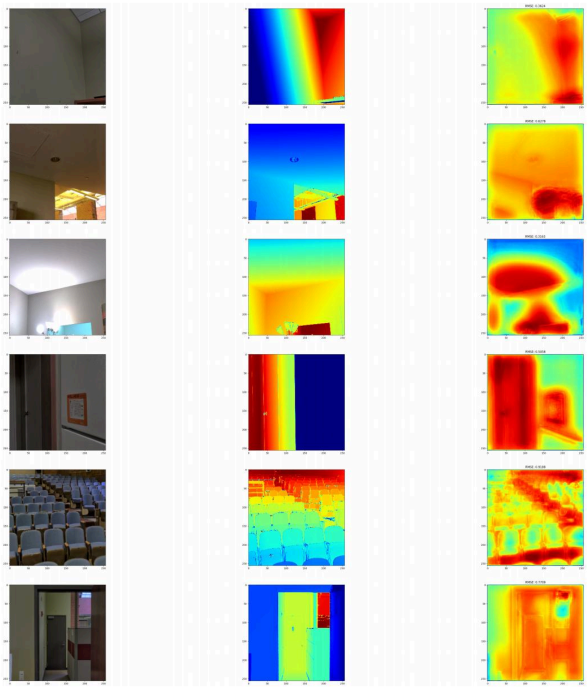

Figure 4. RGB image, ground truth depth map and predicted depth map for test data using model trained using weights given in table 2

4.3. Data Augmentation

I added data augmentation by incorporating flipped, modified contrast and brightness images in the training dataset and plotted the training and validation loss curves in Fig. 6

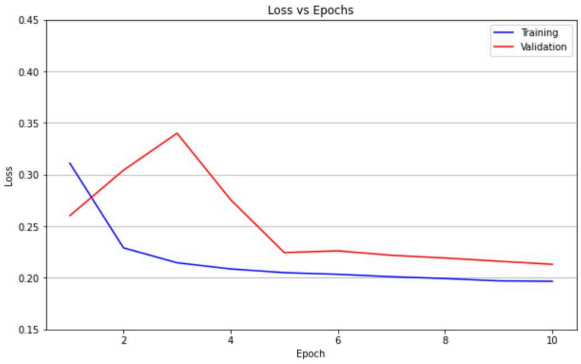

Figure 5. Training and Validation loss Loss with batch normalization

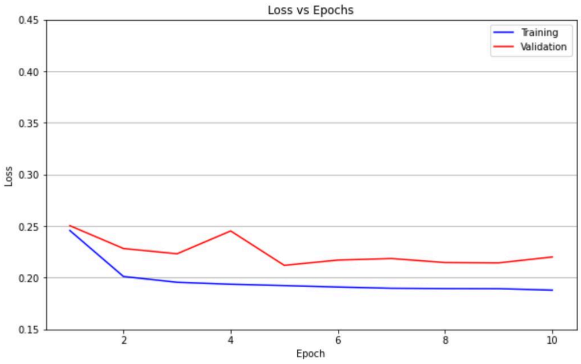

Figure 6. Training and validation loss with data augmentation enabled.

5. Conclusion

In this paper, I presented a depth estimation approach using a modified U-Net architecture. My primary contributions include the introduction of a combined loss function integrating SILog, SSIM, edge depth, and L1 losses, as well as an effective data augmentation strategy that enhances the generalization capabilities of the model. Given below are few findings from the project.

- The dataset used in this study was an indoor dataset and it was expected that SILog loss would not be critical in depth prediction performance. This hypothesis is verified in our study as we got better training performance when SSIM loss was given higher weight that SILog loss.

- A high weight is chosen for edge depth loss and as a result, current implementation of this model is able to capture edge information reasonably well.

- It also captures the depth but it is still far from being a reliable and accurate prediction model. There is still a significant amount of work required to ensure a reliable depth prediction.

6. Scope

- **Objectives:**
  - Develop a deep learning model to accurately estimate depth from monocular images.
  - Evaluate the model's performance using established metrics like RMSE and loss curves with respect to different weights of SILog, SSIM, Edge depth loss and L1 loss in a combined loss function..
- **Deliverables:**
  - A trained U-Net model for depth estimation.
  - A comprehensive evaluation report on the model's performance.
  - A documented codebase with instructions for training and testing the model.
  - A final presentation summarizing the project findings.
- **Tasks and Activities:**
  - Data collection and preprocessing.
  - Model design and implementation.
  - Training the model using the dataset.
  - Evaluating the model's performance.
  - Fine-tuning the model and optimizing hyperparameters.
  - Documenting the process and results.
- **Requirements:**
  - A labeled dataset of images with corresponding depth maps (e.g., DIODE dataset)
  - A computational environment capable of training deep learning models (e.g., Google Colab).
  - Software libraries such as TensorFlow, Keras, matplotlib and OpenCV.
- **Boundaries and Exclusions:**
  - The project will not include multi-view or stereo depth estimation techniques.
  - The project will not cover hardware implementation or real-time depth estimation.
- **Resources:**

  - Google Colab is used as a high-performance computing environment required for training and testing dataset.
  - Software tools for data augmentation and model training.

- **Timeline:**
  - Week 1-2: Select a topic(Monocular depth estimation) to work on for project and present a project proposal.
  - Week 3-4: Literature review and deep dive into monocular depth estimation methodologies.
  - Week 5-6: Documentation of literature review and list possible methodologies. Selection of dataset to be used in the project.
  - Week 7: Designing model and selection of training parameters.
  - Week 8: Model fine-tuning and optimization.
  - Week 9: Final evaluation and documentation.
  - Week 10: Presentation preparation and delivery.

- **Assumptions:**
  - Available dataset is sufficient for training, evaluation and draw reasonable conclusions.

- **Constraints:**
  - The project must be completed within a 10-week timeframe.
  - The budget for computational resources is limited.

- **Future Work:**
  - Model performance can be enhanced by training it with an existing pre-trained model like ResNet and DenseNet
  - Larger dataset with further optimization of loss function may also provide better performance.

- **Github repository** :https://github.com/rish94abh/CS231A

References

[1] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *arXiv:1811.06152v1*, 2018. 1

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

[3] Yan Joe Lee, Jiho Hong, and Nayeun Lee. Image and depth retrieval from single monocular images with various point spread functions. In *Submitted as a project report for Stanford EE367*. Stanford University, 2020. 2

[4] Y. Liang, Z. Zhang, C. Xian, and S. He. Delving into multi-illumination monocular depth estimation: A new dataset and method. In *IEEE Transactions on Multimedia*, 2024. 1

[5] Abhinav Sagar. Monocular depth estimation using multi scale neural network and feature fusion. In *arXiv:2009.09934v1*, 2020. 3

[6] S.Gur and L. Wolf. Single image depth estimation trained via depth from defocus cues. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019,*, pages pp. 7675–7684, 2019. 1

[7] Hyeon Bo Shim, Jaehyung Ahn, Inseok Park, Souk Kim, and Younghoon Sohn. 3d gray level index for pattern depth monitoring based on sem image. In *Proc. SPIE 12955, Metrology, Inspection, and Process Control XXXVIII, 1295511*, 2024. 1

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. In *IEEE transactions on image processing, 13(4):.*, page 600–612, 2004. 1

[9] Y. Yang, Y. Wang, C. Zhu, M. Zhu, H. Sun, and T. Yan. Mixed-scale unet based on dense atrous pyramid for monocular depth estimation. In *IEEE Access*, page 9:114070–114084, 2021. 1