Project Report - ECE 176

Benjamin Liang
Mathematics
A16954232

Andrew Onozuka
Computer Engineering
A16760043

Abstract

Convolutional Neural Networks have been the backbones of computer vision tasks such as depth perception of an image. This is useful to determine how far an object is from everything else in its field of view by determining the depth of each object in a specific image. This is the task that we aim to explore more of which is crucial in computer vision as we discover more about autonomous vehicles and computer vision. This project, we will use a simple UNet model to determine the depth of each pixel in an image from the NYUv2 Depth dataset. We will particulary focus on the setting of a living room to determine how accurate we are able to predict the depth from each position in a living room.

1 Introduction

With the every growing advancements of AI and ML in autonomous vehicles, the problem we hope to solve is the detection of depths of images which would be useful to autonomous vehicles to determine how far they are from everything else in their field of view. There have been many works around this topic with different implementations being used that vary from different deep learning architectures to the features being used to even the loss function within the deep network. We would like to explore different solutions to deepen our understanding of neural networks and how it can be applied to surface normal predictions and depth perception. Although, these experiments may not result in the best performance, we hope to better understand the model used in cutting edge research around this topic.

One of the topics we would like to get deeper into is the role of different models on this depth perception task on the NYU dataset. There have been many works around this topic and a lot of the more successful models utilize convolution layers along with semantic segmentation to get the lowest possible error.

In this research paper, we will explore the UNet model which is similar to having an encoder and decoder to downsample the input image and upsample it to the label image. We decided to explore this model deeply because, UNet is useful for image generation and image impainting. We found similarities with these two tasks as depth perception is essentially just determining how far a specific pixel is relative to the rest of the image.

2 Related Work

In this section, we examine pertinent literature related to our project methodology. We focus on two seminal papers that directly inform our research direction.

One relevant study is "Multi-Scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation" by Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, Nicu Sebe. This paper delves into the task of monocular depth estimation on the same dataset with multi scaled fusion. The authors propose a novel approach that leverages fusion of of outputs from previous layers which utilizes CRFs to output an accurate depth estimation of the input image. Our project draws inspiration

from this work as we aim to incorporate similar techniques for depth estimation by utilizing the idea of a Front CNN module.

Another significant paper in our field is "PolyMax: Solving Maximization and Minimization Through Polynomials" by Yang et al. This paper introduces the PolyMax framework, a versatile approach for solving maximization and minimization problems using polynomials. While our project focuses on depth estimation and surface normal prediction, we find relevance in the methodology proposed in PolyMax for its innovative optimization techniques. By integrating insights from PolyMax, we aim to enhance the efficiency and robustness of our proposed model for dense prediction tasks.

These two papers [3], [1] and [4] serve as foundational references that guide our research approach, providing valuable insights and methodologies that inform the development of our project.

3 Method

In this section, we describe our approach for depth estimation and surface normal prediction using convolutional neural networks (CNNs). Our method involves several components, including data loading, model architecture, training algorithm, and testing procedure.

3.1 Data Loading

We utilize a custom dataset class, `CustomImageDataset`, to load the NYUv2 RGB-D dataset [2]. This class reads image files and corresponding depth maps from the dataset directory and applies transformations as specified. We use PyTorch's DataLoader to create batches of training and test samples for efficient processing.

3.2 Model Architecture

Our proposed model, named `DepthBaseModel`, is a convolutional neural network designed for depth estimation. The architecture consists of several convolutional layers followed by max-pooling operations to downsample the input image. Batch normalization and ReLU activation functions are applied after each convolutional layer to introduce non-linearity and stabilize training. The final layer produces a single-channel output representing surface normals.

3.3 Training Algorithm

We train the model using mean squared error (MSE) loss as the optimization criterion. The Adam optimizer is employed with a learning rate of 0.001 to update the model parameters. During each epoch, we iterate over the training dataset in mini-batches, compute the loss between predicted and ground truth depth maps, and update the model weights through backpropagation.

3.4 Testing Algorithm

After training, we evaluate the performance of the trained model on the test dataset. We iterate over the test samples and compute the predicted depth maps using the trained model. We then compare these predictions with the ground truth depth maps to assess the model's accuracy.

3.5 Proposed Techniques

Compared to previous works, our method introduces several novel techniques:

1. **DepthBaseModel Architecture**: We design a custom CNN architecture tailored for depth estimation tasks. By incorporating multiple convolutional layers followed by max-pooling operations, our model can effectively capture spatial features and learn hierarchical representations from input images.
2. **Interpolation for Ground Truth Alignment**: To align the ground truth depth maps with the predicted outputs, we apply interpolation to resize the ground truth maps to the same dimensions as the model predictions. This ensures consistency in the loss computation and improves the convergence of the training process.

3. **Batch Normalization and ReLU Activation**: We utilize batch normalization and ReLU activation functions after each convolutional layer to accelerate training convergence and prevent vanishing gradients. These techniques enhance the stability and efficiency of the optimization process.

By incorporating these techniques, we aim to achieve superior performance in depth estimation and surface normal prediction tasks compared to existing methods. The modular design of our model allows for flexibility in experimentation and adaptation to different datasets and application scenarios.

4 Experiments

The datasets that we will be using is solely the NYUv2 Depth dataset. This dataset contains the input images and label images in directories for each datapoint. The path to each of these files can be read in a dataframe from the csv files provided for both the train and the testing data. The images were randomly chosen from a video according to [2] the description of the dataset. The input image for each scene contains 3 channels and the label image is a grayscale image which only has 1 channel. We will make use of the UNet architecture to predict the depth of each image.

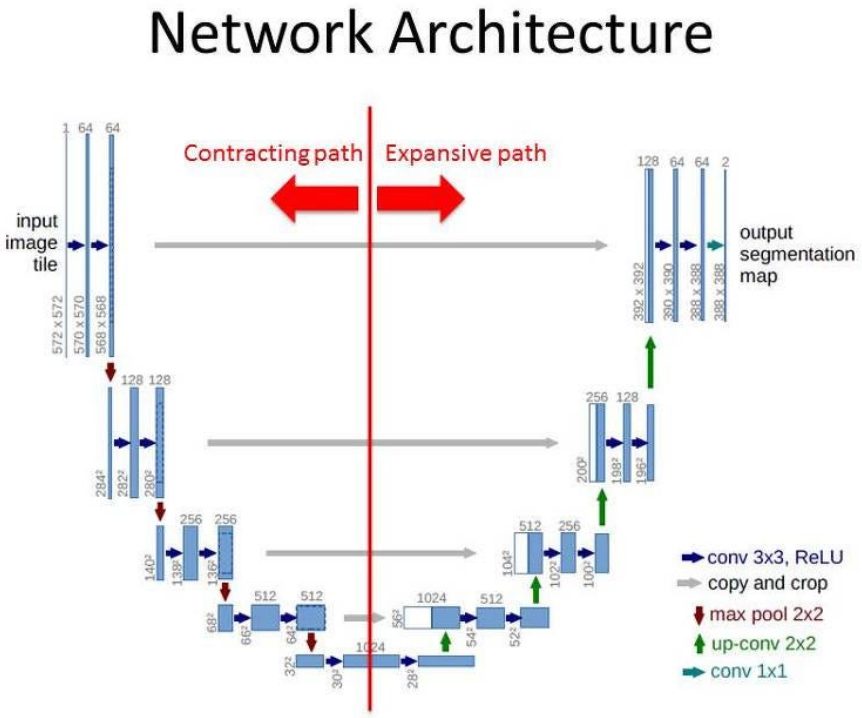

Figure 1: UNet architecture for Depth estimation

To evaluate the accuracy of our model we will use the Mean Squared Error between every pixel of the ground truth image and our predicted labels. This can result to a high MSE as our images are 480x600 pixels. We can expect a high MSE but relatively low for our task. In 1 epoch of our training, we received an error of 2093.7061 and test loss of 2395.8513.

We can see that the model does fairly well around the edges, correctly classifying the label as close by shading it a darker pixel but fails to do so around the further pixels. We can see on the left that the model attempts to distinguish the depth at the window by having lighter pixels but it could be due to other factors such as lighting as well. This is a hard task since if any other white shades occurred in the original image, it would be harder to classify them as closer or further away.

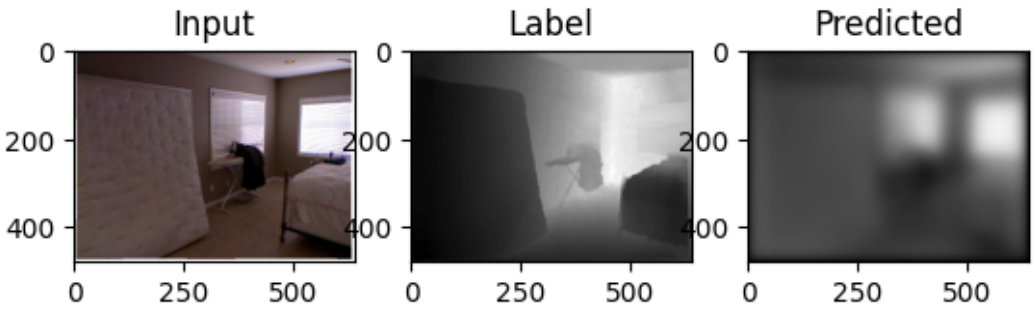

Figure 2: Results of input image, ground truth, and predicted labels on first test point

To combat this, we can definitely expand our model to include other images from outside of this dataset as well. Including more images would definitely help the training of this model. Although, training will take a tremendous amount of time since every image is 480x600 pixels which is a ton of features for a single model. To reduce training time we can increase batch size or even downscale the image to be smaller so we have less pixels to train. Another thing we could explore is adding more decoders and encoders in the UNet. This would allow the model to effectively capture features and learn hierarchical representations in the image that would better help the performance of this model.

For comparison, other research groups have had extensive research on this topic and their predicted labels from their models accurately predicts the depth representation of the input image

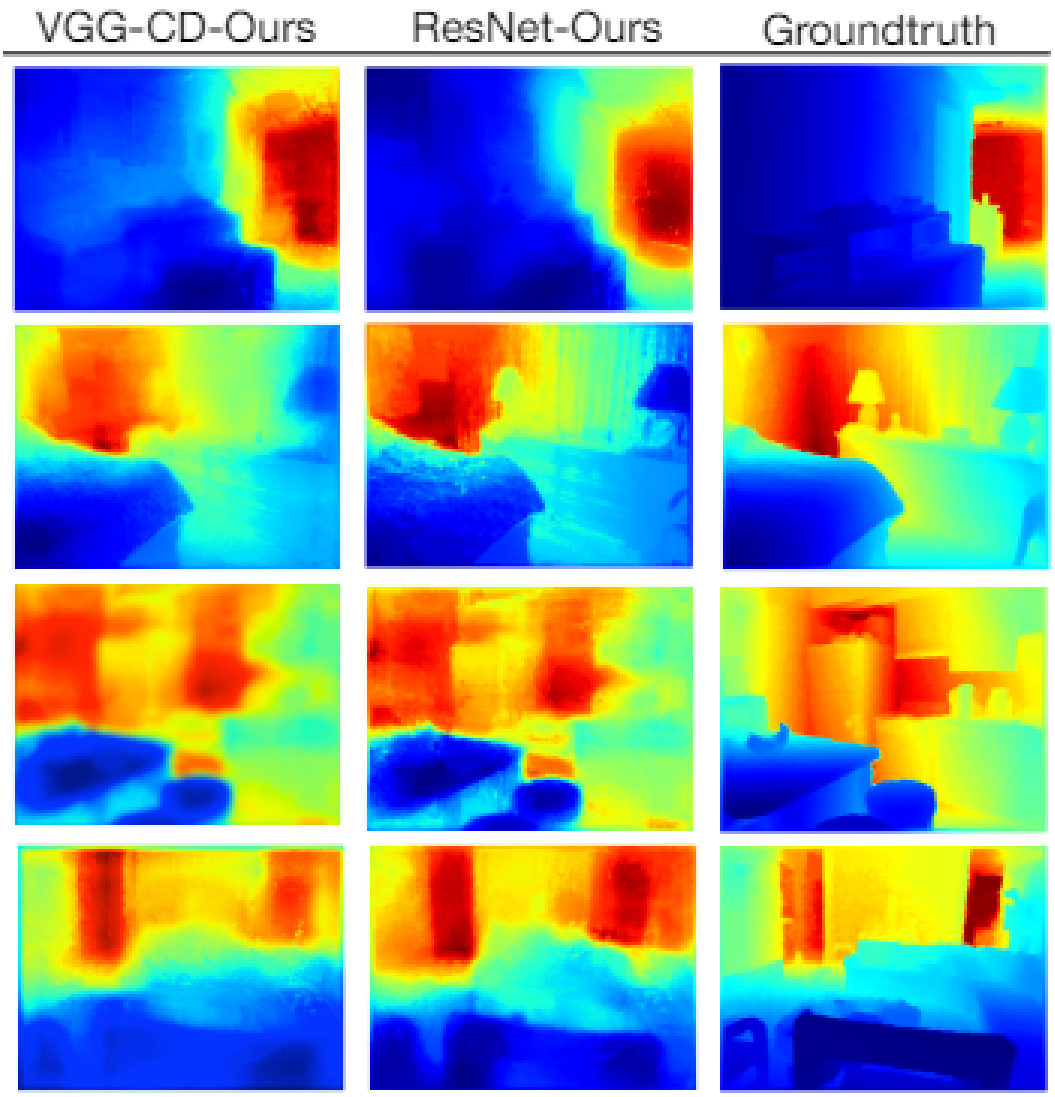

Figure 3: Labels for other models in [3] and the ground truth

References

[1] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *arXiv preprint arXiv:1704.02157v1*, 2017.

[2] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Proceedings of the European Conference on Computer Vision (ECCV), 2012. `https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html`.

[3] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4321–4329, 2017.

[4] Xuan Yang, Liangzhe Yuan, Kimberly Wilber, Astuti Sharma, Xiuye Gu, Siyuan Qiao, Stephanie Debats, Huisheng Wang, Hartwig Adam, Mikhail Sirotenko, et al. Polymax: General dense prediction with mask transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1050–1061, 2024.