A 3D Fully Convolutional Residual Neural Network for Robust Depth Prediction from Monocular Images

A dual degree project report

submitted by

ROHIT KANOJIA

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
&
MASTER OF TECHNOLOGY

INTERDISCIPLINARY PROGRAM IN COMPUTATIONAL ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

JUNE 2021

THESIS CERTIFICATE

This is to certify that the thesis entitled **A 3D Fully Convolutional Neural Network for Robust Depth Prediction from Monocular Images**, submitted by **Rohit Kanojia (EE16B111)**, to the Indian Institute of Technology Madras, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Sharma 1/7/2021

Dr. Mansi Sharma
Research Guide
INSPIRE FACULTY
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 1st June 2020

ACKNOWLEDGEMENTS

I would like to thank several individuals who in one or other way contributed and extended their valuable assistance throughout the course of this project. I owe my debt of gratitude to the Indian Institute of Technology Madras for giving me the opportunity to come here and work with some of the brightest minds in the world. I would like to thank Dr. Mansi Sharma for guiding me through this DD project and helping on this work.

ABSTRACT

The depth information is one of the most critical importance in scene understanding for several industrial projects such as Self driving cars, Robotics for instance. Inferring depth from a single image has taken a prominent place in recent studies With the outcome of deep learning methods. Deep learning based solutions for computer vision problems outperforms other solutions by a far margin. Depth estimation is one such task that has been tremendously improved by the advent of DL.

This thesis proposes two deep fully convolutional networks for monocular depth prediction. First architecture is based on using Bilateral grid for edge-aware depth prediction while other architecture has a u-net parallel to CNN to infer a sharp geometric layout of the scene.

The proposed depth prediction model archives state-of-the-art performance in both qualitative and quantitative evaluation on NYUv2 -Depth Dataset.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Motivation

In the past few years many new industrial projects in the area of Robotics , self-driving cars, were the core of the problem to retrieve spatial information from a single sequence of images. Capturing depth information from images is a non-trivial task in computer vision which has been explored in the past decade. However most classical algorithms need pairs of stereoscopic images to achieve intermediate performance. The outcome of the deep learning method allowed to improve this field of research, in terms of research. Also due to the availability of large amounts of RGB-D(color and depth) data collected with dedicated depth sensors. Deep learning methods allow to solve ill-posed problems such as depth prediction from a single image. The algorithm that rely on supervised learning or self-supervised learning approaches (Clément Godard (ICCV 2019) , Fabio Tosi (CVPR 2019), Dan Xu (CVPR 2018), Jamie Watson (ICCV 2019)).

1.2 Problem Definition

This thesis will focus first on a review of the different pre-existing approaches for depth prediction from a single image. Then we focus on explaining proposed architectures for monocular depth prediction. Finally we compare our architectures with the existing approaches in order to understand the qualitative and quantitative point of view.

CHAPTER 2

Related Work

Most of the recent deep learning approaches to tackle the problem of depth prediction use only a single image, instead of two images as it is the case with classical approaches.

2.1 Eigen Network

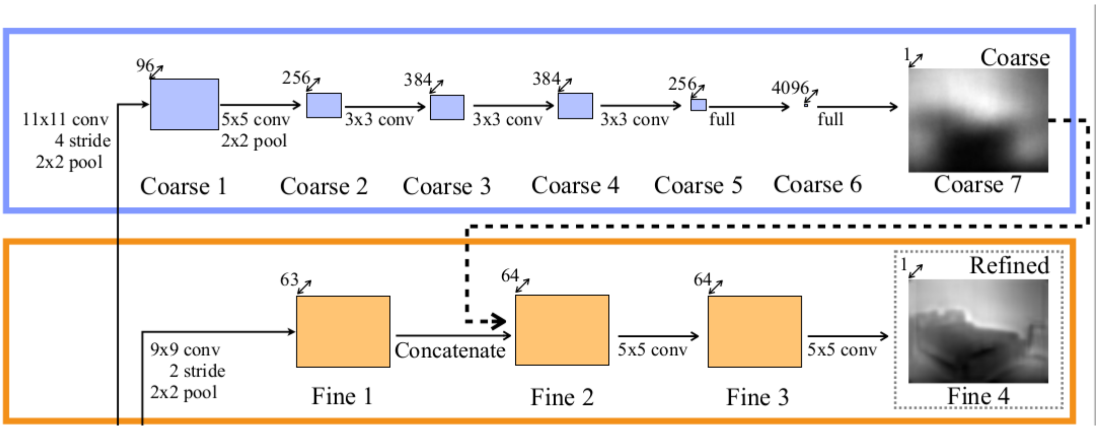


Figure 2.1: Architecture of Eigen network

The first network which tackled the problem of depth prediction was published by and Fergus David Eigen. The Figure 2.1 details the composition of this network. It can be decomposed in two small networks, the first one in blue in Figure 2.1 aims to predict a coarse depth map. The second one in orange, refines the coarse result. Several differences can be observed between those two. The first difference is the presence in the coarse network of two fully connected layers. These layers perform operations which links every coefficient of the features. The local specificity will be lost but global structure will be observed. The second network performs only several convolutions with a small kernel to refine the information locally. This network was the first major contribution for monocular depth prediction using deep neural networks.

2.1.1 Results of Eigen Network on NYU dataset

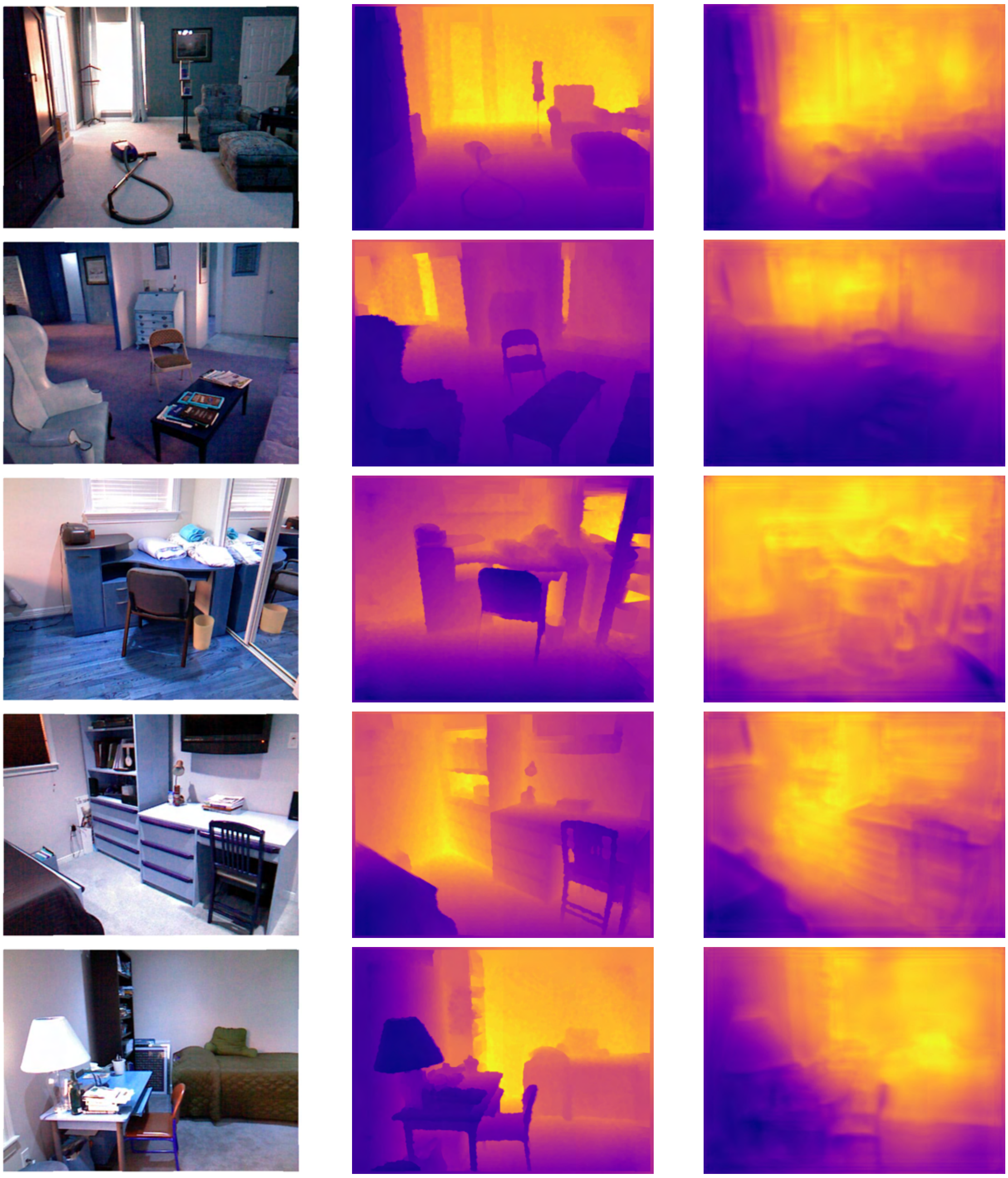


Figure 2.2: Eigen Network Results(1). (left) Input image (middle) ground truth (right) prediction

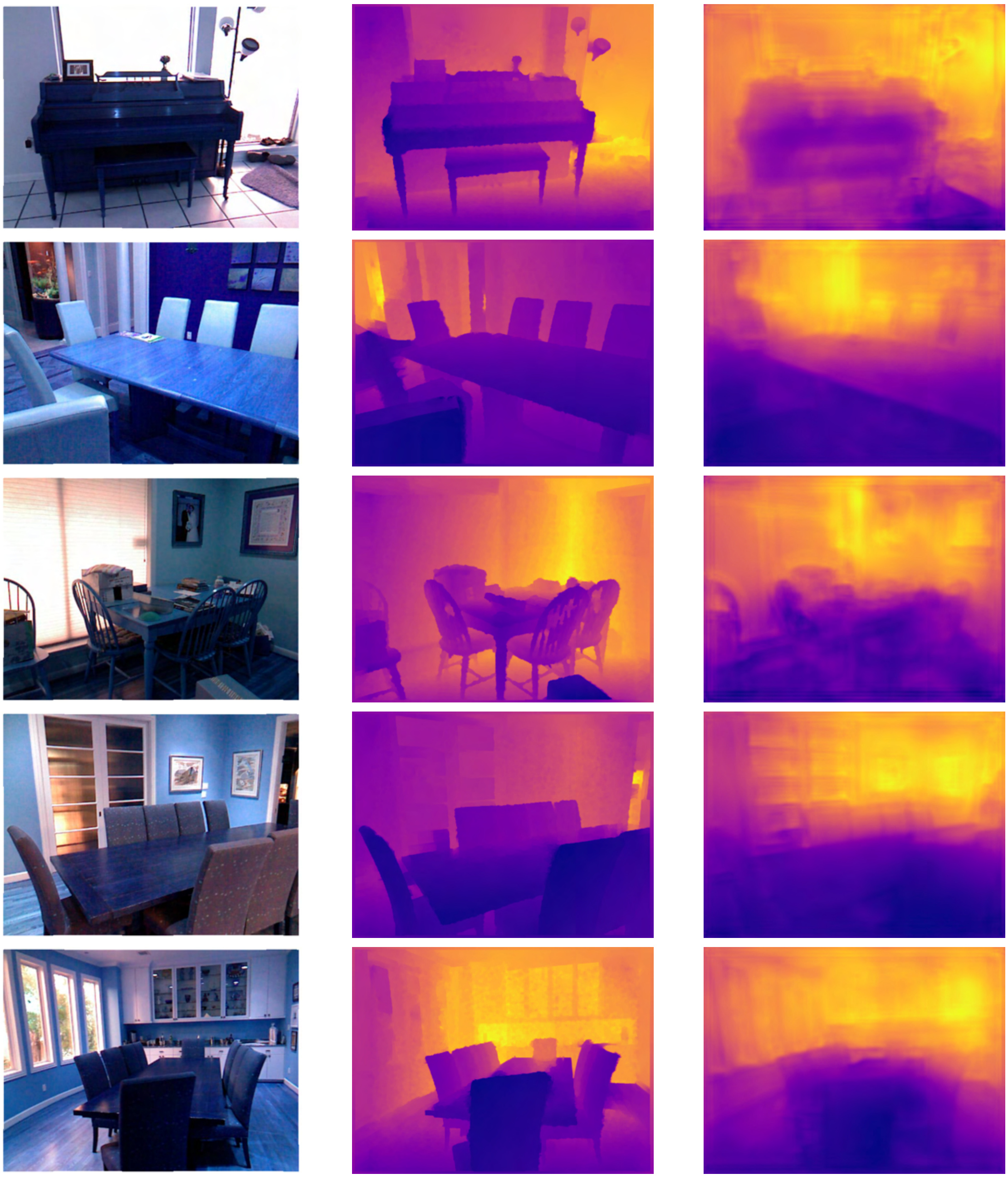


Figure 2.3: Eigen Network Results(2). (left) Input image (middle) ground truth (right) prediction

2.2 Laina Network

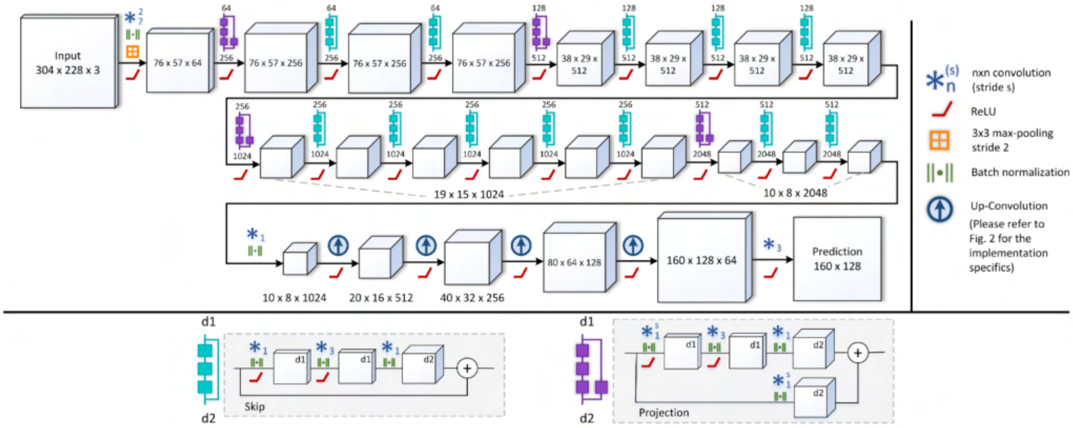


Figure 2.4: Architecture of Laina network

The state-of-the-art network for monocular depth prediction has been implemented by Laina Laina *et al.* (2016). Figure 2.4 illustrates the microscopic composition of it. It presents a classical encoder-decoder structure. The encoding operation is performed by ResNet50. The decoding operation is specific to this network. The basic idea is to reduce the number of features during the upsampling. To do so, Laina Laina *et al.* (2016) introduces a new module called up-projection. This module performs first an unpooling in order to increase the width and height of the features and then convolutions to decode the information by reducing the number of features. The unpoling operation consists in filling the value of a feature map in a two times larger feature map with only zeros. The coefficients are filled only one column on two and one row on two. This network architecture is quite simple in the sense that it inputs a single image and outputs its depth map. It achieves state-of-the-art results on several academic datasets such as NYU depth dataset Nathan Silberman and Fergus (2012) and justify the encoding-decoding approach for depth prediction. All the following will use similar architecture.

2.2.1 Results of Laina Network on NYU dataset

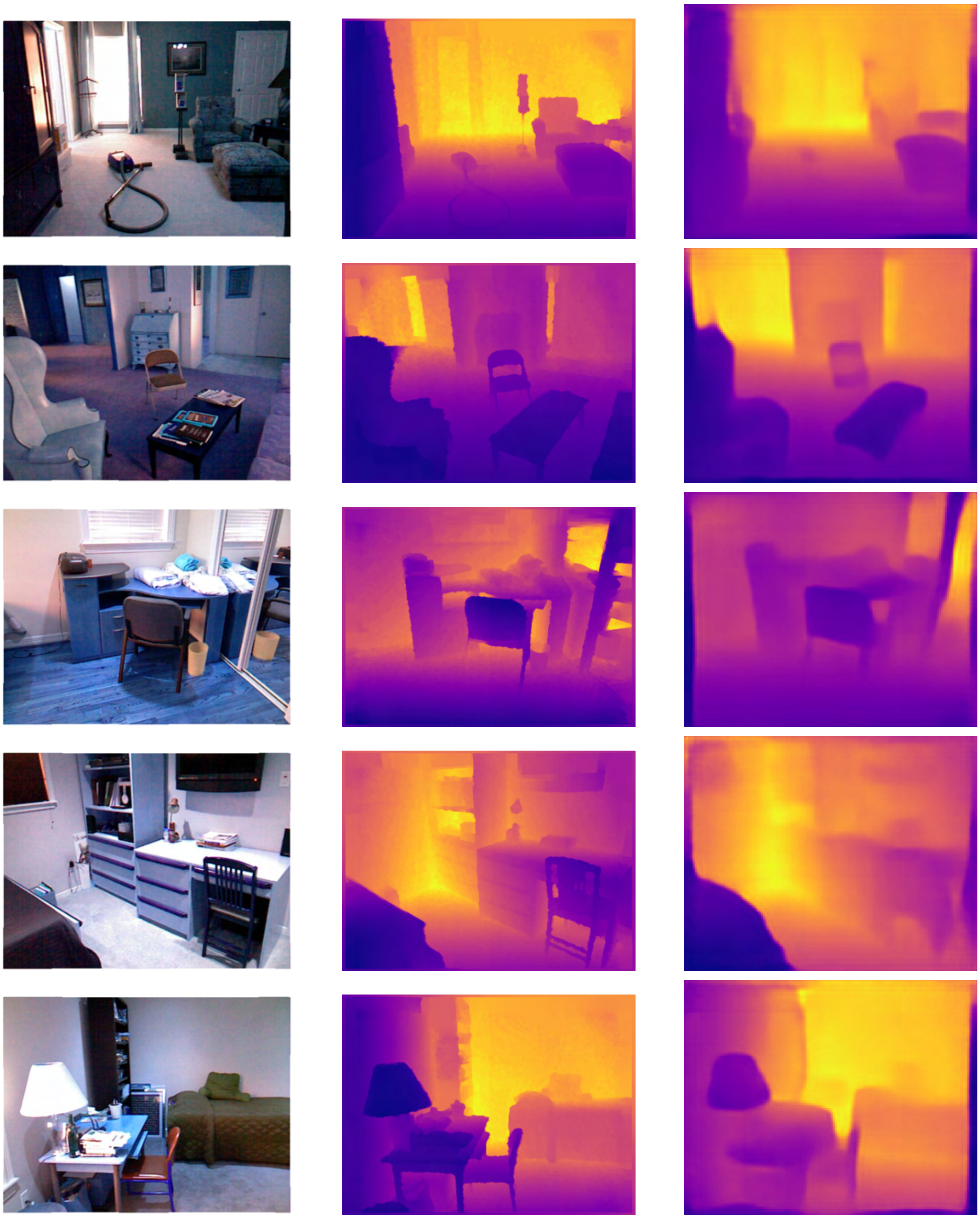


Figure 2.5: Laina Network Results(1). (left) Input image (middle) ground truth (right) prediction

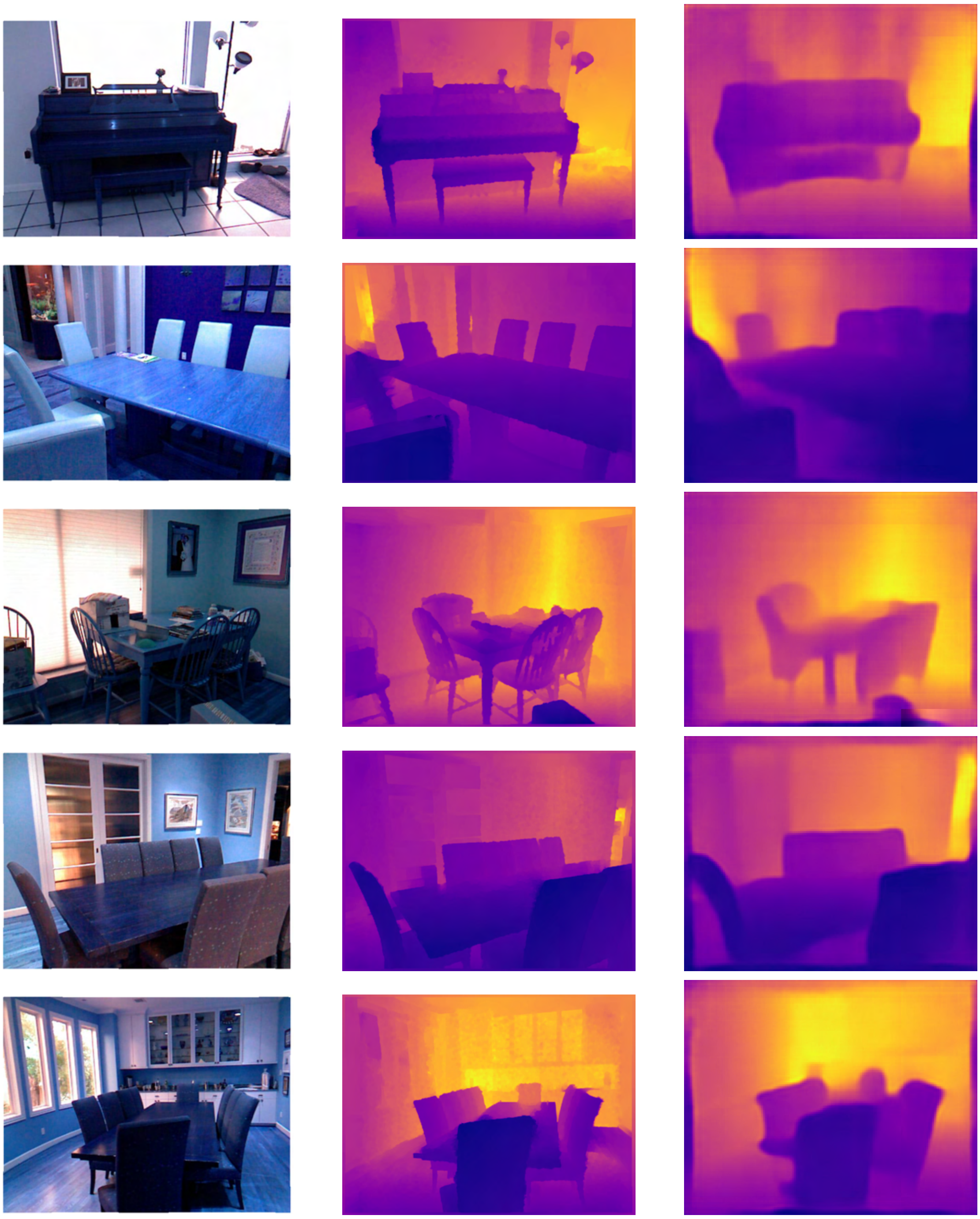


Figure 2.6: Laina Network Results(2). (left) Input image (middle) ground truth (right) prediction

2.3 Sharpnet Network

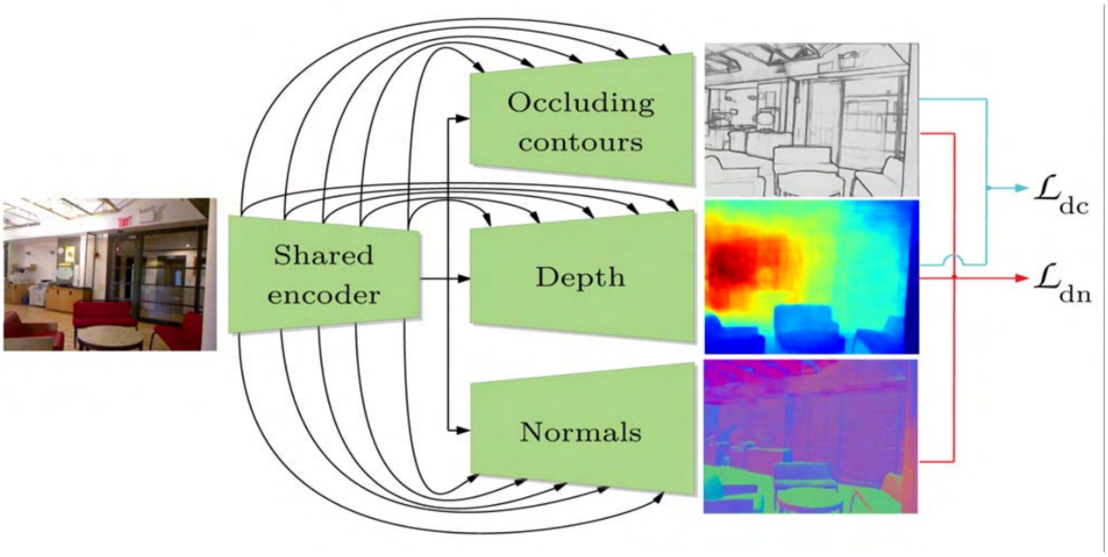


Figure 2.7: Architecture of SharpNet network

Shapnet Ramamonjisoa and Lepetit (2019) method that predicts an accurate depth map for an input color image, with a particular attention to the reconstruction of occluding contours: Occluding contours are an important cue for object recognition, and for realistic integration of virtual objects in Augmented Reality, but they are also notoriously difficult to reconstruct accurately. Main challenge for stereo-based reconstruction methods, as points around an occluding contour are visible in only one image. Inspired by recent methods that introduce normal estimation to improve depth prediction, they introduce a novel term that constrains depth and occluding contours predictions. Since ground truth depth is difficult to obtain with pixel-perfect accuracy along occluding contours, synthetic images for training are used , followed by fine-tuning on real data and demonstrations on NYU-V2 datasetNathan Silberman and Fergus (2012).Figure2.7 contains the Architecture of SharpNet network.

2.3.1 Results of SharpNet Network on NYU dataset

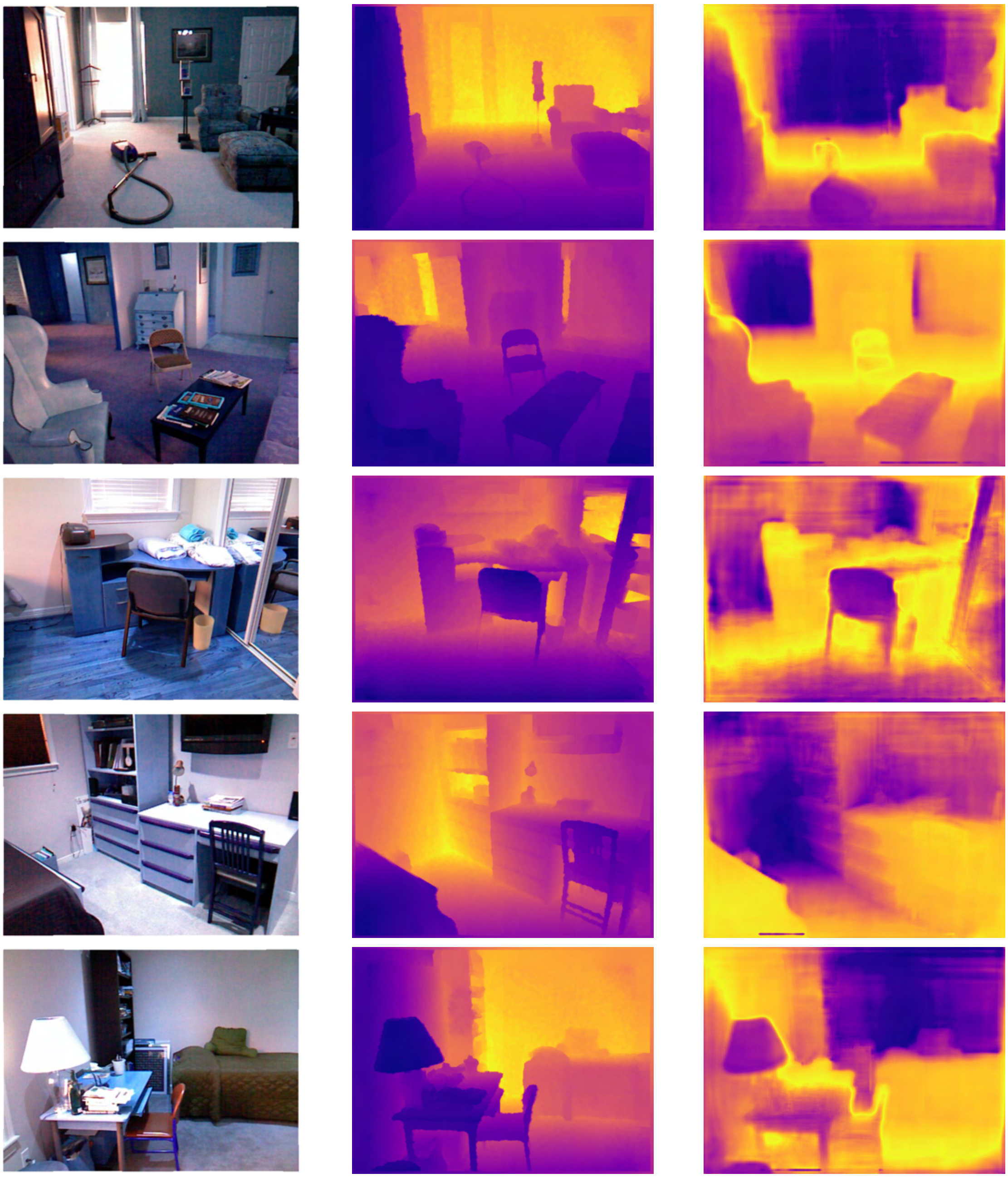


Figure 2.8: SharpNet Network Results(1). (left) Input image (middle) ground truth (right) prediction

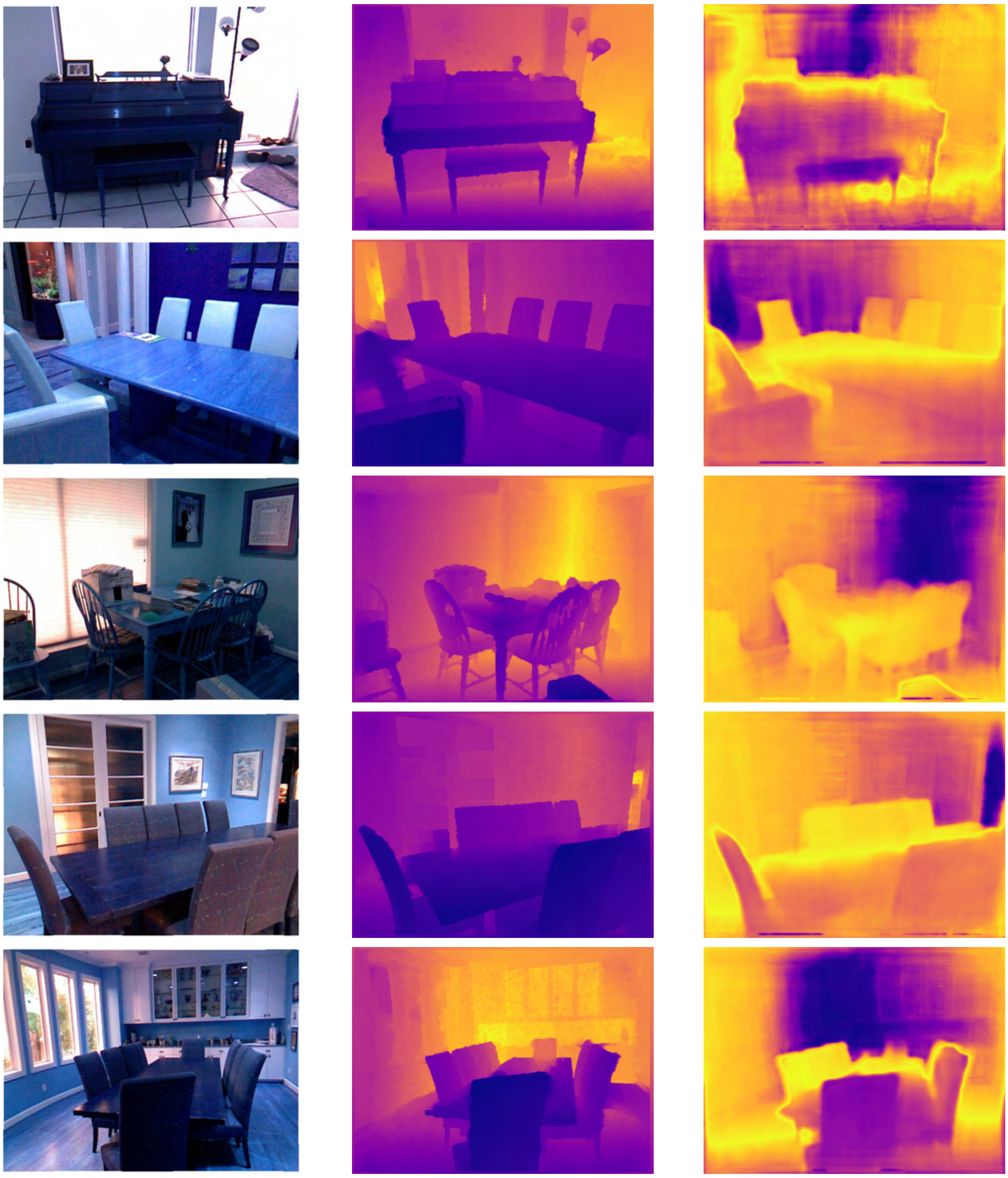


Figure 2.9: SharpNet Network Results(2). (left) Input image (middle) ground truth (right) prediction

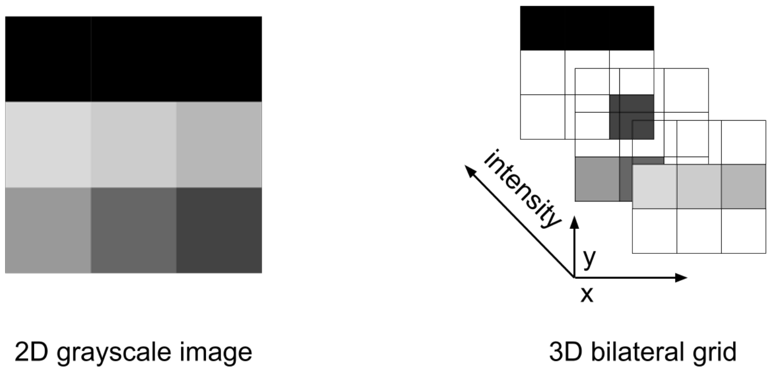
CHAPTER 3

Fully Convolutional 3D Bilateral Grid Based CNN Network For Edge Aware Depth Prediction.

3.1 Method

This model is based on the idea of using a Bilateral grid as an input to our Architecture. Bilateral grid is a data structure used for image processing applications. Bilateral grid is inspired by the bilateral filter.The bilateral grid enables edge-aware image manipulations such as local tone mapping on high resolution images in real time. In general, the bilateral grid is used in three steps. First, we create a grid from an image or other user input. Then, we perform processing inside the grid. Finally, we slice the grid to reconstruct the output. Construction and slicing are symmetric operations that convert between image and grid space.This idea was introduced in Mansi Sharma (2021)

3.2 Bilateral Grid



2D grayscale image

3D bilateral grid

Figure 3.1: 2D image represented as a 3D bilateral grid

Bilateral grid was introduced in Chen *et al.* (2007) as a data structure for images to perform edge aware computations on them. A bilateral grid is a 3D representation of a

2D image that separates pixels not only by their spatial position but also their respective intensity values.

Let $I(x, y) = z$ be a gray scale image where x, y are the pixel indices and z is the intensity value, its corresponding bilateral grid is given by

$$BG(x, y, z) = z \quad \forall x, y, z \in I \tag{3.1}$$

Any 2D operation on image space, becomes a 3D operation in the bilateral space. This is the motivation behind the fact that the network we trained is a 3D CNN.

3.2.1 Bilateral Grid for CNN

Since images can be quite large and adding a 3rd dimension quickly blows up the size, we usually reduce spatial resolution of the bilateral grid by a factor of 2 or 4 (sometimes even 10 if the image is of very high resolution). The choice of intensity dimension is usually 32 or 64.

Also on further experimentation, we found it was easier for the network to learn if instead of $BG(x, y, z) = z$, we use

$$BG(x, y, z) = 1 \quad \forall x, y, z \in I \tag{3.2}$$

thus making gradient values during back propagation equal for all intensity levels and thus learning for all intensities at equal rates.

3.2.2 Representing color images as a bilateral grid

An interesting issue that arises is that only gray scale images can be represented as a bilateral grid. If we were to represent color images we would need a 5 dimensional grid with extra 3 dimensions for RGB values. We can work around this by creating 3 bilateral grids one for each color channel. Even in this case there are 2 ways of representation. Let BG_r, BG_g, BG_b be the bilateral grids corresponding to each color channel and let I be the image to be converted.

$$I(x,y) = (r,g,b) \tag{3.3}$$

$$BG_r(x,y,r) = 1,\ \ BG_g(x,y,g) = 1,\ \ BG_b(x,y,b) = 1 \tag{3.4}$$

$$BG_r(x,y,s) = r/s,\ \ BG_g(x,y,s) = g/s,\ \ BG_b(x,y,s) = b/s \tag{3.5}$$

$\forall x,y,r,g,b \in I$ and where, $s = (r+g+b)$

From experiments, we found that the representation in (3.5) proved to be the best. The models learned faster and were able to generalise to different scenes in images.

3.2.3 Why Grids

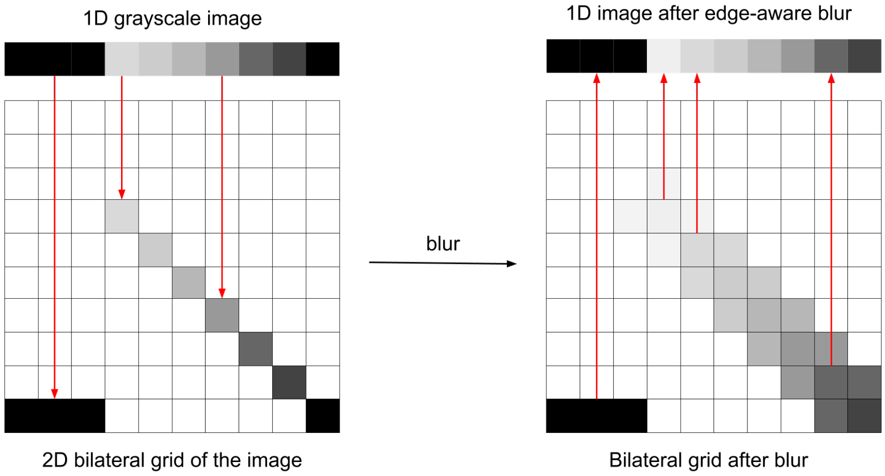

1D grayscale image
1D image after edge-aware blur
blur
2D bilateral grid of the image
Bilateral grid after blur

Figure 3.2: How bilateral grids preserve edges

Bilateral grids offer a simple yet robust way to preserve and retrieve edges present in the original RGB image after performing computations. Consider the example of blurring an image. When we use a Gaussian kernel to blur it affects all neighbourhood pixels equally eventually destroying sharp edges. A Gaussian blur in N Dimensions is treated as a Gaussian blur in $N+1$ dimensions when using bilateral grids. Thus, 2D Gaussian blur becomes 3D and so does the kernel. Edges by definition mean sharp rise or drop in intensity values. Since bilateral grids separate pixels by their intensity values too, the kernel never affects the pixels on the other side of the edge. This is apparent in 3.2. Notice how the first 3 black pixels are untouched and the separation remains sharp even after the blur operation has been performed.

3.3 Proposed CNN architecture

In this section, we explain our proposed convolutional network, Like most of the CNN architectures our model contains the contractive part which progressively decreases the input through a series of convolution and pooling operations followed by upsampling. First the image is Converted into a bilateral grid then it passes through the network and output is the predicted image. Such a deep network helps in capturing large information. Figure 3.3 contains the visual representation of the architecture.

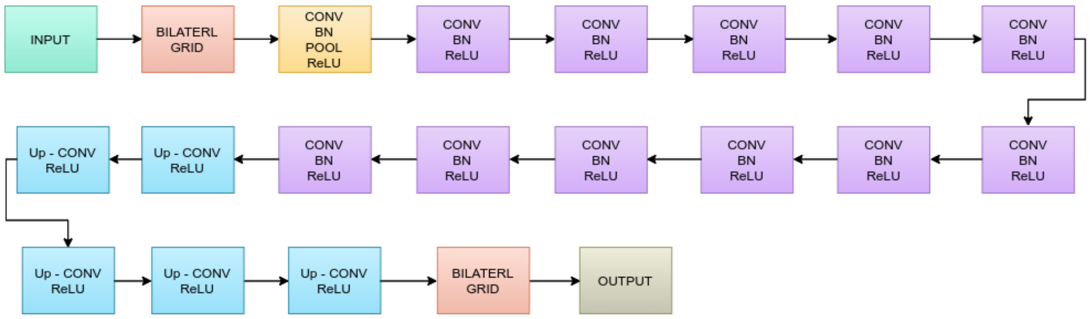


Figure 3.3: Fully Convolutional 3D Bilateral Grid Based CNN Network For Edge Aware Depth Prediction

3.4 Loss function

We have have used reverse Huber(berHu) as our loss function. The BerHu function is equal to $L1$ norm when $x \in [-c, c]$ and $L2$ norm outside this range.

$$B(x) = \begin{cases} |x| & |x| \leq c \\ \frac{x^2+c^2}{2c} & |x| > c \end{cases}$$

We set $c = \frac{1}{5} max_i(|\hat{y}_i - y_i|)$ where i indexes all pixels over each image in the current batch.

3.5 Dataset (NYU-Depth V2)

The NYU v2- depthNathan Silberman and Fergus (2012) dataset improves upon the v1 in number of training samples, depth image quality and overall scene variations. It consists of around 120,000 RGB image and depth image pairs. Although the depth image needs to be obtained from a video stream after synchronisation. Due to the huge size of dataset (420 GB) we only use the readily available sample data (2.3 GB) consisting of around 2000 image pairs to train our model. All images are of the same resolution of (640×480). The depth data has been captured using a Kinect sensor, converted to image and then inpainted to handle the holes (missing data due to reflections and transparent surfaces).

3.6 Implementation details

We implemented our proposed depth prediction architecture using PyTorch. Both training and evaluation are performed on a single high-end HP OMEN X 15-DG0018TX Gaming laptop with 9th Gen i7-9750H, 16 GB RAM, RTX 2080 8 GB Graphics and Windows 10 operating system. Training part took around 8 hours(3 min per epoch). We have used stochastic gradient descent as an optimizer. Training code ran for 150 epochs.

3.7 Results

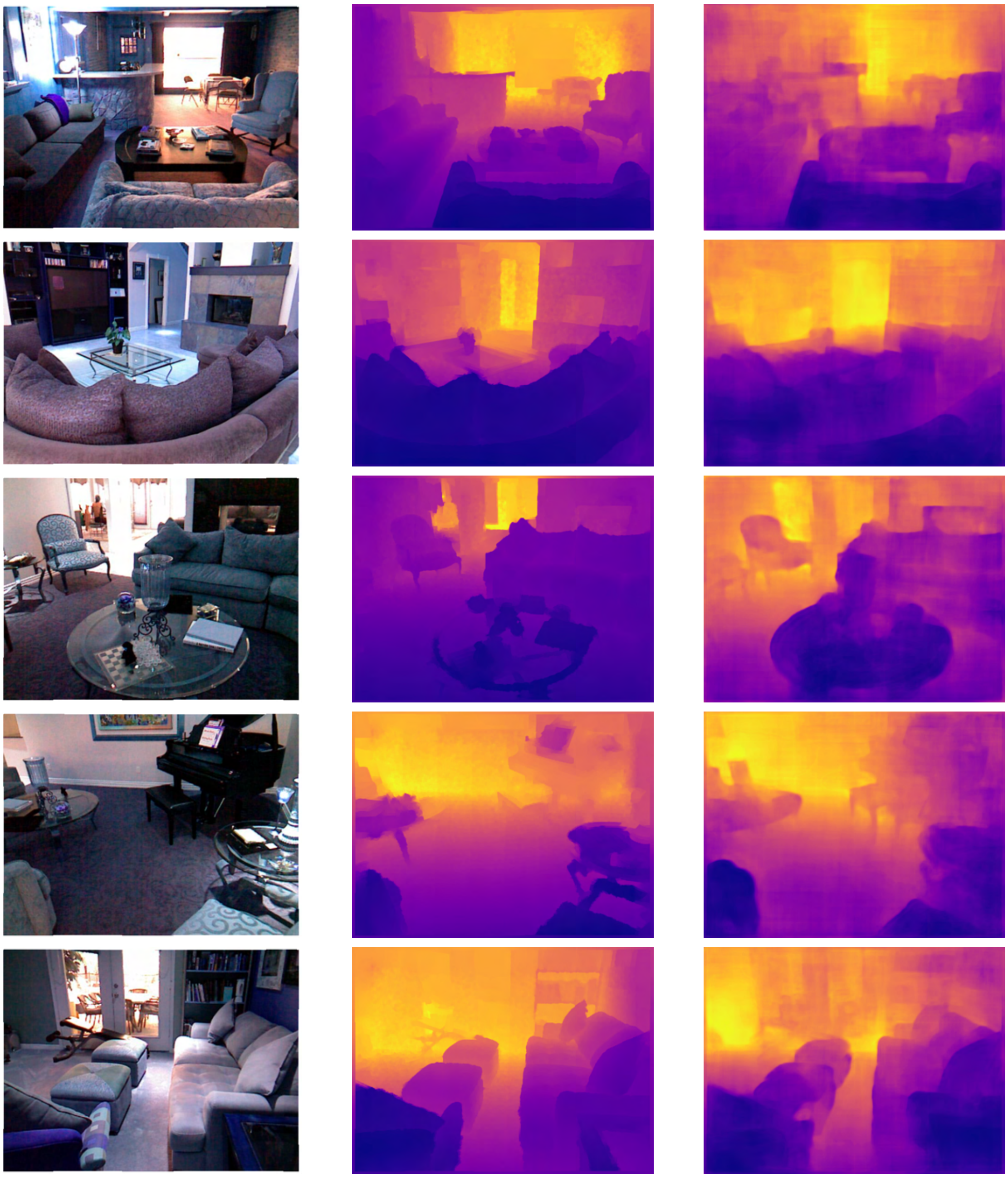


Figure 3.4: 3DBG-net Results(1). (left) Input image (middle) ground truth (right) prediction

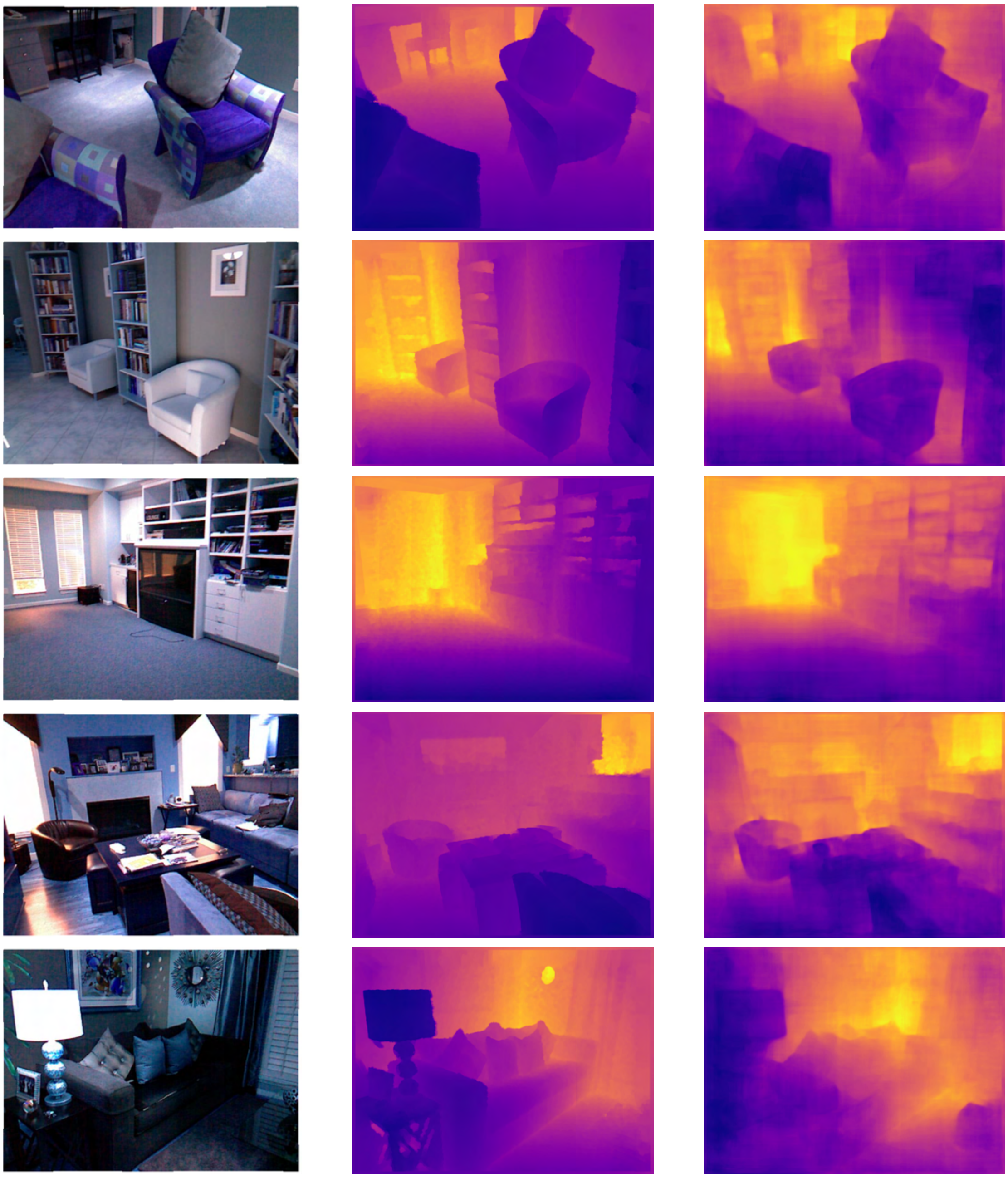


Figure 3.5: 3DBG-net Results(2). (left) Input image (middle) ground truth (right) prediction

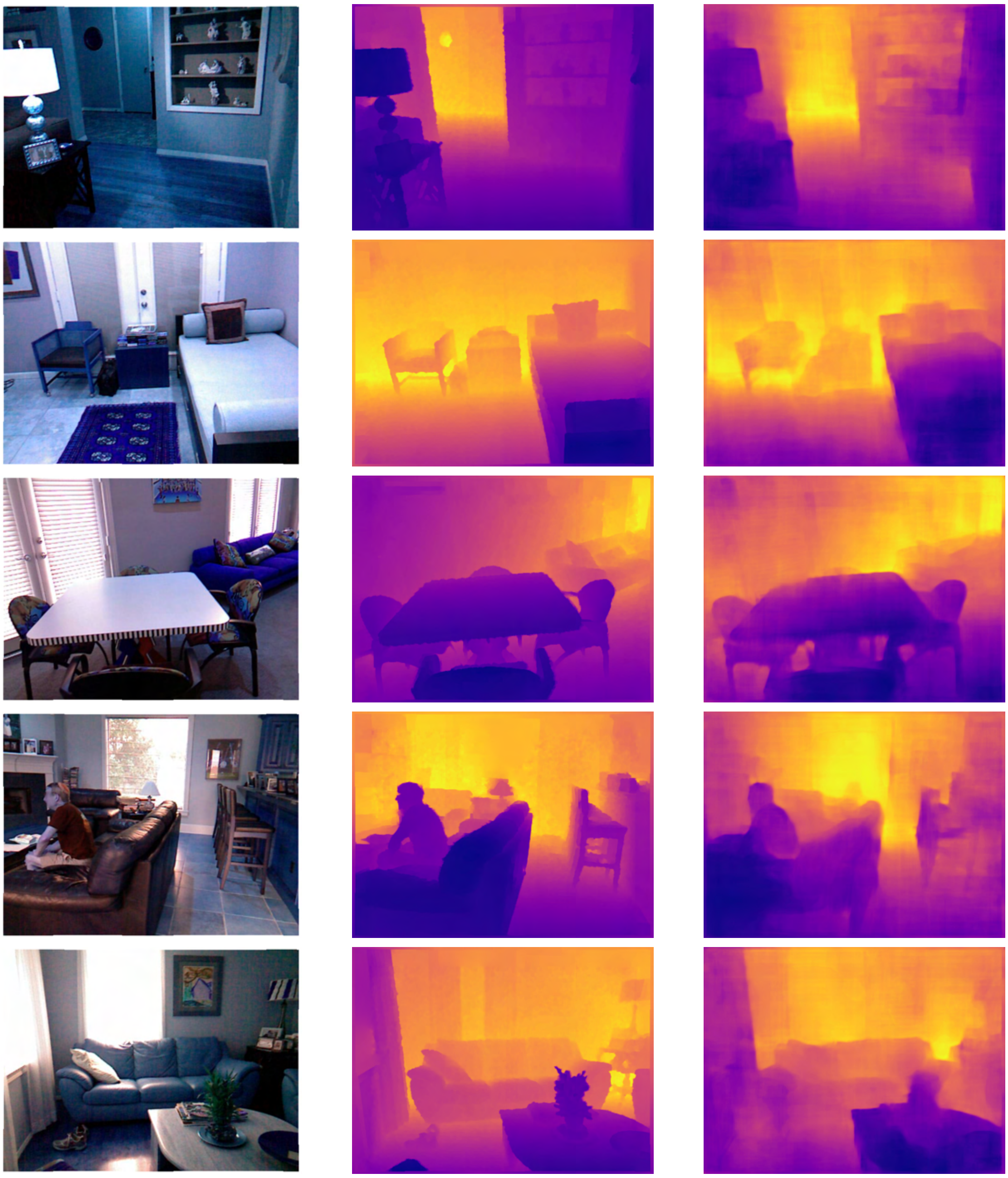


Figure 3.6: 3DBG-net Results(3). (left) Input image (middle) ground truth (right) prediction

CHAPTER 4

FC3D-ResUnet: A 3D Fully Convolutional Residual Neural Network for Robust Depth Prediction from Monocular Images.

4.1 Method

In this section, we explain our proposed convolutional network architecture, dubbed as (FC3D-ResUnet). The objective of the proposed architecture is to strike a balance between Depth map and a good sharp geometric layout of the scene.The input to the network is color rgb image and output is sharp depth maps. Like most of the current CNN architectures contains a contractive part that gradually decreases the input resolution through convolutions and pooling operations and thus capturing more global information. The desired output is a high resolution image in order to achieve that some upsampling is required. We have introduced a fully convolutional network for depth prediction. We have set input resolution to 640 X 480 pixels and produced output map resolution of 492 X 369 pixels.

By applying CNN directly does not produce feasible output as it lacks a sharp geometric layout. To overcome this we introduced a u-net in parallel that helps the network to infer a sharp geometric layout of the scene.Figure 4.1 contains the visual representation of the architecture.

4.1.1 Network

The network passes through convolution followed by batch normalization and pooling. This design is possible to create a much deeper network without degradation. Another property of such architecture is their large receptive field which is large enough to capture high resolution images. Given an input image it will create a feature map of 2048.The later part of the network focuses on a sequence of convolution and unpooling

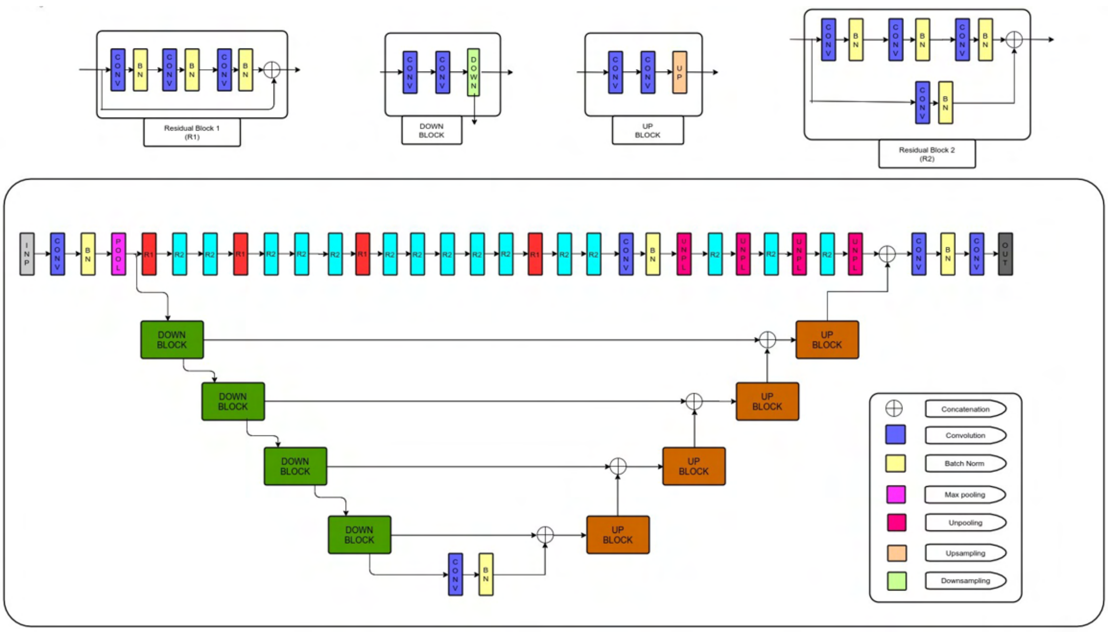


Figure 4.1: FC3D-ResUnet Architecture

layers which guides the network in learning upscaling. The output of the unet is then concatenated to the network followed by convolution and batch normalization.

4.1.2 U-Net

We have introduced a Unet style network that can infer the sharp geometric layout of a scene. The proposed unet works parallel to the network. The Upsampling and downsampling are used to upsize and downsize the features respectively. The input feature after the convolutional block and max pooling block is downsized four times. The output of the unet is concatenated to the original network.This helps in producing sharp geometric layout of the scene.

4.2 Loss function

Like most of the depth regression problem we consider the difference between the ground truth depth map Y and prediction of depth regression as $\hat{Y}$.We seek to define the loss function that balances between reconstructing depth images by minimizing the difference between the depth values while also penalizing distortions of high frequency details in the image domain of the depth map.

The loss function used in the proposed model is defined as.

$$L(Y, \hat{Y}) = L_{depth}(Y, \hat{Y}) + L_{grad}(Y, \hat{Y}) + L_{SSIM}(Y, \hat{Y})$$

The first term in loss function is a pointwise $L1$ loss.

$$L_{depth}(Y, \hat{Y}) = \frac{1}{n} \sum_{i}^{n} |Y_i - \hat{Y}_i|$$

The second term in the loss function is $L1$ loss over the image gradient.

$$L_{grad}(Y, \hat{Y}) = \frac{1}{n} \sum_{i}^{n} |G_x(Y_i - \hat{Y}_i)| + |G_y(Y_i - \hat{Y}_i)|$$

The third term in the loss function uses the structural similarity.

$$L_{SSIM}(Y, \hat{Y}) = \frac{1 - SSIM(Y, \hat{Y})}{2}$$

4.3 Dataset (NYU-Depth V2)

In this section we describe our experimental results and compare with other existing architectures. We evaluated our algorithm on the NYU-Depth v2 dataset in order to compare the robustness of our models. It is a dataset that provides images and depth maps for different indoor scenes captured at resolution of 640 X 480. The training set comprises 120K samples. We trained our model on 2000 subset filled in depth values using colorization scheme and tested on 694 training samples. Our model predicts the depth and produces the depthmap of output resolution of 492 X 369 pixels.

4.4 Implementation details

We implemented our proposed depth prediction architecture using PyTorch. Both training and evaluation are performed on a single high-end HP OMEN X 15-DG0018TX Gaming laptop with 9th Gen i7-9750H, 16 GB RAM, RTX 2080 8 GB Graphics and

Windows 10 operating system. Training part took around 10 hours and the test prediction was 0.23 seconds per image. We have used Adam optimizer with learning rate 0.0002 and parameter value set to beta1 0.9 and beta2 0.999 and batch size was set to 16.

4.5 Results

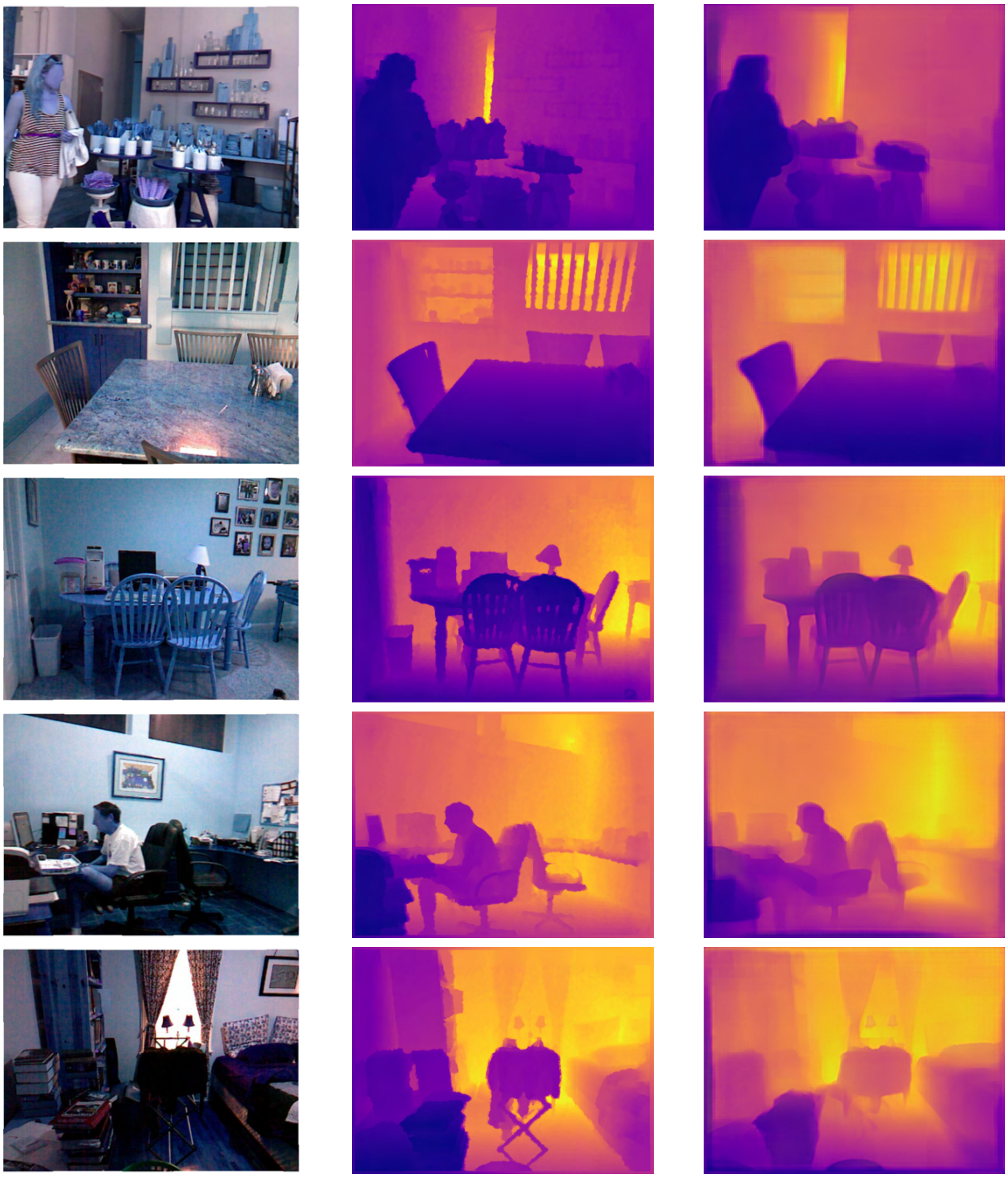


Figure 4.2: FC3D-ResUnet Results(1). (left) Input image (middle) ground truth (right) prediction

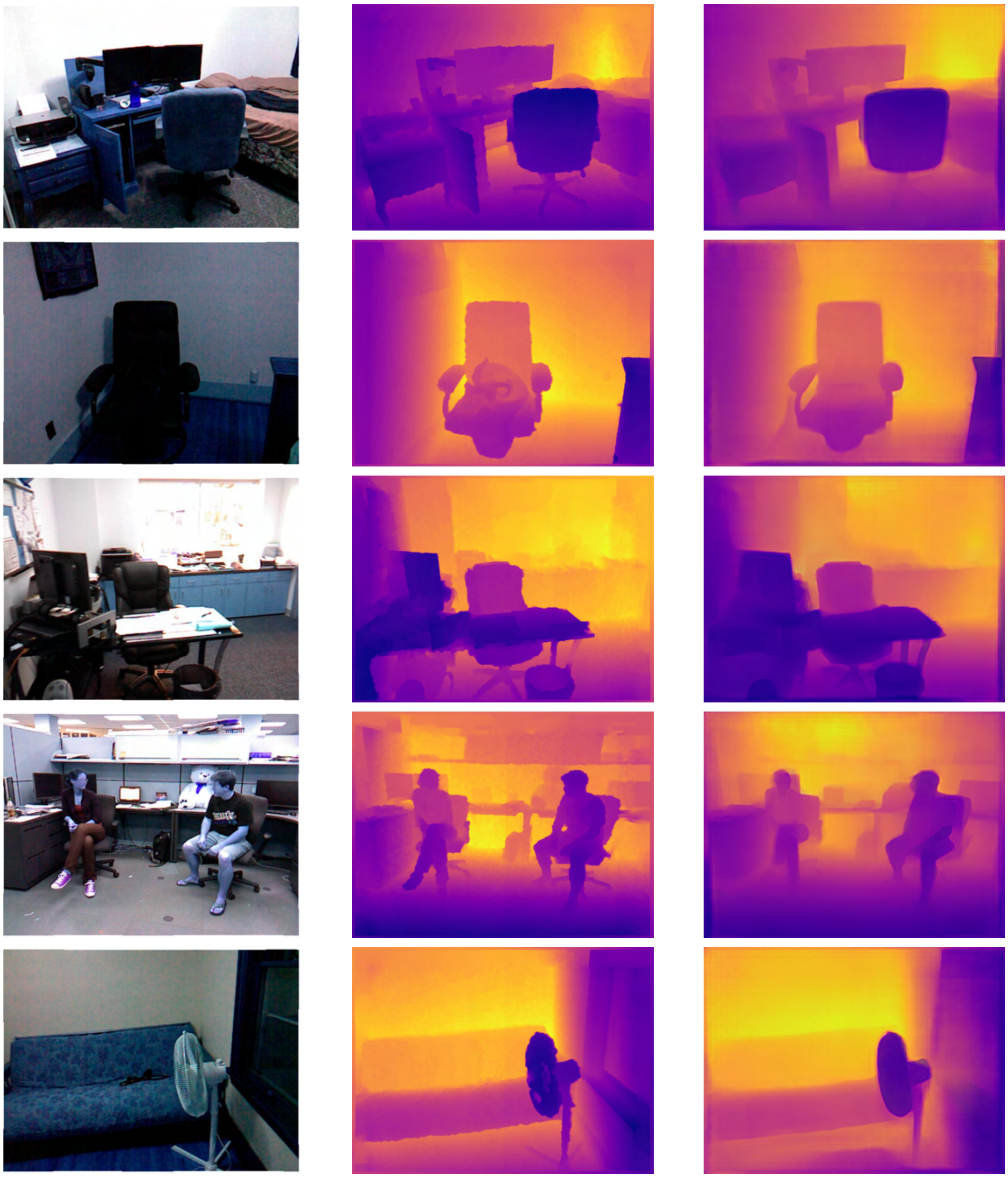


Figure 4.3: FC3D-ResUnet Results(2). (left) Input image (middle) ground truth (right) prediction

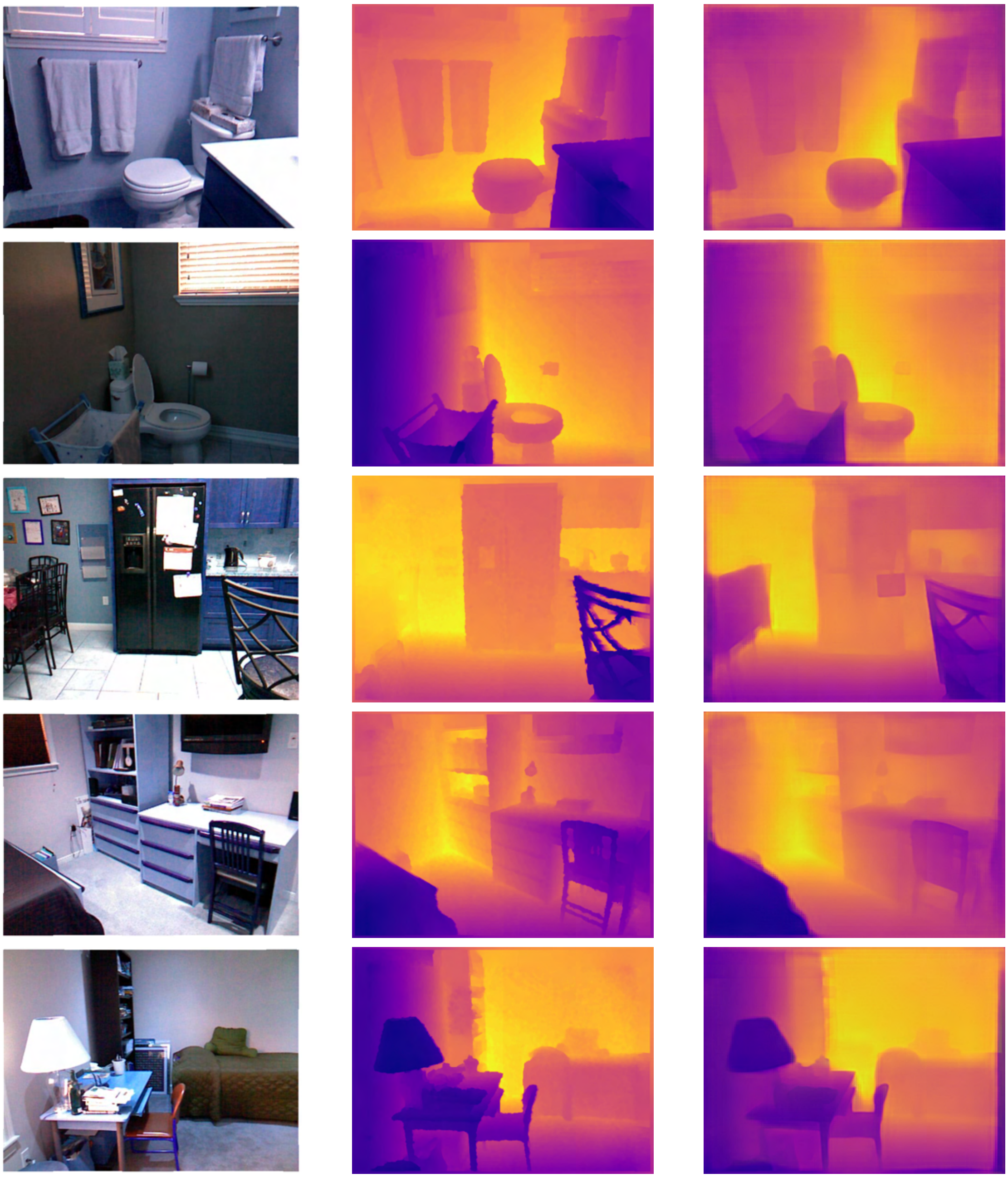


Figure 4.4: FC3D-ResUnet Results(3). (left) Input image (middle) ground truth (right) prediction

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Quantitative Analysis

We compare our model with SharpNet, Eigen, Laina. We evaluated these methods using most common error metric defined as:

- **Average relative error (rel):**

$$\frac{1}{n}\sum_{i}^{n}\frac{|Y_i - \hat{Y_i}|}{Y}$$

- **root mean squared error (rms)**

$$\sqrt{\frac{1}{n}\sum_{i}^{n}(Y_i - \hat{Y_i})^2}$$

- **Average (log_{10}) error:**

$$\frac{1}{n}\sum_{i}^{n}|log_{10}(Y_i) - log_{10}(\hat{Y_i})|$$

- **threshold accuracy (δ_i) :** Percentage of Y_i s.t $Max(\frac{Y_i}{\hat{Y_i}}, \frac{\hat{Y_i}}{Y_i}) = \delta < thr$ for $thr = 1.25, 1.25^2, 1.25^3$

where Y_i is a pixel in depth image Y, $\hat{Y_i}$ is a pixel in the predicted depth image $\hat{Y}$, and n is the total number of pixels for each depth image.

5.2 Qualitative Analysis

We performed qualitative analysis of different methods on Nyu-depth v2 dataset. The perception-based qualitative metric and depth edge reliability metric are computed. The mean structural similarity score (mSSIM) that measures the similarity of resulting depth maps in the image space. The mSSIM scores are computed considering gray scale visualization of the ground truth as a reference and estimated depth map as a predicted image. Mathematically, the mSSIM quality score is computed as:

$$\frac{1}{n}\sum_{i}^{n} SSIM|Y_i - \hat{Y}_i|$$

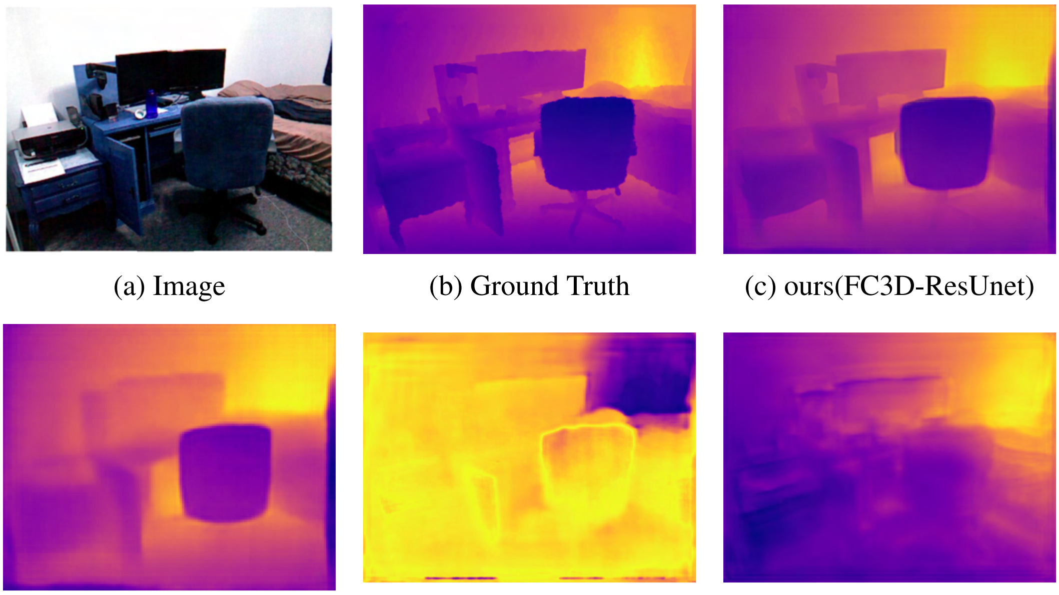
The second depth edge reliability metric (DERM), analyse edges in the predicted depth maps and ground truth. The gradient magnitude image of both the ground truth and the predicted depth image are determined for each sample.

Table 5.1: Comparison of different methods on NYUv2-Depth data.

	Eigen	SharpNet	Laina	Ours(3DBG-net)	Ours(FC3D-ResUnet)
rel ↓	0.207	0.139	0.127	0.185	0.123
log_{10} ↓	0.089	0.047	0.055	0.06	0.053
RMS ↓	0.634	0.495	0.573	0.594	0.465
σ_1↑	0.76	0.888	0.811	0.88	0.892
σ_2↑	0.82	0.979	0.953	0.946	0.980
σ_3↑	0.95	0.995	0.988	0.989	0.996
mSSIM ↑	0.7042	0.7186	0.955	0.949	0.980
DERM ↑	0.476	0.510	0.637	0.675	0.718

5.3 Conclusion

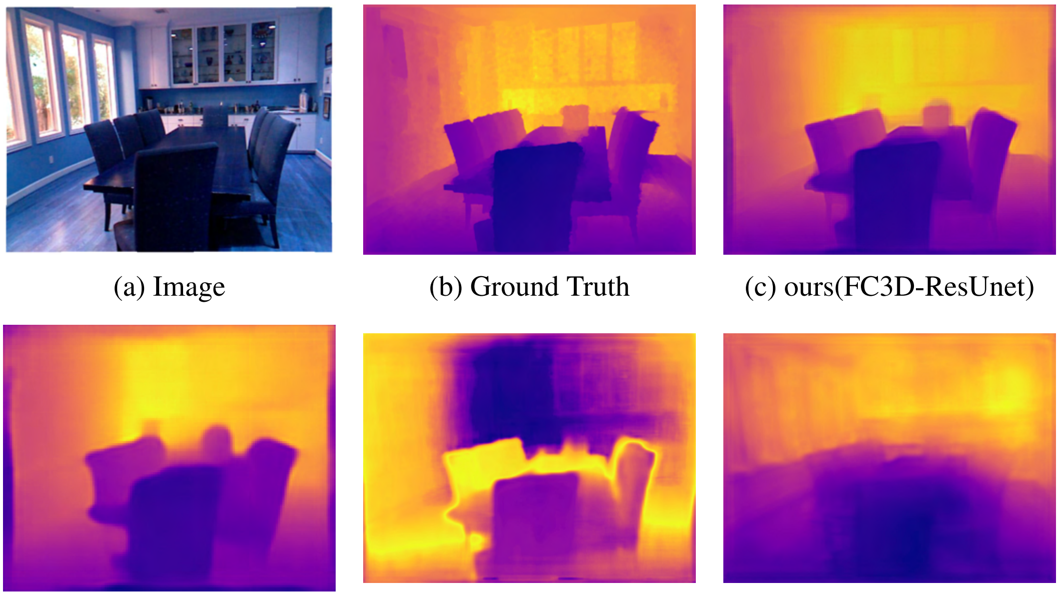
In this thesis we have proposed a deep neural network for depth estimation for single RGB images. Unlike other CNN architectures that require a multi-step process, our model consists of a single powerful step. The proposed architecture is fully convolutional that allows for training much deeper configuration, while reducing the number of parameters learned and number of training samples required. Evaluation on challenging benchmark NYU Depth v2 dataset demonstrates that our proposed model achieves state-of-the-art performance, both quantitatively and qualitatively. Our aim in this work is towards generating higher quality depth maps that capture the object boundaries which is indeed possible using existing architectures.



(a) Image (b) Ground Truth (c) ours(FC3D-ResUnet)

(d) Laina Network (e) SharpNet (f) Eigen Network

Figure 5.1: Depth Maps



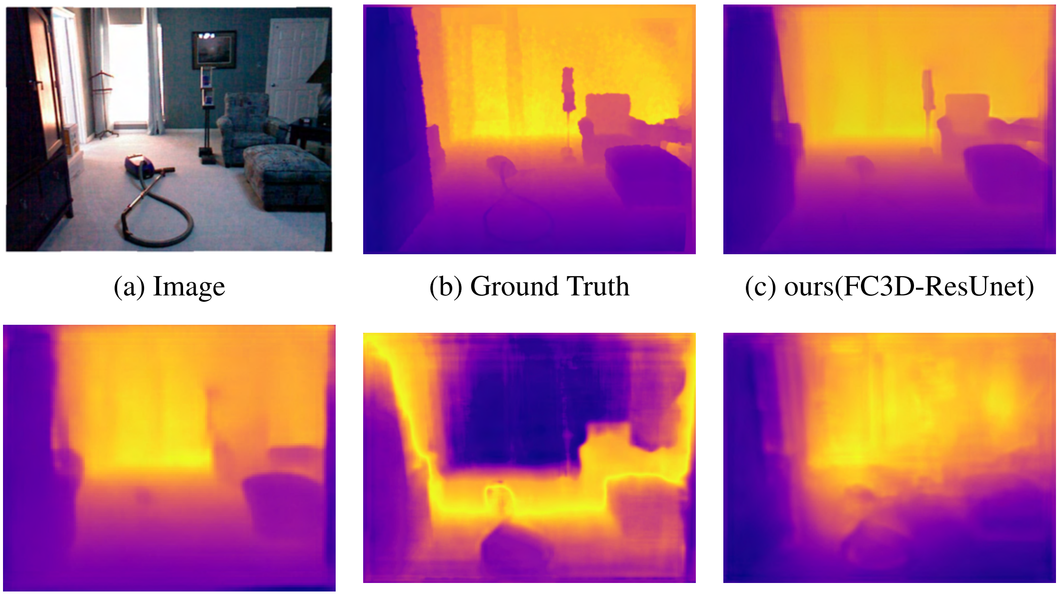
(a) Image (b) Ground Truth (c) ours(FC3D-ResUnet)

(d) Laina Network (e) SharpNet (f) Eigen Network

Figure 5.2: Depth Maps

5.4 Future Work

Our Architecture might not be able to produce good results on real world dataset because it is only trained on NYUv2-depth dataset. Next step should be to collect a lot of real world images with their respective depth maps. This step will be very challenging as training time will also increase. We also believe that the resolution of the depth map can be improved further as described in Aleksandr Safin (2021). We can extend our architecture from single image to sequence of images.



(a) Image

(b) Ground Truth

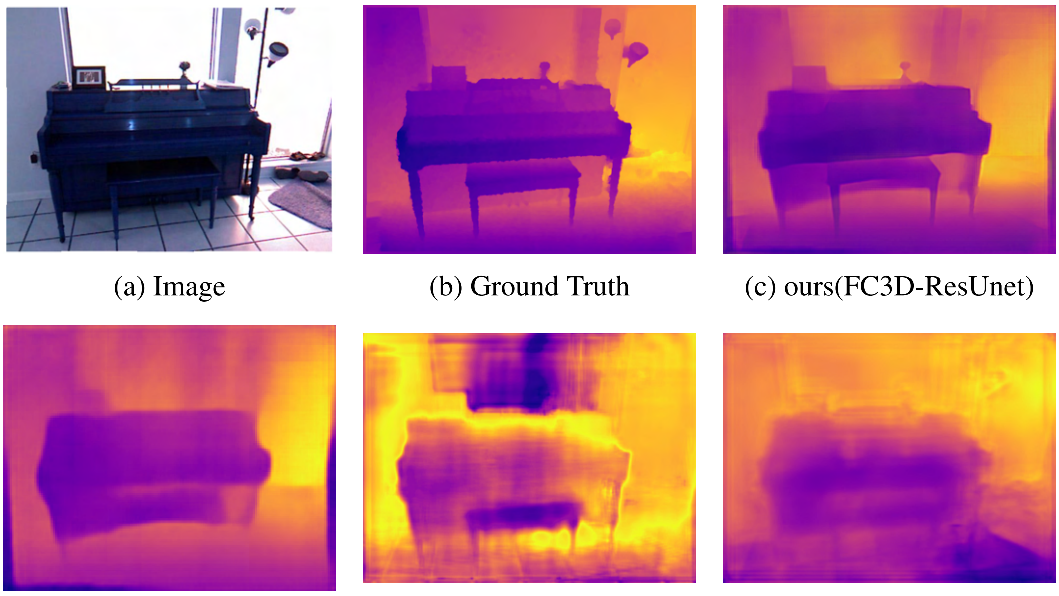
(c) ours(FC3D-ResUnet)

(d) Laina Network

(e) SharpNet

(f) Eigen Network

Figure 5.3: Depth Maps



(a) Image

(b) Ground Truth

(c) ours(FC3D-ResUnet)

(d) Laina Network

(e) SharpNet

(f) Eigen Network

Figure 5.4: Depth Maps

REFERENCES

1. **Aleksandr Safin, N. D. O. V. A. A. A. F. D. Z. E. B., Maxim Kan**, Towards unpaired depth enhancement and super-resolution in the wild. 2021.

2. **Chen, J.**, **S. Paris**, and **F. Durand** (2007). Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics (TOG)*, **26**(3), 103–es.

3. **Clément Godard, M. F. G. B., Oisin Mac Aodha**, Digging into self-supervised monocular depth estimation. ICCV 2019.

4. **Dan Xu, H. T. H. L. N. S. E. R., Wei Wang**, Structured attention guided convolutional neural fields for monocular depth estimation. CVPR 2018.

5. **David Eigen, R. F., Christian Puhrsch**, Depth map prediction from a single image using a multi-scale deep network.

6. **Fabio Tosi, M. P. S. M., Filippo Aleotti**, Learning monocular depth estimation infusing traditional stereo knowledge. CVPR 2019.

7. **Jamie Watson, G. J. B. D. T., Michael Firman**, Self-supervised monocular depth hints. ICCV 2019.

8. **Laina, I.**, **C. Rupprecht**, **V. Belagiannis**, **F. Tombari**, and **N. Navab**, Deeper depth prediction with fully convolutional residual networks. *In 3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016.

9. **Mansi Sharma, K. R. T. A. P., Abheesht Sharma**, A novel 3d-unet deep learning framework based on high-dimensional bilateral grid for edge consistent single image depth estimation. 2021.

10. **Nathan Silberman, P. K., Derek Hoiem** and **R. Fergus**, Indoor segmentation and support inference from rgbd images. *In ECCV*. 2012.

11. **Ramamonjisoa, M.** and **V. Lepetit** (2019). Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. *The IEEE International Conference on Computer Vision (ICCV) Workshops*.