Fall 2018

CSE 527: Introduction to Computer Vision
Depth Map Prediction from a Single Image using a Multi-Scale Deep Network

PROJECT REPORT

Supreeth Narasimhaswamy- 112007225
Priyanka Bedarkar-112046765

Abstract

Predicting depth is an essential component in understanding the geometric relations within a scene. Such relations provide richer representations of objects and their surroundings which could be helpful in other applications like recognition, 3D modeling, robotics, etc. We consider the problem of depth prediction from a single RGB image as proposed by Eigen et. al. [1]. This task required us to integrate both global and local knowledge from the image. For this, we used stacked architecture of two deep networks. One that makes a coarse prediction incorporating the global features of the entire image and the other to refine this predictions locally. We experimented with different heuristics to achieve performance at par to what is mentioned in the paper. We used NYU Depth Dataset.

Introduction

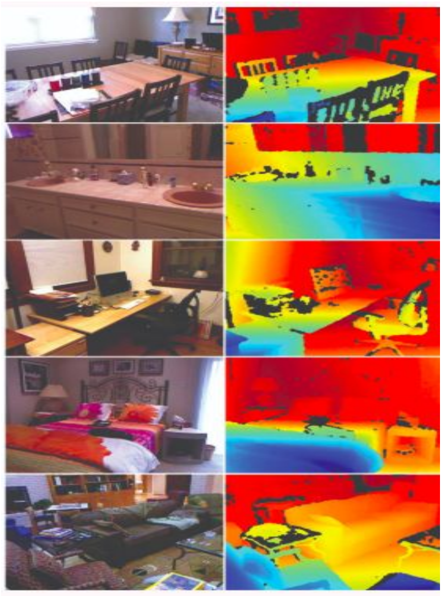
The problem of recovering depth information from images has been widely studied in computer vision. Traditional approaches operate by considering multiple observations of the scene of interest, e.g. derived from two or more cameras or corresponding to different lighting conditions. More recently, the research community has attempted to relax the multi-view assumption by addressing the task of monocular depth estimation as a supervised learning problem. Specifically, given training set of pairs of images and associated depth maps, depth prediction is casted as a pixel-level regression problem, i.e. a model is learned to directly predict the depth value corresponding to each pixel of an RGB image.
The task is inherently ambiguous mainly attributed to how we interpret the global scale of the image. To address this issue, we are using a scale-invariant error measure which allows us to focus on the spatial relations within the image rather than the general scale. The model we used for this task has two components : the first for global structure estimation and the second for refinement using local features. We trained the model using a loss that directly predict the depth value corresponding to each pixel of an RGB image. We compare the performance we achieved using NYU Depth Dataset.

Related Work

The problem of monocular depth estimation has attracted considerable attention in last decade. Mentioned below are some of the many approaches that estimate depth in images. Saxena et al. [2] predict depth from a set of image features using linear regression and a MRF, and later extend their work into the Make3D system for 3D model generation. However, the system relies on horizontal alignment of images, and suffers in less controlled settings. Ladicky et al. [3] show how to integrate semantic object labels with monocular depth features to improve performance; however, they rely on handcrafted features and use superpixels to segment the image. Karsch et al. [4] use a kNN transfer mechanism based on SIFT Flow to estimate depths of static backgrounds from single images, which they augment with motion information to better estimate moving foreground subjects in videos. Scharstein et al. [5] provide an evaluation of many methods for 2-frame stereo correspondence, organized by matching, aggregation and optimization techniques. Konda et al. [6], trained a factored autoencoder on image patches to predict depth from stereo sequences; however, this relies on the local displacements provided by stereo. Levin et al. [7] perform depth from defocus using a modified camera aperture, while the Kinect and Kinect v2 use active stereo and time-of-flight to capture depth.

Dataset

We train the model using NYU Depth V2 dataset by Silberman et al [8]. The NYU Depth dataset is composed of 464 indoor scenes, taken as video sequences using a Microsoft Kinect camera. This dataset contains 1449 RGB image-depth pairs of a variety of indoor scenes. Each image is 640x480. Eigen et. al. [1] trained the model using all images obtained by data augmentation with random online transformations. But due to limited computational resources, we trained the model on 1000 images. We used the remaining 449 images for testing the performance of our model. Eigen et. al. [1] also trained the model using the KITTI dataset which is composed of several outdoor scenes. But, again, due to time constraints and the limitation of computation power, we do not train on KITTI dataset.



Models

We implemented the baseline model as mentioned in the paper Eigen et. al. [1]. We experimented different heuristics to enhance the baseline model architecture in an attempt to get improved performance. Both, the Baseline model and the Experimented model are explained in details below.

1. Baseline Model

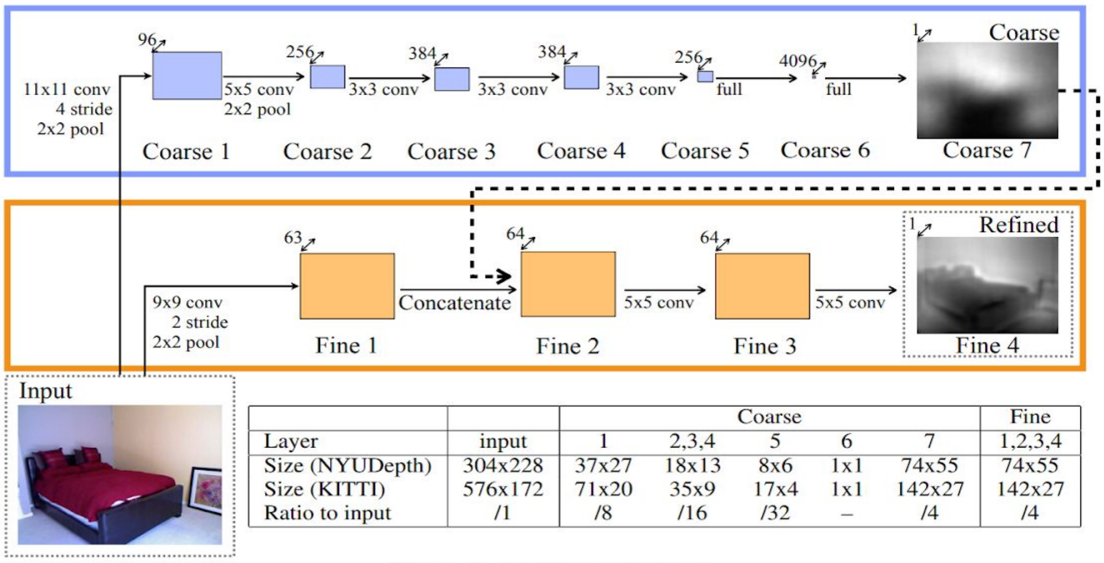
A coarse-scale network first predicts the depth of the scene at a global level. This prediction is then refined by a fine-scale network. Both, the coarse-net and fine-net, are applied to the original input, but in addition, fine-net receives the coarse-net's output as additional first-layer image features.

Global Coarse-Scale Network

This network predicts the overall depth map structure using a global view of the scene. The lower layers combine information from different regions of the image through max-pooling operations. Consequently, the network is able to integrate a global understanding of the full scene which helps the model to predict the depth. Such an understanding is essential to make effective use of cues such as vanishing points, object locations, and room alignment, in case of predicting depth from single images. The last fully connected layer of this network produces a 74x55 output which is passed as an image feature to fine-network's first layer.

Local Fine-Scale Network

This network edits the coarse predictions in order to align them with the local details like object boundaries and wall edges. Pooling is used only at the first layer (after convolutional layer) for edge features. The subsequent layers in this network are convolutional layers only. The last fully connected layer predicts the target depth. We train the coarse-net first against the ground-truth depth targets. Then we train the fine-net keeping the coarse-net's output fixed.



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Figure 1: Model architecture.

Implementation Details

- Epochs = 200
- Batch Size = 4
- Optimizer = SGD with momentum 0.9
- Training Data = 1000 (RGB image-depth pairs from NYU dataset)
- Test Data = 449 (RGB image-depth pairs from NYU dataset)
- Coarse-Net Training time = ~ 8 hours
- Fine-Net Training time = ~ 14 hours

2. Experimented Model

We implemented another model, inspired by the baseline model. The key differences between the baseline model and our experimentation model is as follows:

1. **Additional Layer** : We used an additional convolutional layer followed by a pooling layer after the third convolutional layer in the fine-net. A ConvNet captures low level features in first layer, a little better but still low level features in the next layers and at higher layers, object parts and simple structures are captured. Hence, we experimented to go deep, in an attempt to extract better depth maps from fine-net.
2. **Batch Normalization** : After each ReLU layer we added a batch-norm layer. Batch normalization reduces the amount by what the hidden unit values shift around (covariance shift). Normalizing the features is important. It also helped in speeding up the computation. Hence, we incorporated batch-norm after each ReLU Layer.
3. **Dropout** : We added additional dropout layers after every convolutional layers in the fine-net, in an attempt to reduce overfitting.
4. **Data Preprocessing** : We normalized the input images before passing them through any network.

Implementation Details

- Epochs = 150
- Batch Size = 4
- Optimizer = Adadelta
- Training Data = 1000 (RGB image-depth pairs from NYU dataset)

- Test Data = 449 (RGB image-depth pairs from NYU dataset)
- Coarse-Net Training time = ~ 5 hours
- Fine-Net Training time = ~ 12 hours

Error Measures

For a predicted depth map ***y*** and ground truth ***y****, each with ***n*** pixels indexed by ***i***, we evaluated the performance of our models using the following error measures :

Threshold: % of y_i s.t. $\max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$

Abs Relative difference: $\frac{1}{|T|}\sum_{y \in T} |y - y^*|/y^*$

Squared Relative difference: $\frac{1}{|T|}\sum_{y \in T} ||y - y^*||^2/y^*$

RMSE (linear): $\sqrt{\frac{1}{|T|}\sum_{y \in T} ||y_i - y_i^*||^2}$

RMSE (log): $\sqrt{\frac{1}{|T|}\sum_{y \in T} ||\log y_i - \log y_i^*||^2}$

RMSE (log, scale-invariant): The error Eqn. 1

$$D(y, y^*) = \frac{1}{2n}\sum_{i=1}^{n}(\log y_i - \log y_i^* + \alpha(y, y^*))^2$$

. . . Error Eqn. 1

Results

We evaluated the performance of our trained models on the 449 images of the NYU Depth dataset using various error measures.

Models	**Baseline Model (from paper)**		**Baseline Model (our implementation)**		**Experimentation Model**	
Error Metrics	**Coarse**	**Coarse + Fine**	**Coarse**	**Coarse + Fine**	**Coarse**	**Coarse + Fine**
Lower the better						
Abs relative difference	0.228	0.215	0.451	0.513	0.477	0.494
Sqr relative difference	0.223	0.212	0.495	0.469	0.474	0.485
RMSE (linear)	0.871	0.907	1.633	1.782	1.593	1.752
RMSE (log)	0.283	0.285	0.523	0.558	0.551	0.492
RMSE (log, scale inv.)	0.221	0.219	0.486	0.491	0.527	0.535
Higher the better						
threshold δ < 1.25	0.618	0.611	0.368	0.323	0.332	0.346
threshold δ < 1.25^2	0.891	0.877	0.461	0.454	0.444	0.442
threshold δ < 1.25^3	0.969	0.971	0.494	0.523	0.481	0.499

The results of the models we trained are not as good as the ones presented in Eigen et. al. [1]. This difference in performance could be attributed to the following reasons :

1. **Training Data** : We trained our models on less number of training examples. We used a small fraction of data what Eigen et. al. [1] used.

2. **Training Time** : Eigen et. al. [1] trained their model for around 3 days using a NVidia GTX Titan Black. We comparatively trained our models for a short amount of time.
3. **Fine-net** : We expected better results using the fine+coarse network rather than using only coarse-net. But as we can see from the above table the results obtained using fine+coarse network is slightly worse than the ones obtaining using only the coarse network. We believe this is due to the overfitting in fine-net since we are training using less number of examples.

Contribution by team members

Supreeth Narasimhaswamy:

- Implemented the "Experimented model" as described in the previous paragraph.
- Implemented error metrics such as log-scale invariant error, RMSE (linear), and RMSE (log) errors.
- Implemented data loader function and resizing functions.

Priyanka Bedarkar:

- Implemented the "Baseline model" as described in the previous paragraph
- Implemented error metrics such as Absolute relative difference and squared relative difference.
- Implemented functions to train the network.

Conclusion

Predicting depth estimates from a single image is a challenging task. Yet by combining information from both global and local views, it can be performed reasonably well. The system proposed by Eigen et al. [1] accomplishes this through the use of two deep networks, one that estimates the global depth structure, and another that refines it locally at finer resolution.

Acknowledgement

We would like to thank Professor Dimitris Samaras for his CSE 527 lectures. We would also like to thank Fan Wang and Sagnik Das for giving us the necessary guidance and suggesting possible ideas which we could explore for the successful completion of this project.

References

1. Eigen, David, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network." Advances in neural information processing systems. 2014.

2. J. Michels, A. Saxena, and A. Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In ICML, pages 593–600, 2005.

3. M. P. Lubor Ladicky, Jianbo Shi. Pulling things out of perspective. In CVPR, 2014.

4. K. Karsch, C. Liu, S. B. Kang, and N. England. Depth extraction from video using nonparametric sampling. In TPAMI, 2014

5. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV, 47:7–42, 2002.

6. K. Konda and R. Memisevic. Unsupervised learning of depth and motion. In arXiv:1312.3429v2, 2013.
7. A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In SIGGRAPH, 2007.
8. N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In ECCV, 2012.