Depth Prediction with Fully Convolutional VGG Network

Hailin Yu, 21721039, Computer Science

hailinyu@zju.edu.cn, 13819495414

Abstract

In this paper we address the fundamental problem of estimating depth from a single RGB image in computer vision area. Estimation depth from a single image is essentially an ill-posed problem.But state-of-the-art deep learning methods can get a relatively accurate depth map with respect to other techniques. We propose a fully convolutional network which uses convolutional layers of VGG-19 as features extracting layers followed by several deconvolutional and convolution layers to get higher resolution. In order to accelerate the convergence of the network, we put batch normalization to deconvolutional layers. The most important is that the network can get a relatively good result when trained by a small data set. Experiments on NYU Depth v2 dataset shows that our depth predictions relatively competitive with state-of-the-art methods.

Key Words

Fully convolution, Depth Estimation, Deep Learning, Neural Network

1 Introduction

Estimating depth is a basic problem that have been substantially researched in computer vision. Depth information in images is a key element to do further research in these areas, such as visual simultaneous localization and mapping, structure from motion and 3D reconstruction. Except mentioned above, there are other areas that can be leaded to improvements such as image recognition[1]. A large amount of methods have been proposed to address this problem. These methods can be classified into two categories, estimating depth from a single image and from more than one image.

There is much prior work on estimating from more than one image like based on stereo image and motion[14].But there has been relatively little on estimating from a single image compared with from more than one image. Though the depth can be accurately computed from two or more image with the geometric information among images, predicting depth from single image is also needed in practice. For example, these images that come from websites or social networks are commonly just a single images, but sometimes we also need depth information of these images for some particular reasons.

The reason why monocular case is not thoroughly researched as the stereo one is that the task is inherently ambiguous and ill-posed problem. According to the theory of camera producing image, an image can be get from infinite number of possible world scenes. Fortunately, a large amount of these world scenes is essentially not existing in the real world, which gives possibilities to calculate depth values from a single image with considerable accuracy.Especially with the spreading of deep learning technique in vision area, the accuracy of depth estimating from a single image has been largely improved.

In these paper, we proposed a new fully convolutional network for estimating depth from a single image. The network is consist of VGG-19[8] convolutional layers and up-sampling layers designed by myself and trained by the part of NYU Depth v2 labeled dataset. The remaining labeled dataset images are used as test data. Then we make comparison our method with others, and in order to discuss the effect of convolution in up-sampling layer, we make the experiments show the differences between the networks that equipped with convolution in up-sampling layers and that without convolution. The main contributions is as follows:

- Apply VGG-19 convolutional layers to depth estimation. VGG-19 has strong ability to extract abstract features from RGB images, which is beneficial to depth estimation.
- Design a new kind of up-sampling layer, which is consist of deconvolution and convolution.
- Modify the activation function of VGG-19 convolutional layers so that the network is more suitable to estimating depth rather than recognition.

The rest of this paper is organized as follows. In section 2, we simply introduce the related work in this area. In section 3, we thoroughly describe the network architecture, training process , parameter configurations and other techniques used in our works. The experiments and results are presented in section 4. Section 5 concludes the paper and describes the future work to improve the method proposed by this paper.

2 Related Work

Estimating depth from image data is initially researched in stereo vision, which find the correspondences first and then compute the corresponding depth value with triangulation[4,5]. This kind of methods must rely on motion or multiple cameras, and can accurately recover the depth with the condition that correspondences should be rightly found.

Classical methods on monocular depth estimation have mainly relied on hand-crafted features and used probabilistic graphical models to address the problem. One of the first works uses a MRF to infer depth from local and global features extracted from image by Saxena et al[6].

More recently, with the remarkable advances in the field of deep learning, a large amount of methods based on convolutional neural networks have been proposed to address this problem, and get large improvements. Eigen et al.[1] have been the first to use convolutional neural networks to regressing dense depth maps from a single image in two-scale architecture, where the first stage based on AlexNet produces a coarse output and the second stage refines the result produced by the first.Their work is later extended to additionally predict normals and labels with a deeper and more discriminative model based on VGG-16 and a three-scale architecture for further refinement[2].

Laina et al.[3] go a step further, and use residual networks to deepen networks and design a kind of up-sampling layer to replace fully connected layer, and eventually build the networks with fully convolutional layers, which has no limits on image size and need relatively less weights compared to using fully connect layers.

Jun Li et al.[7] proposed a fast-to-train mutil-streamed CNN architecture to predict image depth. Their methods can simultaneously predict depth and depth gradient. The two networks the designed all use VGG-16 as the feature extracting layers. In our method, we use VGG-19 which has more strong feature extracting ability than VGG-16.

Our networks architecture is also a fully convolutional networks without fully connected layers, which is consist of VGG-19 convolutional layers and up-sampling layer. In addition, we add batch normalization into up-sampling layers to accelerate the networks convergence and boost the ability of generalization.

3 Approach

This section mainly talk about the model architecture and the data augmentation,also the way to train this network is also described.

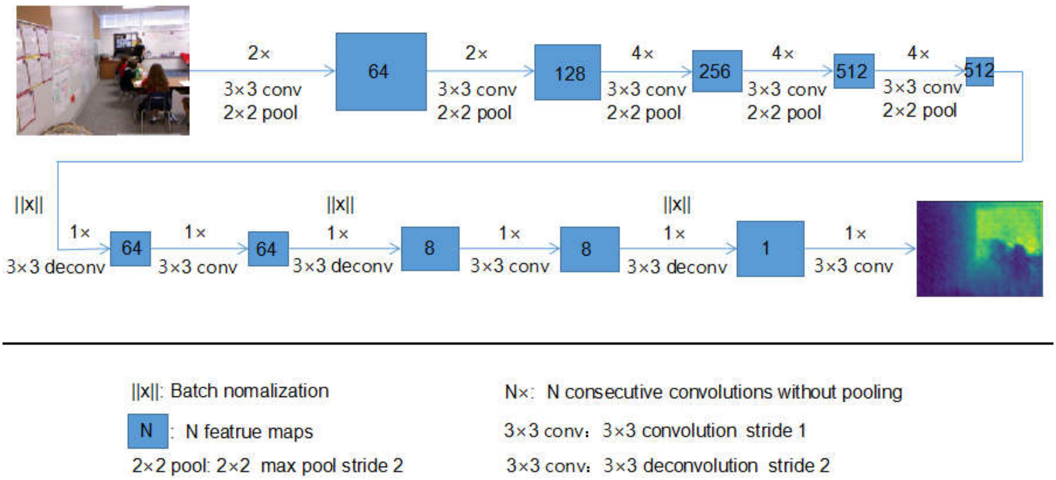

Figure 1. Network architecture.

3.1 Model Architecture

Our network is made of two parts, shown in Fig.1. The first part of the network is the convolutional layers of VGG-19.This part as down-sampling layers is consist of convolutional layers and pooling layers, and can be seen as extracting high-level features.

The second part of our architecture guides the network into learning its up-scaling through a sequence of deconvolutional and convolutional layers.

Because the architecture is fully convolutional, to some extent , we can feed it without limit on image size. But with limit on hardware,like memory size, we set an input of 160*120 pixels and predict an depth map with resolution of 80*60.

3.1.1 Down-sampling Part

Almost all current convolutional neural network architectures contain a contractive part that progressively decreases the input image resolution through a series of convolutions and pooling operations, giving higher-level neurons large receptive fields, thus capturing more global information. We extract the convolutional layers of VGG-19 and remove the last pooling layer. Here, as mentioned above, assume the input image size is 160*120 pixels, the receptive field of the last layer of this part is 228*228.

We have found that if we directly use VGG-19 convolutional layers, the final feature map is more than half filled with zero, and with the training going on, the number of zero is increasing because of dying neurons made by relu activation function. The result is not suitable for depth estimation, so we replace relu activation function by leaky relu, which produces a better result.

3.1.1 Up-sampling Part

In order to get a high resolution depth map, some form of up-sampling is required.The up-sampling layer in our network is consist of deconvolution and convolution. As shown in Fig.1, our network up-sampling part is made of three up-sampling layers, which is actually six layers with deconvolution and convolution.In theory, the up-sampling function can be achieved with only deconvolutional layers.The experiments shown that the result depth map is not consecutive, so we put deconvolution and convolution together to get a more smoothed depth map.

3.2 Loss Function

For training our network,we just choose L2 loss function as training loss,

$$\mathrm{L}(y^*, y) = \frac{1}{n}\sum_i d_i$$

where $d_i = y_i^* - y_i$ .There are many other loss function than can be used, but for the time reason, in this paper we just test this loss function.In the future, we will choose other loss function to train the network and then compare the result each other, such as scale-invariant loss function proposed by Eigen et al[1].Thought the loss is simple, the result is also good. So the network performance can be improved by other better loss function.

3.3 Data Augmentation

We augment the training data with random online transformations. The way of transformations such as color， flips and scale, is similar to Eigen et al[1].But it should be mentioned that rotation will produce blank space at the margin of images,and these blank space is filled with zero for both images and depths. The other methods can be seen in[1].

4 Experiments and Results

We train out model on the NYU Depth v2[15] labeled dataset. In this section, we will briefly introduce the commonly used data set in depth estimation area. We will show the results that our network produced on this dataset, and then thoroughly analyse our methods. The quantitative and qualitative results will be reported at the end of this section.

4.1 NYU Depth

The NYU Depth dataset[15] is composed of 464 indoor scenes, taken as video sequences using a Microsoft Kinect camera. We use the official train/test split,using 249 scenes for training and 215 for testing, and construct out training set using the labeled data for these scenes. RGB inputs are donwsampled by quarter, from 640*480 to 160*120.The labeled data includes 1449 images, 654 of which are left as test data and others are used for training.Every RGB image in the labeled data correspond to a depth map, which have been filled in depth values.

As configuration above, the training set has 800 unique images. We train our network using SGD with batches of size of 16. We initialize the down-sampling part by the pre-trained weights of VGG-19 and the up-sampling part by normal random value with 0 mean and 1 variance.All layers of the network have the same learning rate:0.001.Training took 8 hours total 1000 epochs using a NVidia GTX970.Test prediction takes 0.02 second per batch.

4.2 Results

We trains our network on NYU depth v2 labeled data with commonly used 795 images. For accurately evaluating our methods, we use official test data 654 images to compute the errors with several error metrics, such as RMSE(liner), RMSE(log) and squared relative difference and so on, which has clearly definitions in [1].

	A1(no conv)	A2(no aug)	A3(Final)	Eigen	Ladicky & al	
Threshold $\delta < 1.25$	0.569	0.609	0.629	0.611	0.542	higher is better
Threshold $\delta < 1.25^2$	0.849	0.862	0.880	0.887	0.829	
Threshold $\delta < 1.25^3$	0.944	0.951	0.961	0.971	0.940	
Abs relative difference	0.245	0.248	0.226	0.215	-	lower is better
Sqr relative difference	0.269	0.260	0.219	0.212	-	
RMSE(liner)	0.978	0.918	0.850	0.907	-	
RMSE(log)	1.036	0.315	0.299	0.285	-	
RMSE(log,scale invarient)	0.997	0.240	0.249	0.219	-	

Table 1. Comparison on the NYU Depth dataset. A1 represents the network that use 3 transpose convolution layer to regress the depth map with data augmentation. A2 represents the network that also use 3 transpose convolution layer, but we insert a convolution layer after each transpose convolution layers. We don't use data augmentation method in this network. A3, which is the final model we maintained, represents the model that added data augmentation processing upon A2.

As is shown in Table 1, the result produced by the network trained by the data augmentation way is better than no data augmentation. So data augmentation indeed can improve the ability of generalization of networks. Comparing A1 and A3, we can find that the network, which up-sampling layers is equipped with convolution, is better than that no convolution, and in experiments we noticed that the former go to convergence more quicker than the later.

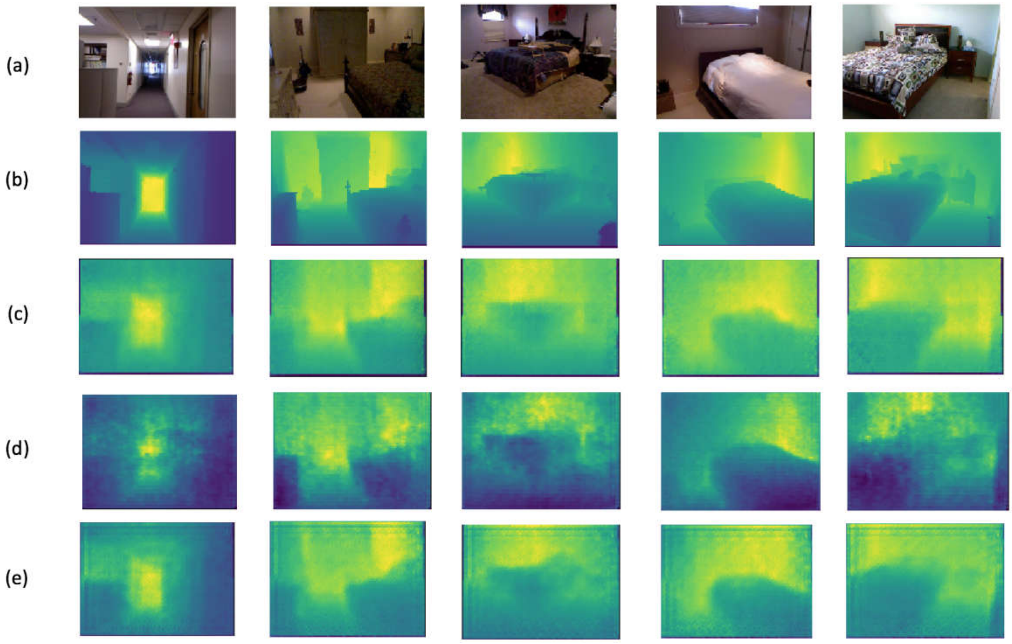

Figure 2. (a) input image; (b)ground truth; (c)A1 network(no_conv); (d)A2 network(no_aug); (e)A3 network;

As shown above, it is demonstrated that data augmentation can actually improve the accuracy but make the result not smooth. When adding convolution into up-sampling layers, the result get smooth effect.

It should be mentioned that when adding convolution into up-sampling layers, training spends less time than that no convolution. According to experiments, up-sampling layers equipped with convolution just need about 2000 iteration to convergence while no convolution need more than 20000 iteration to convergence.

5 Conclusion

Estimating depth from a single image is a challenging task. Deep convolutional neural network is indeed a effective tool to address this problem, especially with deeper layers. But with layers increasing, training gets hard. Fortunately, we can use the pre-trained model such as AlexNet[10], VGG[8] and ResNet[13], as feature extracting layers not only to facilitate convergence but also to improve the ability of generalization.Our network uses VGG-19 convolutional layers as feature extracting layers and is initialized by its pre-trained weights and then is trained by part of NYU Depth v2 labeled data, and finally get a relatively good result on official test data. Actually, the network will be improved a lot by training with a large dataset or go step further to tune its parameters. As is shown above, the result depth maps just have a blurry outline of objects and is lack of details.So we want to refine it through adjusting network architecture in the future.

In the future, we plan to go step further to adjust our network architecture.The network is composed of convolution, pooling and deconvolution, and convolution and pooling is used as down sampling to reduce the image resolutions, and in this process images lose a lot of details so that in the final stage depth maps is just an outline. In order to retain object details, we plan to additionally link every down-sampling layer output to corresponding up-sampling layer, which has the same input feature map size as down-sampling output.We believe that it can get a better result.

References

[1] Eigen, David, et al. “Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network.” Neural Information Processing Systems, 2014, pp. 2366–2374.

[2] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In ICCV, 2015.

[3] Laina, Iro, et al. “Deeper Depth Prediction with Fully Convolutional Residual Networks.” 2016 Fourth International Conference on 3D Vision (3DV), 2016, pp. 239–248.

[4] R. Memisevic and C. Conrad. Stereopsis via deep learning. In NIPS Workshop on Deep Learning, volume 1, 2011.

[5] F. H. Sinz, J. Q. Candela, G. H. Bakır, C. E. Rasmussen, and M. O. Franz. Learning depth from stereo. In Pattern Recognition, pages 245–252. Springer, 2004

[6] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In Advances in Neural Information Processing Systems, pages 1161–1168, 2005.

[7] Li, Jun, et al. “A Two-Streamed Network for Estimating Fine-Scaled Depth Maps from Single RGB Images.” 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3392–3400.

[8] Simonyan, Karen, and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” International Conference on Learning Representations, 2015.

[9] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[10] Krizhevsky, Alex, et al. “ImageNet Classification with Deep Convolutional Neural Networks.” Advances in Neural Information Processing Systems 25, 2012, pp. 1097–1105.

[11] Ioffe, Sergey, and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” International Conference on Machine Learning, 2015, pp. 448–456.

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

[13] He, Kaiming, et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[14] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1253–1260. IEEE, 2010.

[15] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(5):978–994, 2011.

Appendix(Task Arrangement)

Neural Network Architecture Design and Implement: Hailin Yu
Data Processing and Augementation: Zhaoyang Huang
Experiments and Conclusion: Hailin Yu and Zhaoyang Huang