Images Classification with Estimated Depth Map *

Yihui He[†]
Xi’an Jiaotong University
heyihui@stu.xjtu.edu.cn

Metehan Ozten
University of California, Santa Barbara
m_ozten@umail.ucsb.edu

Abstract

We consider the problem of doing image classification using estimated depth information. This problem clearly falls into the domain of transfer learning, since we are using a model trained on a set of depth images in order to generate depth maps (additional features) for use in another classification problem using another disjoint set of images. It is a challenging task as no direct depth information is provided. Previous related research efforts have been focused on image classification tasks using RGB images and depth image estimation but none have attempted to use depth image estimations in order to aid image classification over RGB images. Therefore, in this paper we present a way of transferring domain knowledge on depth estimation to a seperate image classification task over a disjoint set of training,validation and test data. To our knowledge, we are the first to bridge gap between image classification and depth estimation.

Specifically, we attempt to implement the recent work by Fayao Liu et al. *[1], build a RGBD dataset and do image classification on the RGBD dataset we built and then compare the performance of both a simple feedforward neural network and a multi-layer convolutional neural network of the RGBD dataset compared to the RGB dataset Our project code, models, and example results are available on github: github.com/yihui-he/Depth-estimation-with-neural-network github.com/netzo92/cs291k-FP*

1. Introduction

Estimating depths from a single monocular image depicting general scenes is a fundamental problem in computer vision, which has widespread applications in scene-understanding, 3D modeling, robotics, and other challenging problems. It is a notorious example of an ill-posed problem, as one captured image may correspond to numerous real world scenes[2]. it remains a challenging task for computer vision algorithms as no reliable cues can be exploited, such as temporal information, stereo correspondences, *etc*. Previous research involving depth-maps usually involv geometric[3]–[5] approach or CNN[1] approach, , and Semantic labeling[6] with depth information. Nevertheless, all these works didn’t try to do recognition with depth map.

Different from previous efforts, we propose to put depth map into practice on classification task. While extensively studied in semantic labeling and accuracy improvement, depth map regression has been less explored for classification problems.

Recently, the efficacy and power of the deep convolutional neural network (CNN) has been harnessed. With a CNN, we are able to perform depth estimation on a single image[1]. However, most classification tasks still perform on RGB images. With only RGB images, CNN features have been setting new records for a wide variety of vision applications[7]. Despite all the successes in depth estimation and image classification, the deep CNN has been not yet been used for learning on RGBD images, since RGBD datasets are not as widely-used as RGB datasets . To our knowledge, we are the first to bridge gap between depth estimation and image classification.

To sum up, we highlight the main contributions of this work as follows:

- We implemented a deep convoluntional neural field on depth estimation problem and obtained similiar results
- We created the first RGBD image dataset for CIFAR10.
- We define a new metric for ill-posed depth prediction problem.
- We prove that depth channel has a better feature representation than R,G,B channels, and show that training on RGBD images can somehow improve accuracy.

[*] This work is our course project for 2016 Spring CS291K Advanced Data Mining, taught by prof. Xifeng Yan in UCSB.

[†] Yihui is a CS 2nd year undergrad, from Xi’an Jiaotong University, China. Interested in Deep Learning and Computer Vision. He is an exchange student at UCSB during 2016 Spring.

2. Related Work

Convolutional networks have been applied with great success for object classification and detection. ConvNets have recently been applied to a variety of other tasks, like depth estimation. Depth estimation from single image is well addressed by Liu*et al.* [1] and Eigen*et al.* [8]. They both agree that depth estimation is an ill posed problem, since there's no real groud truth depth map. By contrast, we define transfer learning accuracy metric for depth estimation model. It becomes easier to compare performance of different depth estimation model.

Depth map has been successfully applied to some problems. based on depth information, performance improvement on semantic labeling[8] has been seen. however, depth map hasn't been combined with classification task. to our knowledge, we are the first to bridge gap between depth estimation and image classification.

our work builds upon state-of-the-art depth estimation model[1] which is a two loss neural network. we build RGBD dataset, and investigate the quality the depth maps more deeply. moreover, we improve accuracy of image classification task with depth map.

3. Deep Convolutional Neural Field

We present the details of deep convolutional neural field model we used for depth estimation in this section.

3.1. Theory and Architecture

The goal here is to infer the depth of each pixel in a single image depicting general scenes. we make the common assumption that an image is composed of small homogeneous regions (superpixels). Let x be an image and $y = [y_1, ..., y_n]^T \in R^n$ be a vector of continuous depth values corresponding to all n superpixels in x. We model the conditional probability as softmax:

$$Pr(y|x) = \frac{exp(-E(y,x))}{\sum_i exp(E(y_i,x))} \qquad (1)$$

where E is energy function. To predict the depths of a new image, we solve the maximum a posteriori (MAP) inference problem:

$$y^\star = \arg\max_y Pr(y|x). \qquad (2)$$

We formulate the energy function as a typical combination of unary potentials U and pairwise potentials V over the nodes (superpixels) N and edges S of the image x:

$$E(y,x) = \sum_{p \in \mathcal{N}} U(y_p, x) + \sum_{(p,q) \in \mathcal{S}} V(y_p, y_q, x). \qquad (3)$$

The unary term U aims to regress the depth value from a single superpixel. The pairwise term V encourages neighboring superpixels with similar appearances to take similar depths. We aim to jointly learn U and V in a unified CNN framework. In Figure.1 , we show a sketch of our deep convolutional neural field model for depth estimation. As we can see, the whole network is composed of a unary part, a pairwise part and a CRF loss layer.

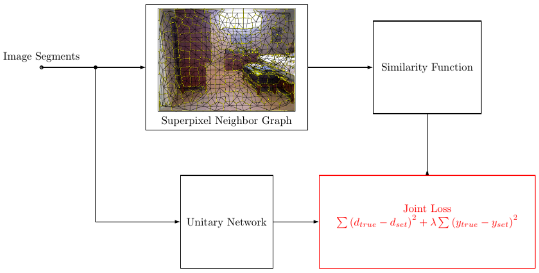
Image Segments
Superpixel Neighbor Graph
Similarity Function
Unitary Network
Joint Loss
$\sum (d_{true} - d_{set})^2 + \lambda \sum (y_{true} - y_{set})^2$

Figure 1. Deep Convolutional Neural Field model

For an input image, which has been segmented into n superpixels, we consider image patches centered around each superpixel centroid. The unary part then takes all the image patches as input and feed each of them to a CNN and output an n-dimensional vector containing regressed depth values of the n superpixels. The network for the unary part is composed of 5 convolutional and 4 fully-connected layers with details in Figure.2.

Kindly note that the CNN parameters are shared across all the superpixels. The pairwise part takes similarity vectors (each with K components) of all neighboring superpixel pairs as input and feed each of them to a fully-connected layer (parameters are shared among different pairs), then output a vector containing all the 1-dimensional similarities for each of the neighboring superpixel pair. The CRF loss layer takes as input the outputs from the unary and the pairwise parts to minimize the negative log-likelihood.

3.1.1 Unary part

The unary potential is constructed from the output of a CNN by considering the least square loss:

$$U(y_p, x; \theta) = (y_p - \hat{y}_p(\theta))^2, \ \ \forall p = 1, ..., n. \qquad (4)$$

Here $\hat{y}_p$ is the regressed depth of the superpixel p parametrized by the CNN parameters θ. The network architecture for the unary part is depicted in Figure.2. It is composed of 5 convolutional layers and 4 fully connected layers. The input image is first segmented into superpixels, then for each superpixel, we consider the image patch centered around its centroid. Each of the image patches is resized to 224224 pixels and then fed to the convolutional neural network. Note that the convolutional and the fully-connected layers are shared across all the image patches of different superpixels.

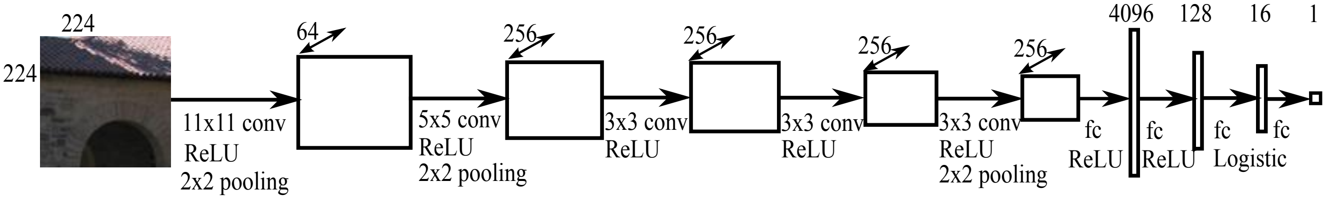

Figure 2. unary part of Deep Convolutional Neural Field model

3.1.2 Pairwise part

We construct the pairwise potential from 3 types of similarity observations: color difference, color histogram difference and texture disparity[9]. Each of them enforces smoothness by exploiting consistency information of neighbouring superpixels:

$$V(y_p, y_q, x; \beta) = \sum_{k=1}^{K} \beta_k S_{pq}^{(k)} (y_p - y_q)^2, \ \forall p, q = 1, ..., n. \tag{5}$$

Here, $K = 3$ in our case. $S_{pq}$is similarity of two neighbor superpixels p and q. β is trainable parameters, so we can let CNN decide which similarity is more important.

3.2. Implementation Details

We implement the network training on Make3D[10] dataset with Tensorflow[11]. Make3D dataset contains more outdoor scenes, which makes it easier for us to transfer learning onto the CIFAR10 dataset. During each SGD iteration, around 700 superpixel image patches are processed. Our implementation differs from the original implementation[1], Since we have enough memory, we feed 700 superpixel image patches into memory at once. Other parts of implementation are similar.

During implementation, we initialize the first 6 layers of the unary part in Figure.2 using a CNN model trained on the ImageNet from[12]. First, we do not back propagate through the previous 6 layers by keeping them fixed and train the rest of the network with momentum 0.9, learning rate 0.0001, and weight decay 0.0005. Then we train the whole network with the same momentum and weight decay.

3.3. Experiment

We measure our performance on Make3D dataset and compare our result with Liu *et al.* [1] as a sanity check.

The Make3D dataset contains 534 images depicting outdoor scenes. As pointed out in [13], this dataset is with limitations: the maximum value of depths is 81m with far objects are all mapped to the one distance of 81 me- ters. As a remedy, two criteria are used to report the prediction error (C1) Errors are calculated only in the regions with the ground-truth depth less than 70 meters; (C2) Errors are calculated over the entire image. We follow this protocol. Performance is shown in Table1. You can see that our model achieve pretty close result, which allows us do further research on depth map.

Method	Error(C1) (lower is better)			Error(C2) (lower is better)		
	rel	log10	rms	rel	log10	rms
Our implementation	0.335	0.137	9.49	0.338	0.134	14.60
Original paper	**0.314**	**0.119**	**8.60**	**0.307**	**0.125**	**12.89**

Table 1. Sanity check (**Bold** is better)

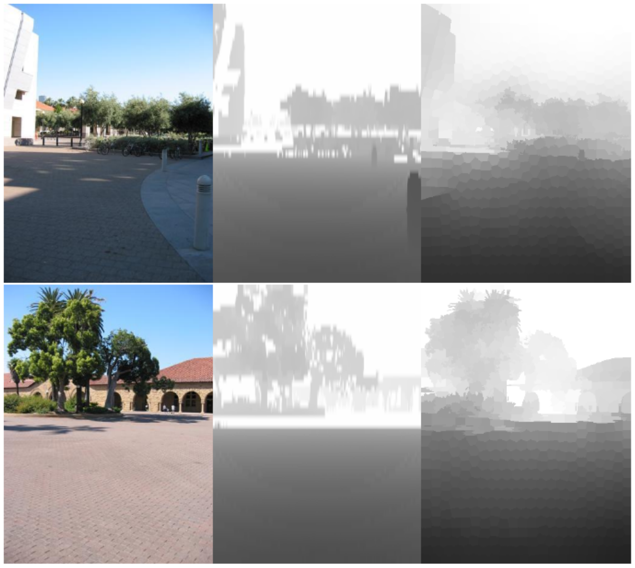

Figure 3. original image, ground truth, depth estimation(from left to right)

In Figure3 we also show depth maps our model learned.

4. RGBD Image for Classification

Recent depth image research works mainly focus on depth-estimation[1] and segementation with depth image[8]. And we've witnessed significant improvement on depth estimation accuracy in these years. However, most image classification tasks nowadays are still performed on RGB images. So we want to transfer depth knowledge learned by depth estimation model into our image-classification model. In this section, we first built a RGBD imageset for CIFAR10[14], based on a trained deep convolutional neural field model in the previous section. To investigate the effect of the depth channel on image classi-

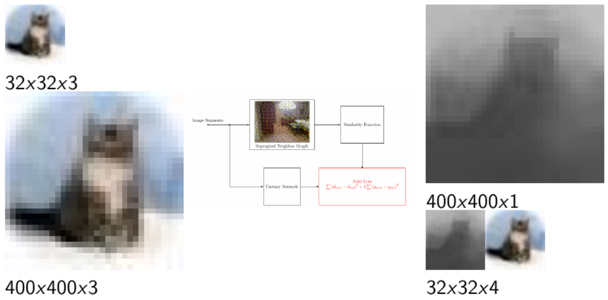

Figure 4. Transer Learning: Build RGBD CIFAR10 dataset

fication task, we design two experiments (one with a simple feedforward NN and one with a CNN) Finally, we propose a new metric for depth estimation performance measurement.

4.1. Build RGBD CIFAR10 Dataset

Since the Deep Convolutional Neural Field model accepts images that are much larger that CIFAR10 tiny images(32x32), we build RGBD dataset as follow:

1. resize CIFAR10 tiny image(32x32x3) to normal size(400x400x3) in order to feed in CNF.
2. perform depth estimation on normal size image.
3. downscale the output image(depth image, 400x400x1) back to tiny image(32x32x1).
4. combine RGB and D channels together as our RGBD image(32x32x4).

Figure4 shows the transfer learning procedure. Since there is no groundtruth depth image for CIFAR10 dataset, we can't directly measure the accuracy of our depth estimation attempts for these tiny images. However, we can infer this indirectly in two ways. First we can look at these depth images and make sure that most of them is reasonable. Figure5 shows some depth maps. Second, we can use the accuracy results of our two experiments as a new metric to quantify depth map quality.

4.2. Classification Task on RGBD CIFAR10

In order to make it easier to show effect of depth channel, we employ a simple two layer neural network for classification task. The architecture for learning on the RGBD dataset is shown in Figure6. The number of neurons in the input layer depends on input. If input is a single channel(R,G,B,D), we have 32x32 neurons. The amount of hidden neurons is not determined. We perform fine tuning for each situation. The number of output neurons is always number of classes(10 classes for CIFAR10). Technical details of our architecture is shown in Table2. Hyperparameters not dicussed here will be fine tuned.

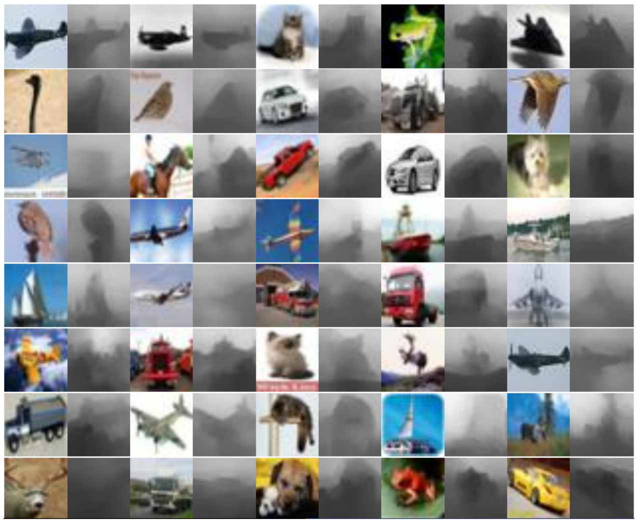

Figure 5. Depth map estimated by deep convoluntional neural field

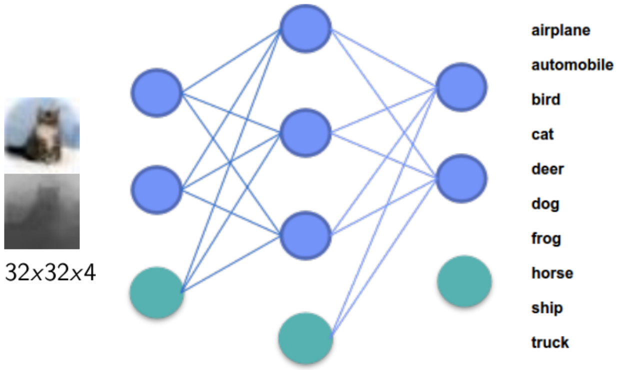

Figure 6. Learning on RGBD dataset

regularization	Activation	Update	batch
Dropout	ReLU	Momentum	128

Table 2. Archtecture details for classification task on our new RGBD dataset

4.3. Experiment

We measure depth map quality in two ways. First, we train neural network on R, G, B, D channel as input respectively. And compare their loss and accuracy. Second, we train neural network on RGB, RGBD respectively. And compare their loss and accuracy.

4.3.1 R vs G vs B vs D

We perform fine tuning on each channel. So that their performances are approximately optimal. Figure7 shows training accuracy comparision through time. Figure8 shows validation accuracy comparision through time.

You can see that, at testing time, depth channel out per-

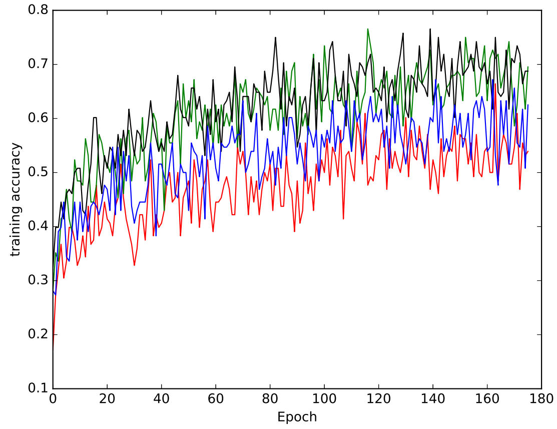

Figure 7. R vs G vs B vs D, training time

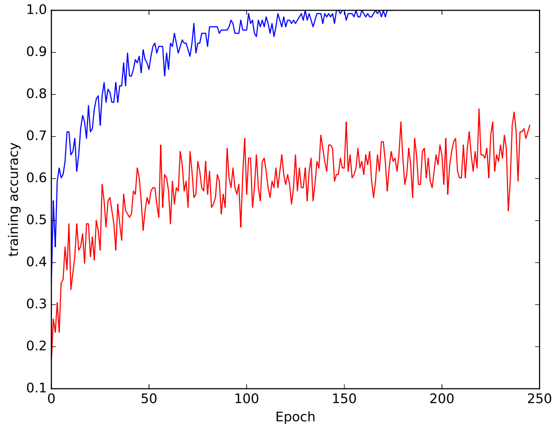

Figure 9. RGB vs RGBD, training time(RGBD:blue, RGB:red)

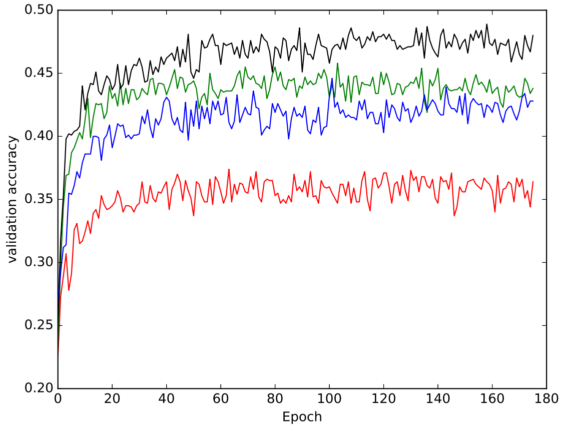

Figure 8. R vs G vs B vs D, testing time

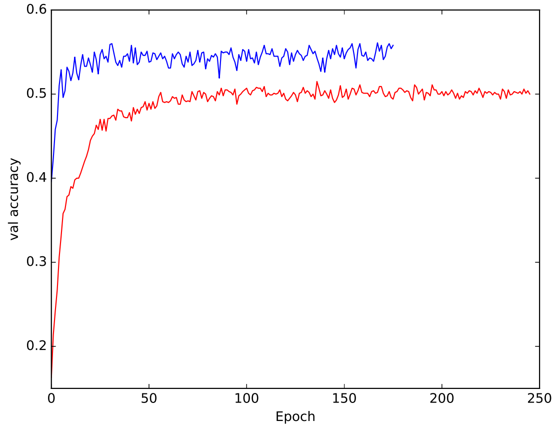

Figure 10. RGB vs RGBD, testing time(RGBD:blue, RGB:red)

form R,G,B channels under the same architecture. It implies that, depth channel has a better feature representation than R, G, B channels.

4.3.2 RGB vs RGBD

We perform fine tuning for both RGB and RGBD situations. So that their performances are approximately optimal. Figure9 shows training accuracy comparision through time. Figure13 shows validation accuracy comparision through time.

We get **56%** and **52%** validation accuracy with RGBD and RGB dataset respectively. This can be seen as a sign that depth map brings extra knowledge learned by deep convolutional neural field to our classification task.

You can also notice that, although RGBD dataset have more inputs and neurons, it has a much higher converge rate than RGB dataset. It can be interpreted as a better feature representation brought by depth map.

4.3.3 CNN Experimentation

Our previous experiments using a 2-layer feedforward neural network yielded a performance increase of 4% when comparing the accuracy of a NN trained on the RGB dataset to the NN trained on the RGBD dataset. These results did not satisfy us as a simple feedforward neural network is not the optimal tool used for image classification and so we decided to test the performance of our CNN [15] over the CIFAR-10 RGB dataset and compare it to the performance achieved over our CIFAR-10 RGBD dataset. We achieved an accuracy of **57.5%** and **53%** using the RGBD and RGB datasets respectively. This performance gain of **4.5%** in accuracy between RGBD and RGB using the CNN is slightly better than the original accuracy gain **4.0%** using the 2-layer ANN.

Our convolutional neural network was composed of two convolutional layers (with subsequent max-pooling layers) and

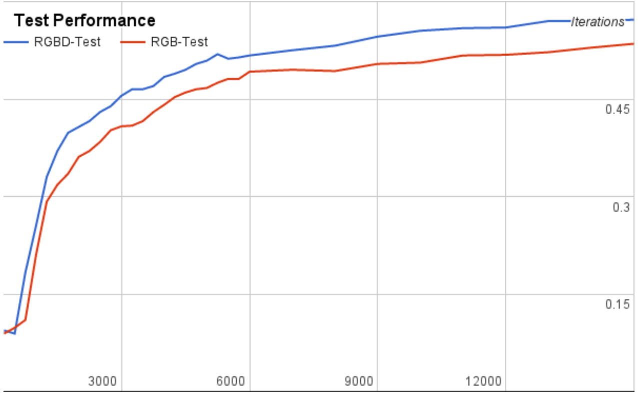

Figure 11. RGB vs RGBD Test CNN Accuracy, Iterations

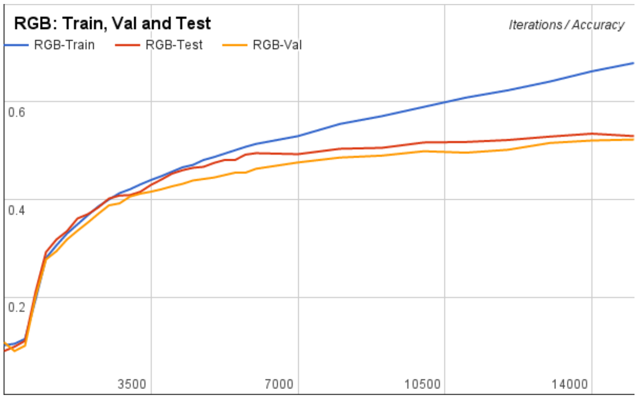

Figure 12. RGB CNN Train/Val/Test Accuracy, Iterations

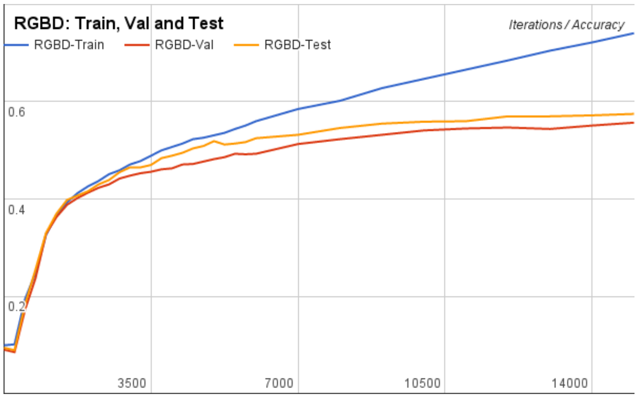

Figure 13. RGBD CNN Train/Val/Test Accuracy, Iterations

5. Further Work

5.1. Depth Estimation

We've mentioned that depth estimation is an ill-posed problem, since we can not find the ground truth depth map which would allow us compare the performance of different depth estimation models. However, using the accuracy metric we proposed, performance can be measured indirectly. The only drawback is that our running on our metric need much more time than most other typical metrics. If we had more time, we would measure the performance of existing depth estimation models.

5.2. Learning on RGBD Dataset

In our experiment, we didn't implement state-of-the-art image classification model[16] for simplicity. We consider to test RGBD dataset on that model. Maybe we can witness accuracy surpassing current record.

We also plan to build more RGBD datasets and publish them for research usage.

5.3. Learning on RGBD(ground-truth) vs RGBD(estimated)

Compare the accuracy of Learning on RGBD(ground-truth) vs RGBD(estimated).

6. Conclusion

We successfully reimplemented the state-of-the-art depth-estimation model using We create the first RGBD image dataset for CIFAR10, and investigate its quality using our metric. We define a transfer learning accuracy metric for depth prediction problem. On RGBD CIFAR, we prove that depth channel has a better feature representation. We also show that training on RGBD images can somehow improve image classification accuracy.

Role Clarification

We divide our teamwork as follow. **Yihui**'s contribution: 1. Proposed idea. 2. Implemented deep convolutional neural field 3. Create RGBD CIFAR10 dataset 4. Perform experiment for 2-layer neural network on RGBD dataset. 5. prepare presentation 6. edit final report. **Metehan**'s contribution: 1. Discussed idea. 2. implement and perform experiment using AlexNet (CNN) on RGBD dataset. 3. prepare presentation 4. edit final report.

References

[1] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5162–5170.

[2] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[3] V. Hedau, D. Hoiem, and D. Forsyth, "Thinking inside the box: Using appearance models and context based on room geometry," in *Computer Vision–ECCV 2010*, Springer, 2010, pp. 224–237.

[4] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei, “Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces,” in *Advances in neural information processing systems*, 2010, pp. 1288–1296.

[5] A. Gupta, A. A. Efros, and M. Hebert, “Blocks world revisited: Image understanding using qualitative geometry and mechanics,” in *Computer Vision–ECCV 2010*, Springer, 2010, pp. 482–496.

[6] L. Ladicky, J. Shi, and M. Pollefeys, “Pulling things out of perspective,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 89–96.

[7] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: An astounding baseline for recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[8] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multiscale convolutional architecture,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[9] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on kullback discrimination of distributions,” in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision &amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, IEEE, vol. 1, 1994, pp. 582–585.

[10] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in *Advances in Neural Information Processing Systems*, 2005, pp. 1161–1168.

[11] Martn Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vigas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *Tensorflow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: `http://tensorflow.org/`.

[12] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *ArXiv preprint arXiv:1405.3531*, 2014.

[13] M. Liu, M. Salzmann, and X. He, “Discrete-continuous depth estimation from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 716–723.

[14] A. Krizhevsky and G. Hinton, *Learning multiple layers of features from tiny images*, 2009.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *ArXiv preprint arXiv:1512.03385*, 2015.