Depth Estimation from a Single Image Using a Deep Neural Network

Milestone Report

Rawan Alghofaili

February 2015

1 Introduction

As previously mentioned in the project proposal, I will be using a convolutional neural network to estimate depth from a single image. It will be explained later in this report how the Places [5] pretrained convolutional network will be utilized to solve this problem. Further architectural implementation details will be explained as well.

2 Method

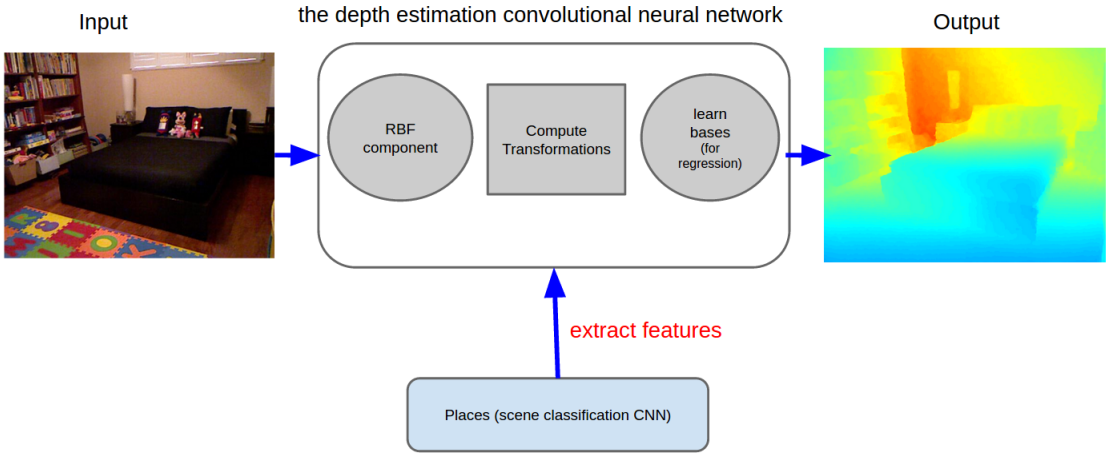

Figure 1: The architecture of the convolutional network

This convolutional neural network architecture is inspired by [1]'s approach of using RBF kernels in estimating depth. The *RBF component* in Figure 1 refers to $\phi_j(x) =$

$exp(-||f(x) - f(c_j)||^2/2\sigma^2)$ defined in [1]. Both the transformation and bases will be translated into fully connected convolutional neural net layers.

As Figure 1 shows, features from the Places CNN[5] will be extracted and used as centers.

3 Implementation

3.1 Layer set up

Caffe [2] was used as a skeleton for implementation. The *RBF component* ($\phi_j(x) = exp(-||f(x) - f(c_j)||^2/2\sigma^2)$) from [1] was implemented as a caffe "Blob" or layer, where the center c_j is, as previously mentioned, the fixed feature vector extracted from the Places CNN.

3.2 Feedforward

The feedforward operation was relatively trivial to implement once the caffe skeleton code was understood. By overriding the *ForwardCPU()* method in the Blob with my implementation of $\phi_j(x)$ here referred to as the function RBF():

for *each center* c_i **do**
| output[i] = RBF(c_i, feature vector)
end
where $i \in \{1, 2, ..., n\}$ and n is the number of centers(feature vectors extracted from places)

3.3 Backpropogation

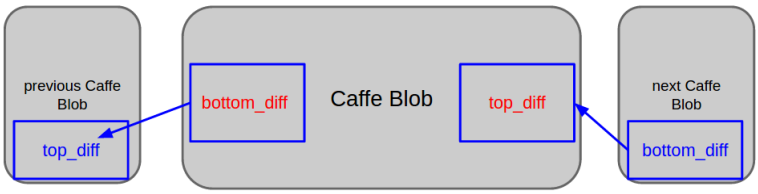

Figure 2: Gradient storage in Caffe

In caffe, the gradient for each Blob is computed and stored in *bottomdiff*. The contents of *bottomdiff* are automatically copied to *topdiff* of the previous later. The gradient from the next layer (*topdiff* in the current Blob) is multiplied by the output of RBF() and added to *bottomdiff* of the current Blob:

for *each center* $c_i \in$ *previous Blob* **do**
 | bottomdiff += RBF(c_i, feature vector)*topdiff
end

4 Testing

Unit tests were written to test the layer setup, feedforward and backpropagation. Layer setup was tested by creating a Blob and verifying its dimensionality. Feedforward was tested by injecting an input vector of ones and ten center feature vectors of ones as well. In order for feedforward to produce the correct results each center should output a value of one. This is because the l2-norm of $x = [1, 1, ..., 1]$ and center $c_i = [1, 1, ..., 1]$ should equal to 0. Because $e^0 = 1$ each center should output one in the feedforward operation.

As for the backpropogation unit test, caffe provides a GradientChecker utility which I've used to my advantage. Using finite differencing it estimates a gradient and compares it to the gradient computed by backpropogation. As you can see in Figure 3, the layer passed all three tests as well as tests conducted by caffe.

```
[----------] 3 tests from ThetaLayerTest/1, where TypeParam = caffe::DoubleCPU
[ RUN      ] ThetaLayerTest/1.TestForward
[       OK ] ThetaLayerTest/1.TestForward (0 ms)
[ RUN      ] ThetaLayerTest/1.TestSetUp
[       OK ] ThetaLayerTest/1.TestSetUp (0 ms)
[ RUN      ] ThetaLayerTest/1.TestGradient
[       OK ] ThetaLayerTest/1.TestGradient (3 ms)
[----------] 3 tests from ThetaLayerTest/1 (3 ms total)
```

Figure 3: RBF component *ThetaLayer* passing unit tests

5 Future work and goals

The next step would be to set up the convolutional neural net and start finetuning Places in the hope of improving the performance beyond the performance achieved in [1]. Introducing Multitask Learning by bifurcating the network to learn the depth and scene information simultaneously is also a good step.

References

[1] Baig, M., Torresani, L.: *Coarse-to-fine Depth Estimation from a Single Image via Coupled Regression and Dictionary Learning.* arXiv preprint arXiv:1310.1531, 2013.

[2] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: *Decaf: A deep convolutional activation feature for generic visual recognition.*

[3] Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: *A comparison and evaluation of multi-view stereo reconstruction algorithms.* In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2006).

[4] Silberman, N., Derek Hoiem, Fergus, R.: *Indoor segmentation and support inference from rgbd images.* In ECCV, 2012.

[5] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: *Learning Deep Features for Scene Recognition using Places Database.* Advances in Neural Information Processing Systems 27 (NIPS), 2014.