StereoPhonic: Depth from Stereo on Phones

Robert Konrad, Nitish Padmanaban

CS 231n Final Project

Introduction

Widespread adoption of augmented reality will make it necessary to extract depth from the environment on a portable device with limited computational resources.

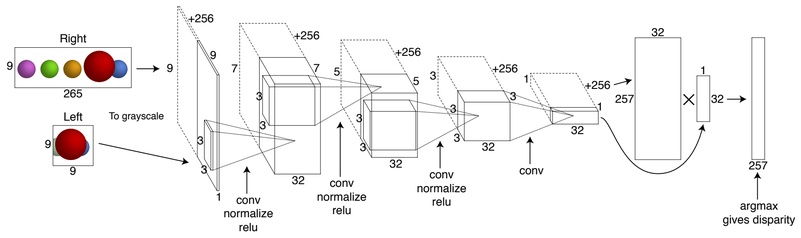
A recent trend in mobile phones is the inclusion of multiple cameras, making stereo images an attractive option for calculating depth.

Estimating depth from stereo images is an ideally suited problem for neural networks: humans seem capable of robust judgments of relative depth from stereo images, but doing so computationally remains on open problem.

However, running a network on a phone imposes constraints on memory, speed, and therefore the resulting size of the network, making complex models difficult to deploy.

We implement an efficient network based on Luo et al. with the ARM Compute Library for GPU-enabled computation on a mobile device.

Model



We use a Siamese architecture. Both the left and right images are fed into the same network, then we take the inner product of the output of the final convolutional layer of the left and right images. The inner product is computed for all candidate disparity shifts, with the argmax used as the disparity.

Phone-specific Features

Batch normalization was folded into the convolutional layer to reduce the compute footprint. This is computed from the original convolutional and batch normalization weights and moving averages as

$$x' = k * x + b$$

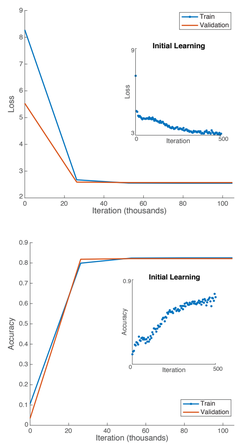
$$\left(\frac{x' - \mu}{\sqrt{\sigma^2 + \varepsilon}}\right)\gamma + \beta = \left(\frac{k * x + b - \mu}{\sqrt{\sigma^2 + \varepsilon}}\right)\gamma + \beta$$

$$= \left(\frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}}k\right) * x + \left(\frac{(b - \mu)\gamma}{\sqrt{\sigma^2 + \varepsilon}} + \beta\right)$$

$$= k' * x + b'$$

with the equivalent kernel k' and bias b' used on the phone.

Training



We train using the KITTI Stereo 2015 Dataset. When training, we find that the network trains mostly in under an epoch with a higher learning rate, and no other parameters significantly affect the result. No regularization is needed due to lack of overfit in the validation error.

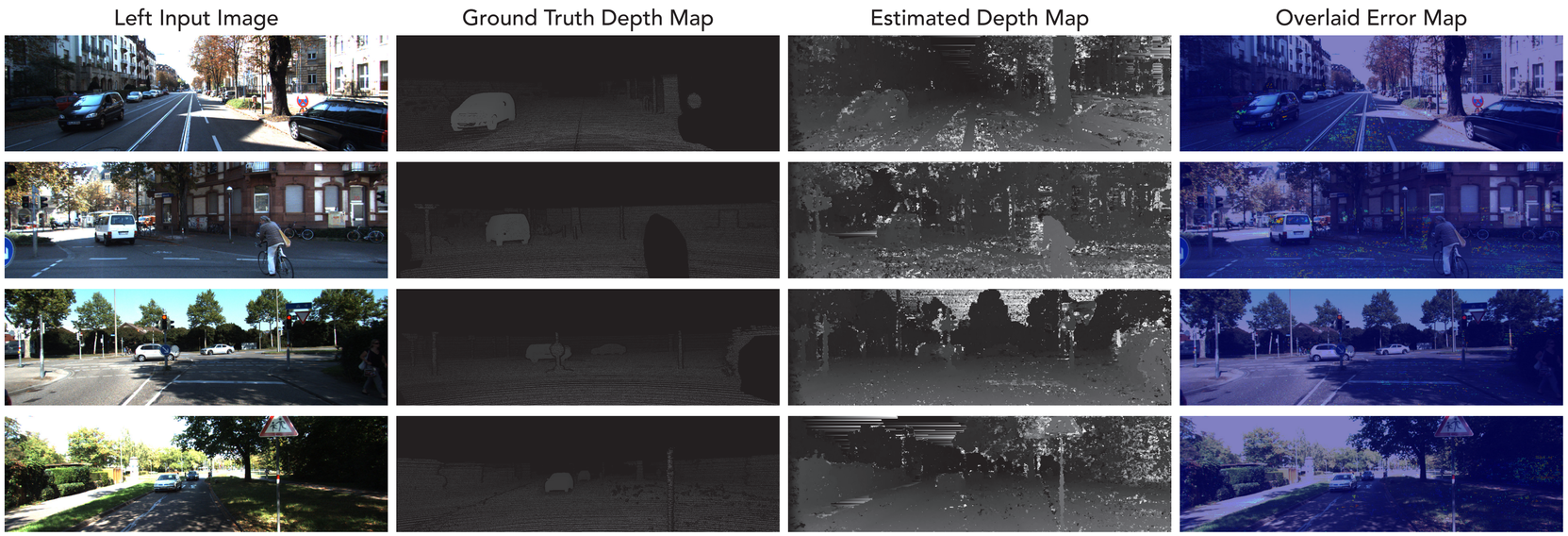
We observe better accuracy with more complex models without incurring any overfit, as seen in the table to the right. However, we choose to use the less complex model for efficiency.

Architecture		Hyperparameters			Training		Validation	
Layers	*Filters/layer*	*Learning Rate*	*Optimizer*	*Batch size*	*Loss*	*% > 3px*	*Loss*	*% > 3px*
9	32	0.1	Adam	512	5.5491	100	n/a	n/a
4	32	0.01	Adagrad	512	3.4358	34.23	3.3875	32.6
4	32	0.01	RMSProp	512	3.2485	30.39	3.2319	29.43
4	32	0.001	Adam	512	3.2577	30.77	3.2208	29.17
4	32	0.01	Adam	128	3.2449	30.17	3.226	28.85
4	32	0.01	Adam	1024	3.1834	29.25	3.1968	28.78
4	32	0.01	Adam	256	3.197	29.43	3.1969	28.69
4	32	0.01	Adam	512	3.2002	29.5	3.1833	28.39
4	64	0.001	Adam	512	3.0807	27.9	3.0658	26.59
9	32	0.01	Adam	512	2.6154	19.58	2.7334	19.82
9	64	0.01	Adagrad	128	2.2317	11.59	2.1731	9.63

We also recreate the results of the original paper, to verify our network on TensorFlow. The 9-layer, 64-filter network in the last row of the table gives a validation error of 9.63%, which is within range of Luo et al.'s result of 8.95%.

Results

We achieve a final test set error of 19.69% (i.e. an accuracy of 80.31%). This corresponds to an average offset from the correct disparity of about 7.5 pixels. Example inputs from the validation set, our estimate, and the error are below.



Performance on Mobile Platform

As reported by Luo et al., the performance on a desktop machine tasks less than a second to process an image. However, as expected, the phone takes a significantly longer time, taking over a minute.

Considering the recent developments in this space, such as Caffe2Go, Core ML, and TensorFlow Lite, we expect that the process of putting large networks on a mobile will become easier. New models could also benefit from a focus on efficiency over the increasingly marginal gains in accuracy provided by deeper, more complicated networks.