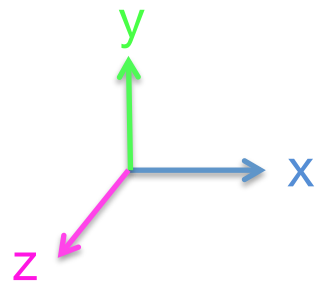
DESIGNING DEEP NETWORKS FOR SURFACE NORMAL ESTIMATION

Xiaolong Wang, David F. Fouhey, Abhinav Gupta

Presented by Yu-Cheng Lin

The Problem

Given a single image (RGBD), estimates the surface normal at each pixel.



Input Image Surface Normal (Output)

Input Image Surface Normal (Output)

Dataset NYU Depth v2

Left:

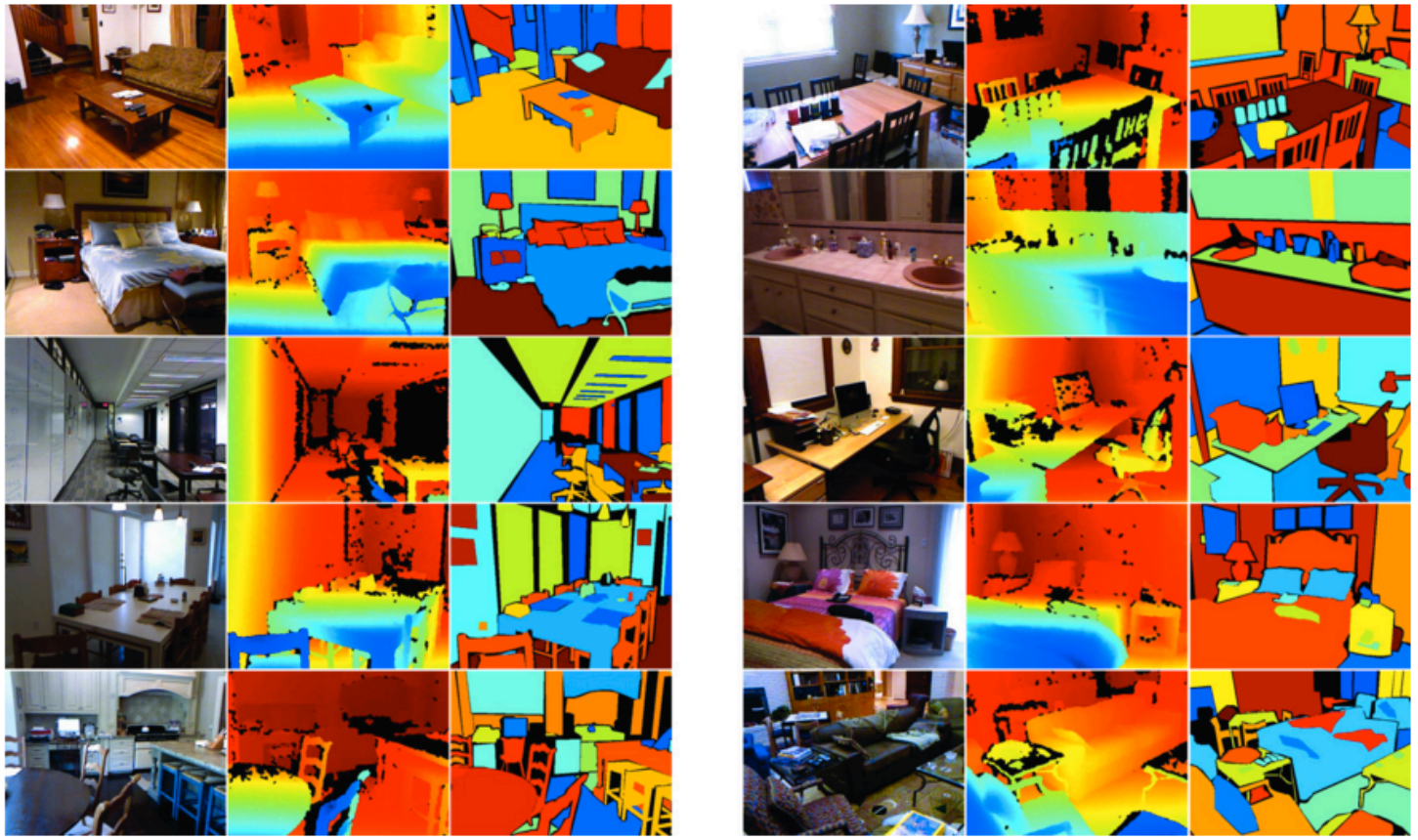
Samples of the RGB image

Center:

The raw depth image

Right:

The class labels from the dataset



http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Dataset NYU Depth v2

Left:

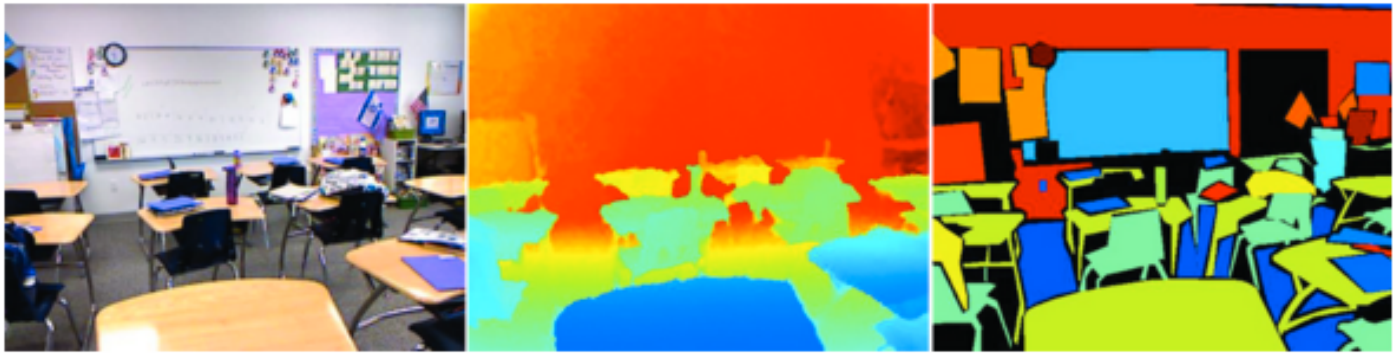
Output from the RGB camera

Center:

Preprocessed depth

Right:

A set of labels for the image



http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Heart of the Problem: Two Questions

1. What are the right primitives for understanding?

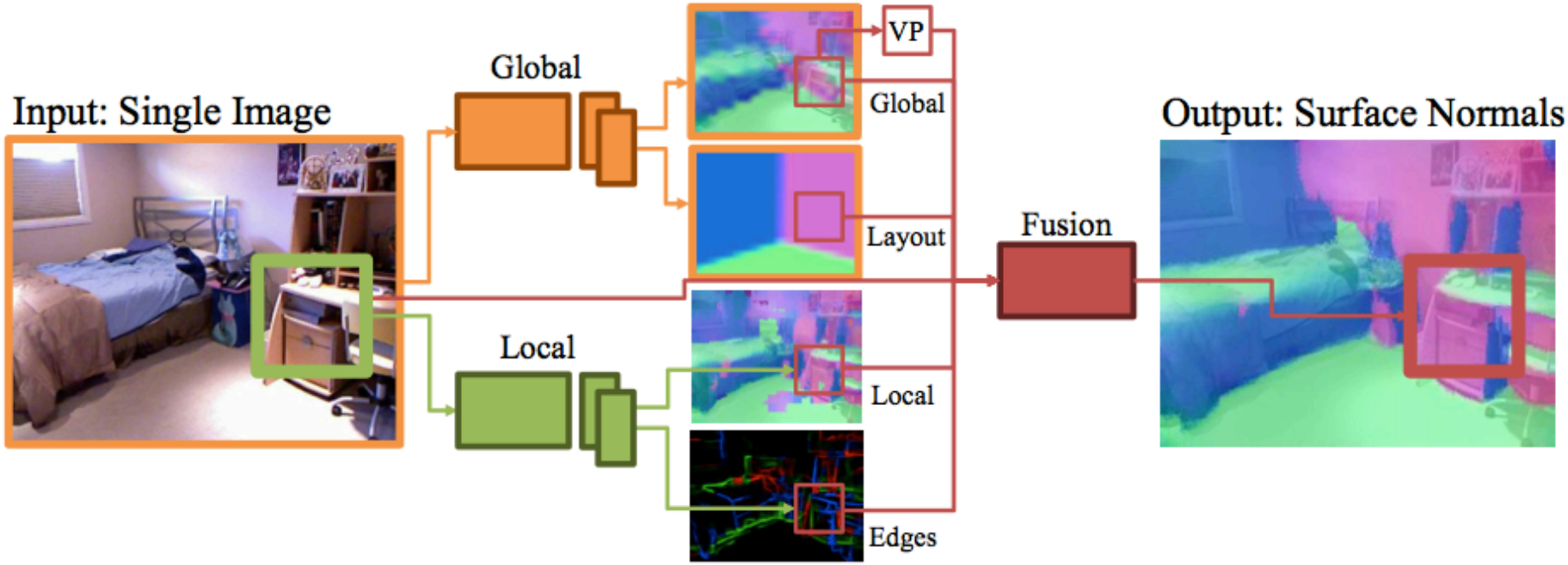
Use the data to derive a representation right from the pixels.

2. Given the local evidence, how can one obtain a global 3D scene understanding?

Fusing global and local evidence with several constraints (man-made, Manhattan world) and meaningful intermediate representations (room layout, edge labels)

Overview

1. Separately learn global and local processes
2. Use a fusion network to fuse the contradictory beliefs into a final interpretation



Making Regression as Classification

Surface Normal:

1. Use k-means to learn a codebook
2. Delaunay triangulation cover is constructed over the words
3. Normals rewritten as weighted codeword combinations of the triangles

Room Layout:

Use k-medoids clustering over 6000 room layouts

Edge Label: convex, concave, occluding, no-edge

Global Network

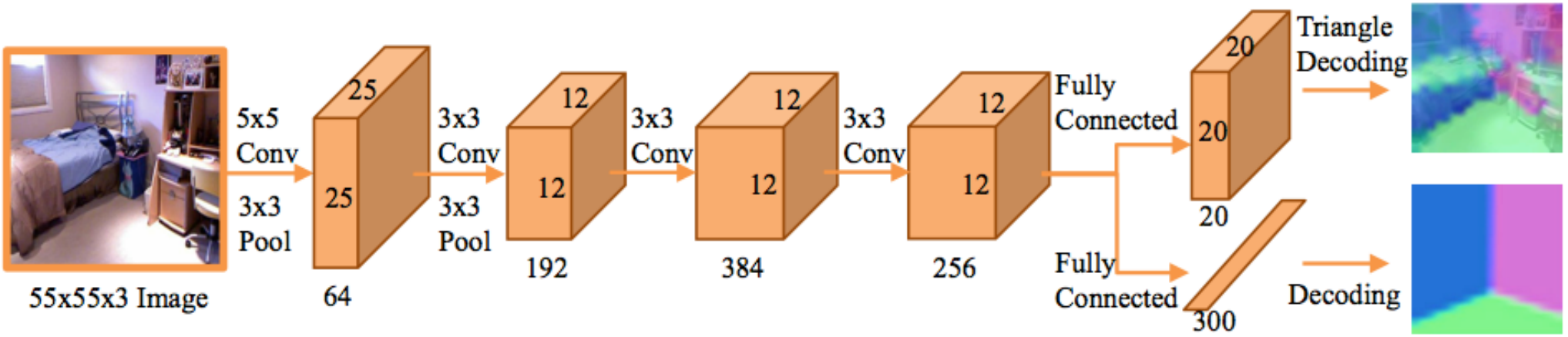
Goal: capture the coarse structure, clarifying ambiguous image portions

Input: Whole image rescaled to 55 x 55 x 3

Output: (Mt = 20, Kt = 20)

surface normal estimation: Mt x Mt x Kt, Kt is #(classes in codebook)

room layout: simple classification over 300 categories



UCDAVIS

Local Network

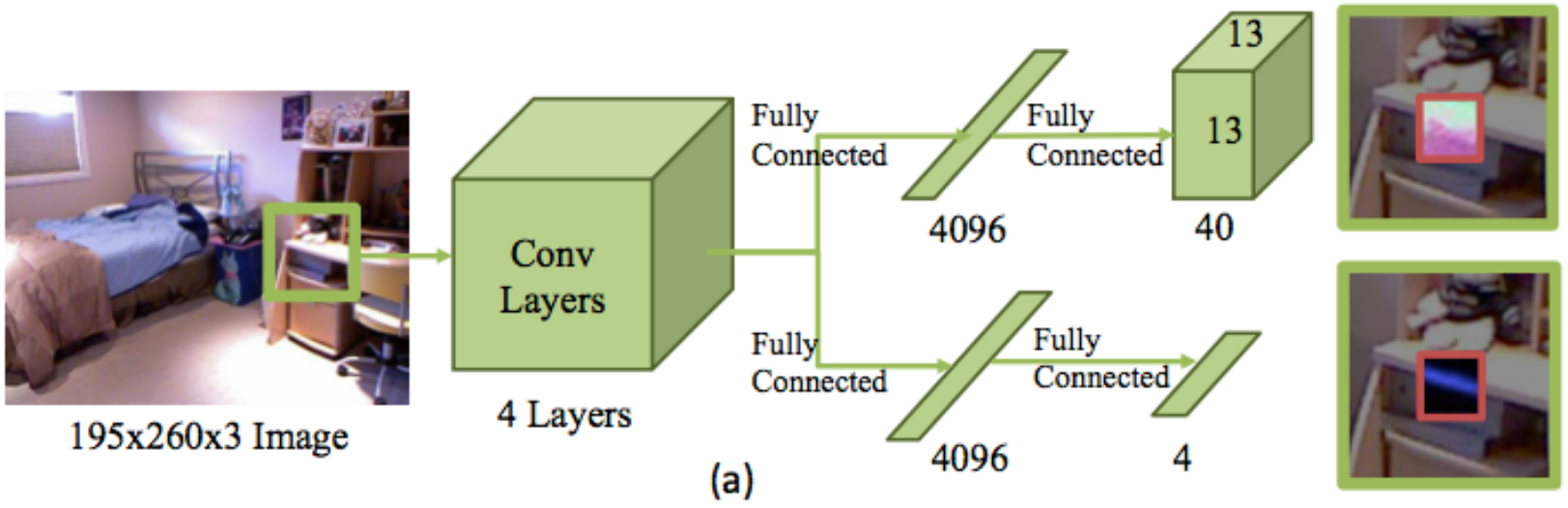
Goal: capture local evidence at a higher resolution

Input: 55 x 55 sliding window on an image (size = 195 x 260) with stride = 13

Output: (Mb = 13, Kb = 40, to capture finer details)

surface normal for M x M pixels at center of window: Mb x Mb x Kb

edge label: one for 13 x 13 pixels



Loss Functions

room layout and edge label: softmax regression

surface normal estimation:

$$L(I,Y) = -\sum_{i=1}^{M\times M}\sum_{k=1}^{K}(\mathbb{1}(y_i = k)\log F_{i,k}(I)), \quad (1)$$

$F_{i,k}$ (I): probability that i^{th} pixel has the normal defined by the k^{th} codeword

$1(y_i = k)$: indicator function, $Y = \{y_i\}$ are ground truth labels for normals

Visualization (Global Network)

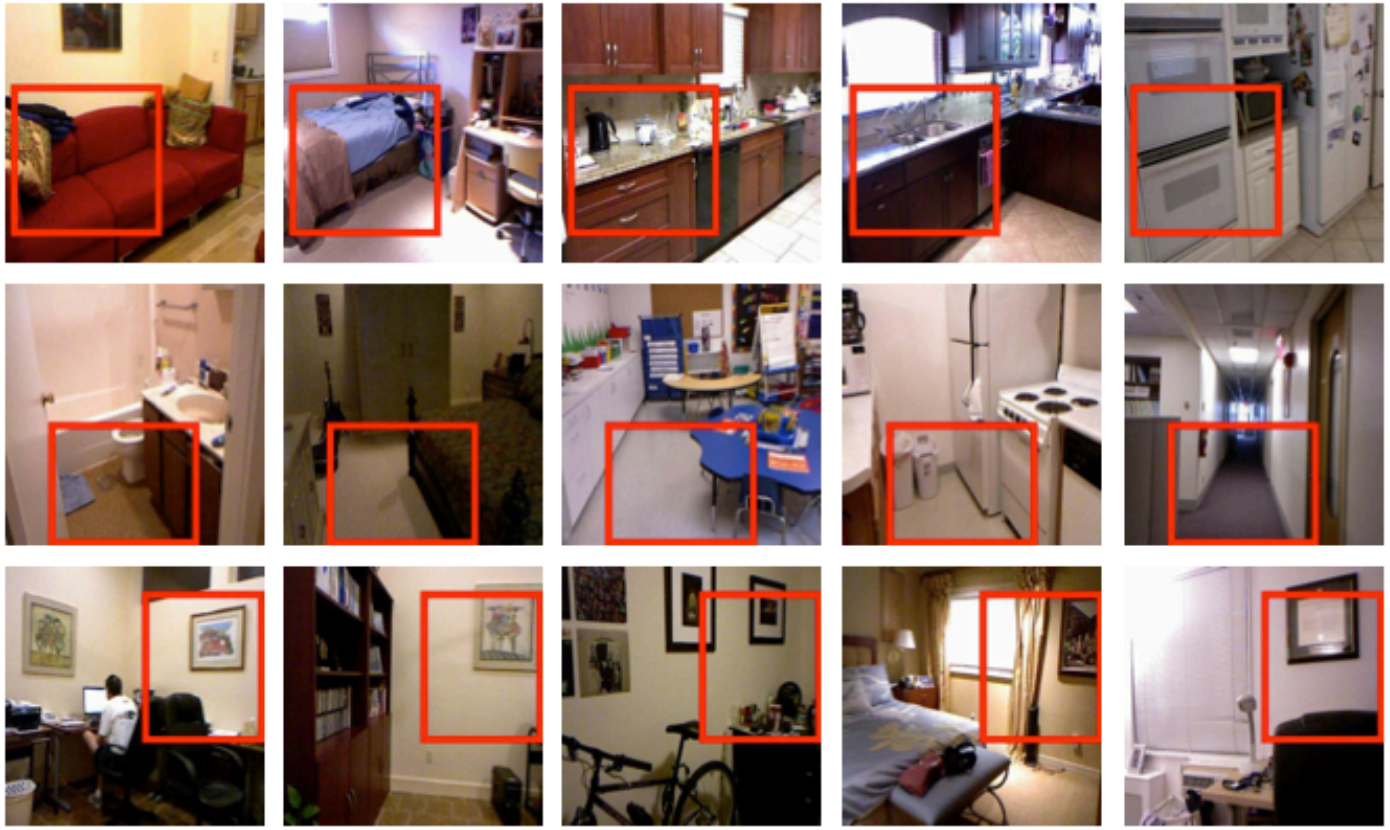
Top regions for 4th convolutional layer units

Receptive field:

31 x 31

Tendency:

Capture structure information



Visualization (Local Network)

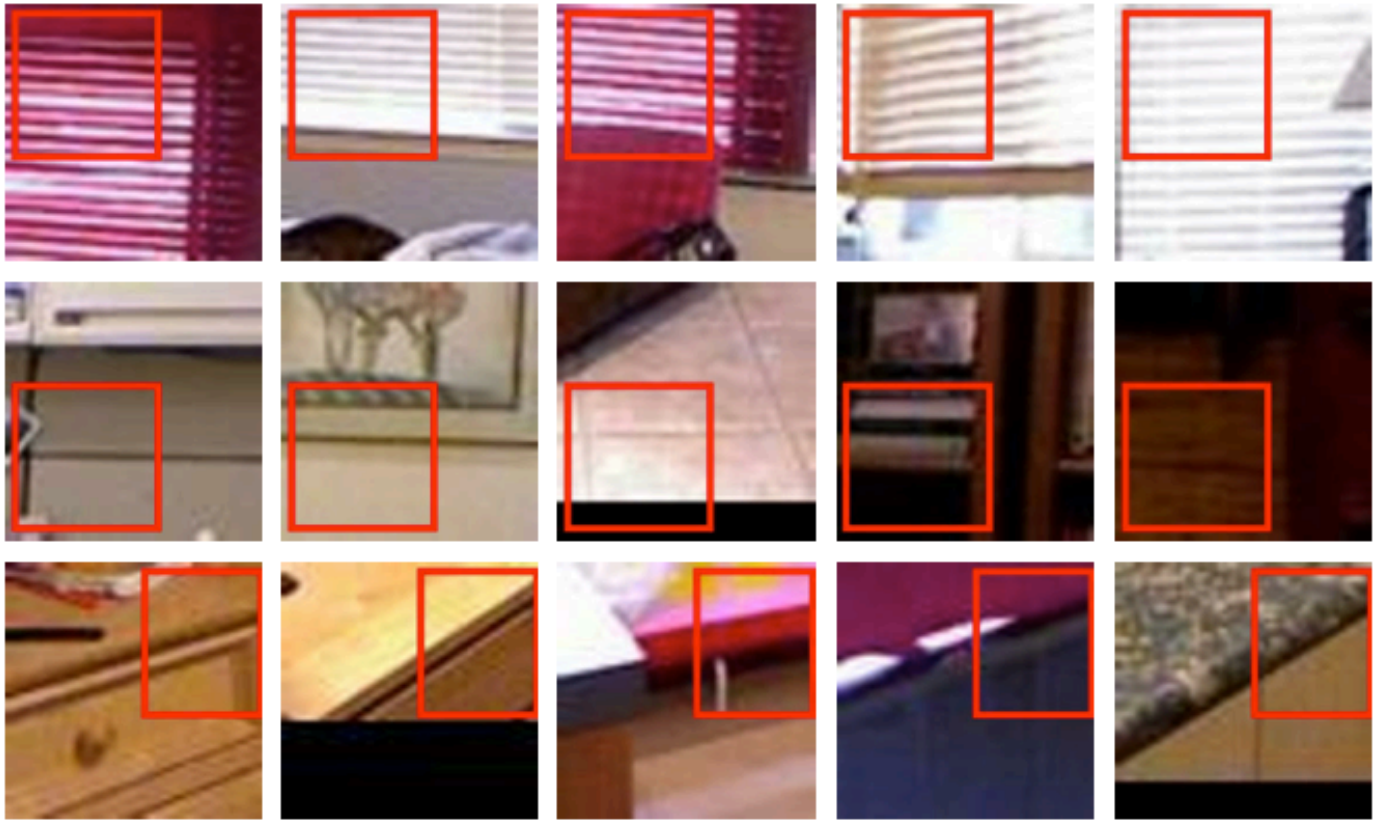
Top regions for 4th convolutional layer units

Receptive field:

31 x 31

Tendency:

Respond to local texture and edges



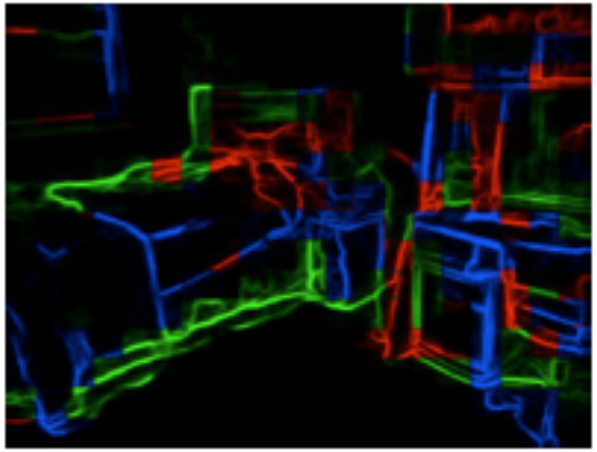
Visualization (Local Network)

After sliding window, we obtain the surface normals and edge labels for the whole image.

Blue: convex Green: concave Red: occlusion

Edge label plotting: output of Structured Edges [6].

[6] P. Dolla´r and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.



Fusion Network

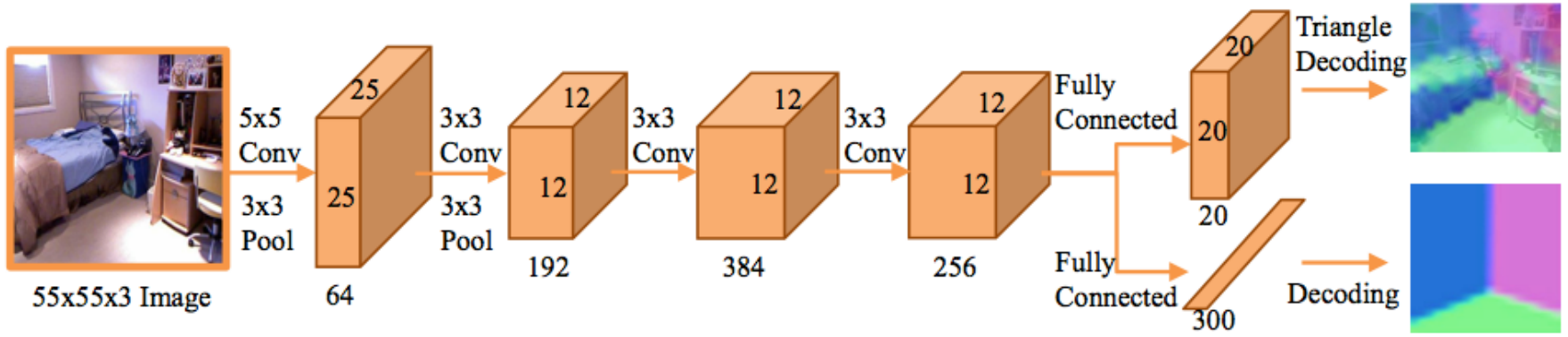
Input: 55 x 55 sliding window on an image (size = 195 x 260) with stride = 13

• Global Coarse Output: 20 x 20 with 20 classes. Decode the output to a 3-d continuous surface normal map and upscale it to 195 x 260 x 3

• Layout: 3-channel normals in the layout. Resize it to 195 x 260 x 3

• Vanishing Point-Aligned Coarse Output: vanishing points estimated by [14], yielding another feature representation with the same size.

[14] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009

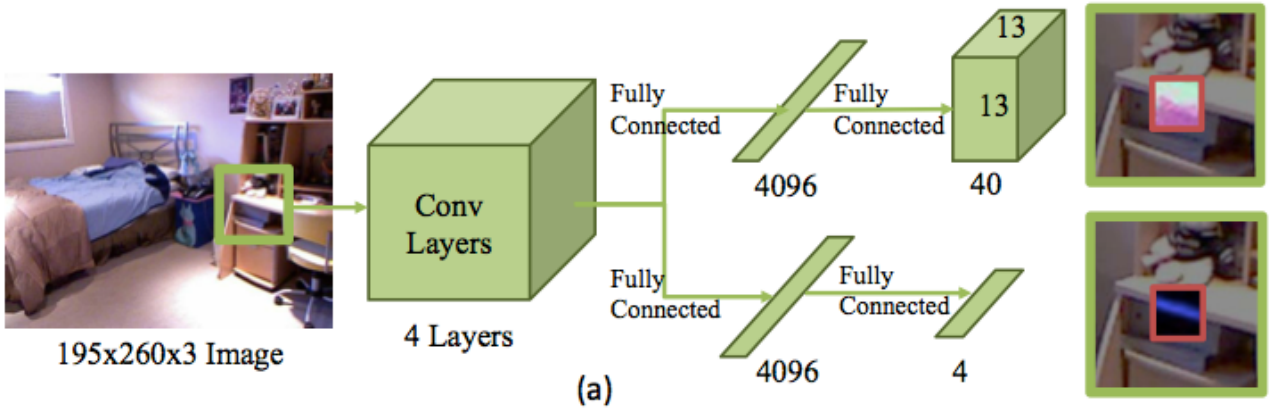


Fusion Network

Input: 55 x 55 sliding window on an image (size = 195 x 260) with stride = 13

- Local Surface Normals: 195 x 260 x 3
- Edge Labels: Upsample this 3-d vector to size 13 x 13 x 3 for each window and obtain 195 x 260 x 3 inputs. (no-edge excluded)

Final Input: 195 x 260 x 18 = (15 channels described above, original image)



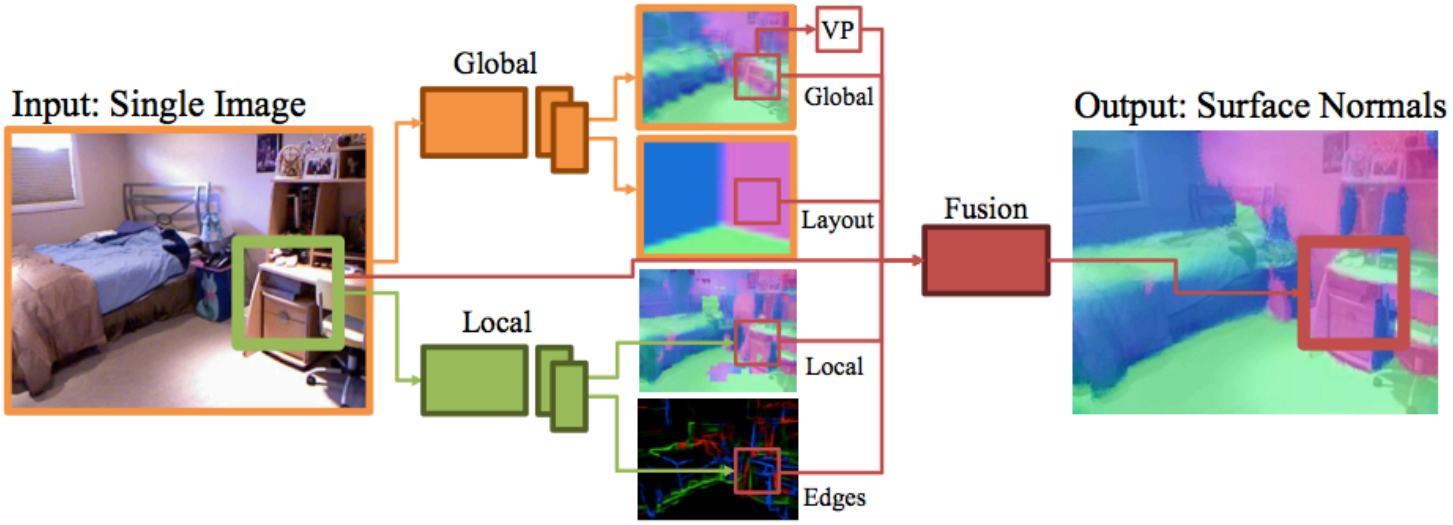
Fusion Network

Output: (Mb = 13, Kb = 40, to capture finer details)

Surface normal for M x M pixels at center of window: Mb x Mb x Kb

Testing:

Apply the fusion network on the feature maps with the stride of Mb



Training Details

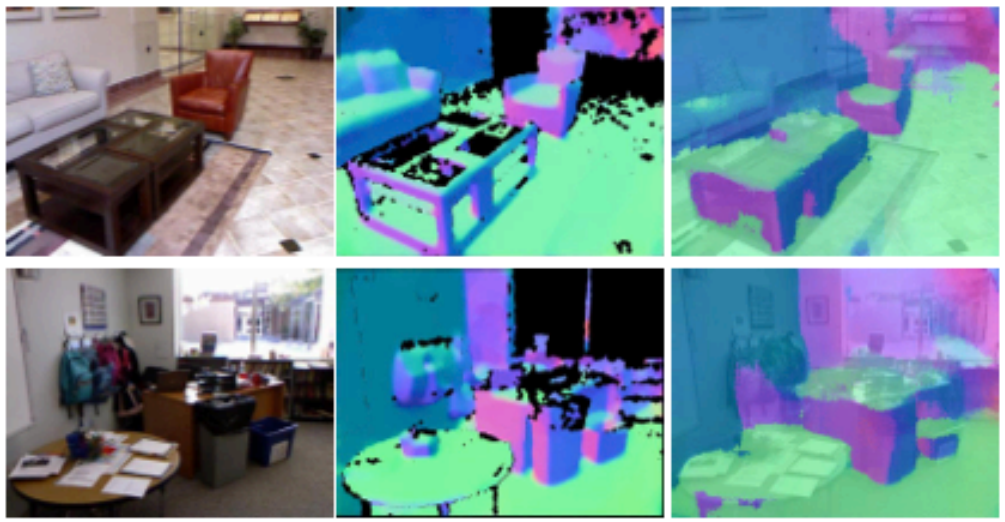
Fine-tune the network with stochastic gradient descent with learning rate sigma = 1.0×10^{-6}

During joint tuning with the layouts and edges, the learning rate set as 50*sigma

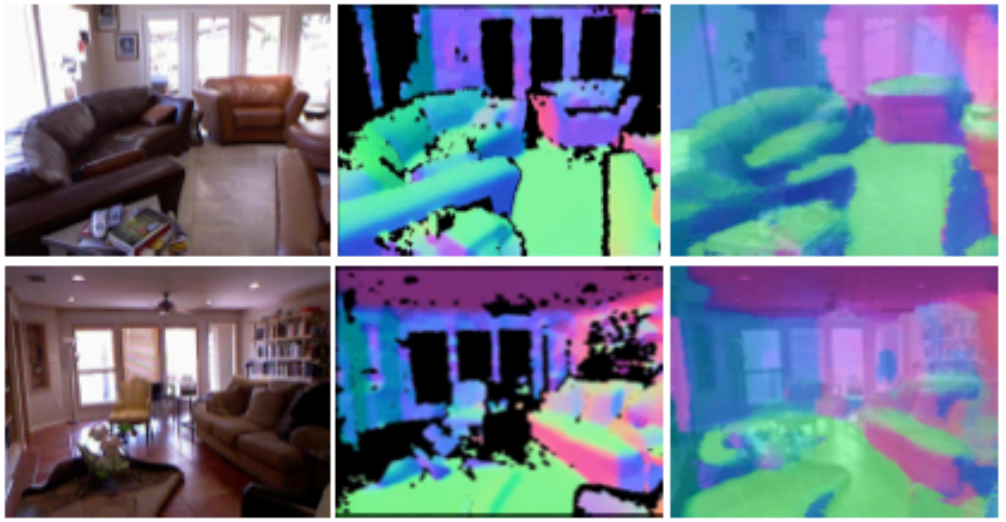
For training the local and fusion network, rescale the training images to 195 x 260 and randomly sample 400K patches with size 55 x 55 from them

Qualitative Results

1. Captures the coarse layout of the room
2. Preserves the fine details. Notice that fine details like the top of couches and the legs of table are captured.

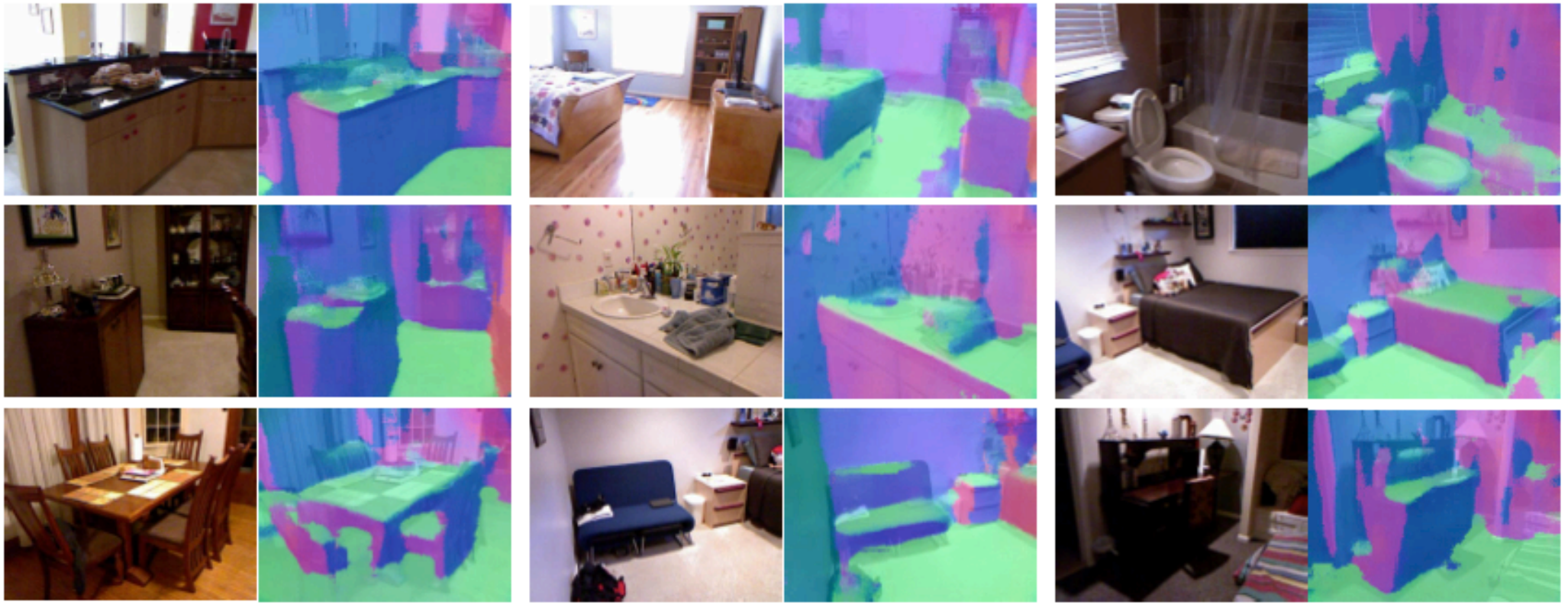


Input Ground Truth Output



Input Ground Truth Output

Qualitative Results

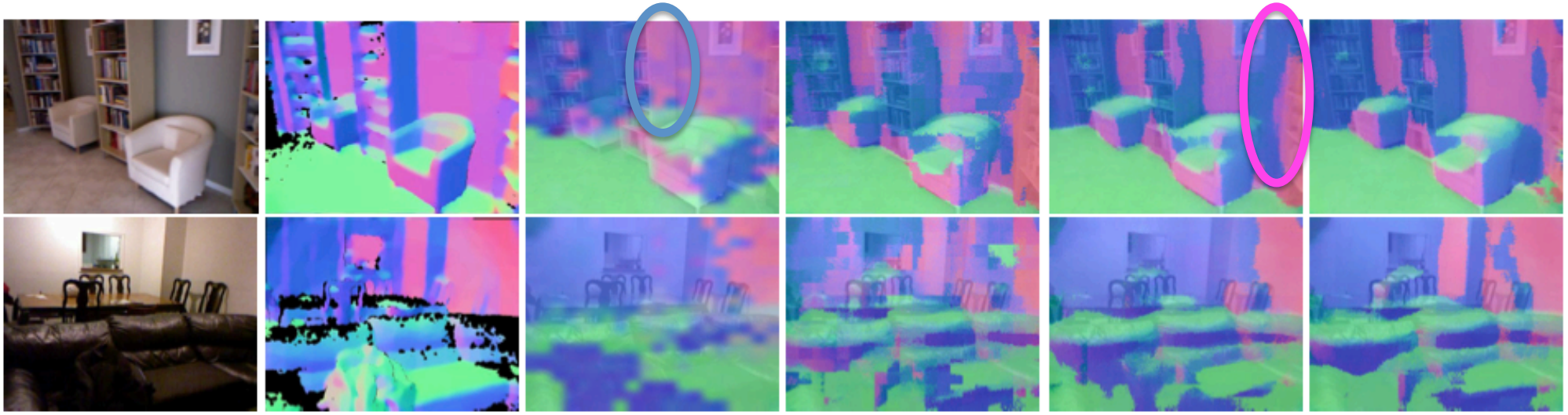


Quantitative Results

Table 1: Results on NYU v2 for per-pixel surface normal estimation, evaluated over valid pixels.

	(Lower Better)		(Higher Better)		
	Mean	Median	11.25°	22.5°	30°
Our Network	26.9	**14.8**	**42.0**	61.2	68.2
Stacked CNN [7]	**23.7**	15.5	39.2	**62.0**	**71.1**
UNFOLD [10]	35.2	17.9	40.5	54.1	58.9
Discr. [22]	33.5	23.1	27.7	49.0	58.7
3DP (MW) [9]	36.3	19.2	39.2	52.9	57.8
3DP [9]	35.3	31.2	16.4	36.6	48.2

Ablative Analysis



Input | Ground Truth | Coarse Network | Local Network | Fusion (Only Normals) | Full Fusion

Table 2: Ablative Analysis

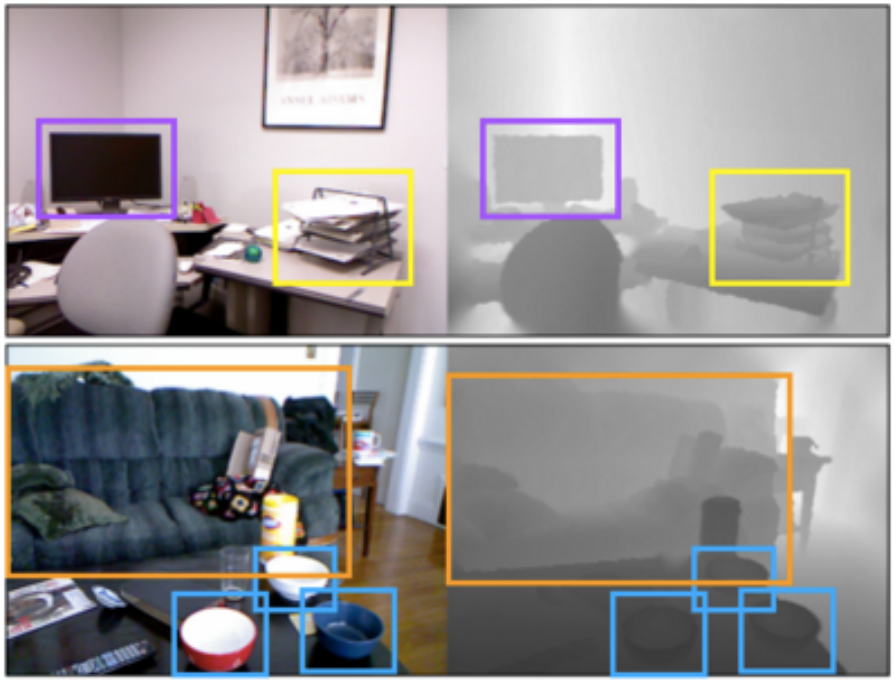
	Mean	Median	11.25°	22.5°	30°
Full	**26.9**	**14.8**	**42.0**	**61.2**	**68.2**
Full w/o Global	28.8	17.7	34.6	57.8	66.0
Fusion (+VP)	27.3	15.6	40.2	60.1	67.5
Fusion (+Edge)	27.8	16.4	37.5	59.4	67.4
Fusion (+Layout)	27.7	16.0	38.8	59.9	67.4
Fusion	27.9	16.6	37.4	59.2	67.1
Local	34.0	25.1	25.6	46.4	56.2
Global	30.9	20.8	31.4	52.3	60.5
Coarse CNN [8]	30.1	24.7	24.1	46.4	57.9

Berkeley B3DO Dataset

Mismatch in dataset bias:

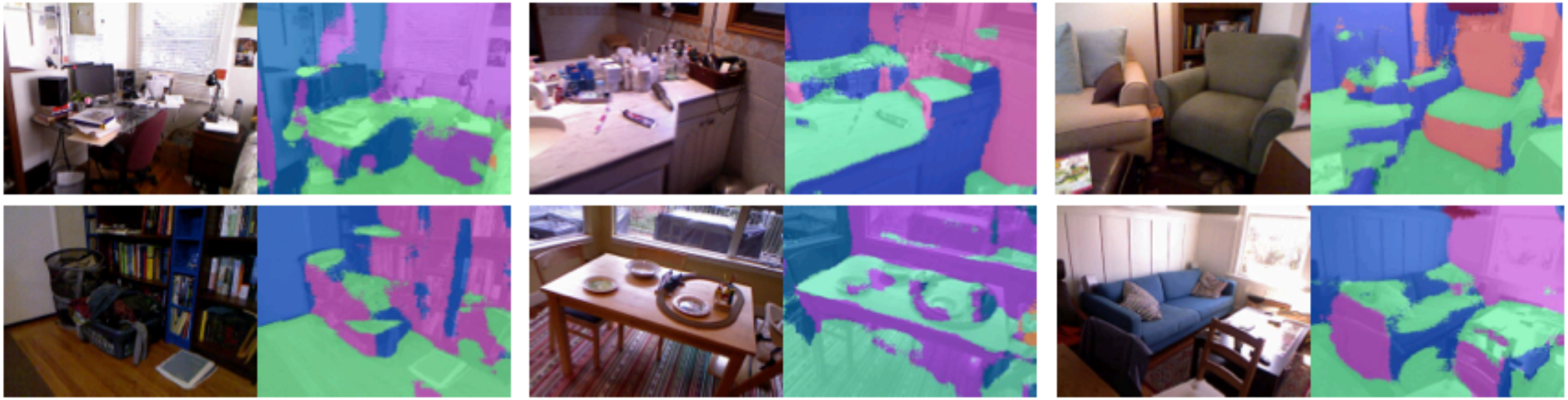
NYU: contains almost exclusively full scenes

B3DO: contains many close-ups



http://kinectdata.com/

Generalization Results to B3DO



Input Output Input Output Input Output

Table 3: B3DO

	Mean	Median	11.25°	22.5°	30°
Full	**34.5**	**20.1**	**36.7**	**52.4**	**59.2**
3DP(MW) [9]	38.0	24.5	33.6	48.5	54.5
Hedau et al. [14]	43.5	30.0	32.8	45.0	50.0
Lee et al. [25]	41.9	28.4	32.7	45.7	50.8

Possible Extensions

[7] introduced a stacked CNN model for surface normal estimation.

The contributions of this paper are complementary to [7] and combining both should provide further improvement.

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *CoRR*, abs/1411.4734, 2014.

Conclusion

1. A novel CNN-based approach for surface normal estimation
2. Designing networks based on meaningful intermediate representations and constraints can help improve the performance
3. Making Regression as Classification and adopt CNN architectures