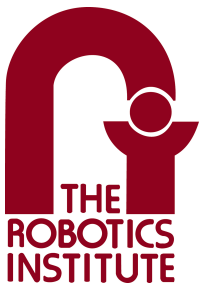
Multi-Scale Convolutional Architecture for Semantic Segmentation

Aman Raj, Daniel Maturana, Sebastian Scherer

CMU-RI-TR-15-21

September 2015



Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Advances in 3D sensing technologies have made the availability of RGB and Depth information easier than earlier which can greatly assist in the semantic segmentation of 2D scenes. There are many works in literature that perform semantic segmentation in such scenes, but few relates to the environment that possesses a high degree of clutter in general e.g. indoor scenes. In this paper, we explore the use of depth information along with RGB and deep convolutional network for indoor scene understanding through semantic labeling. Our work exploits the geocentric encoding of a depth image and uses a multi-scale deep convolutional neural network architecture that captures high and low-level features of a scene to generate rich semantic labels. We apply our method on indoor RGBD images from NYUD2 dataset [1] and achieve a competitive performance of 70.45 % accuracy in labeling four object classes compared with some prior approaches. The results show our system is capable of generating a pixel-map directly from an input image where each pixel-value corresponds to a particular class of object.

Contents

1 Introduction

Perception is an integral part in all robotic systems that try to understand the environment. Scene understanding is a central problem in perception having many different aspects such as semantic labels describing the identity of various scene regions; affordances capturing interaction information of a robot with the environment; and depth describing the physical geometry of the scene. Accurate scene understanding can help a robot to learn about the rich features of its surrounding which can further assist in path planning & navigation, detecting good grasp to pick up an object, to achieve automation in decision making and can improve task-relevant 3D perception. Our work is focused on semantic labeling–pixel-wise classification in an image, for scene understanding to aid vision in robotics. Semantic labeling in a 2D image involves pixel-wise classification among a possible number of classes of objects chosen in a scene. We consider the use of the RGBD information for this task. While the RGB data provides the various region proposals of an object present in an image, the depth data is useful for understanding it's geometry and shape. A combination of this information helps our system to identify the various regions in an image that can possibly belong to a particular class e.g. a wall is always vertical in shape but can have different colors.

We also explore the necessity of depth information for our scene understanding framework by experimenting with the learning of a ConvNet with two different inputs – RGB and RGBD. We draw rich representation from the RGBD information of indoor scenes using the method of [2], [3] in the form of geocentric encoding of depth images and train our proposed deep ConvNets inspired from [4] to create a benchmark in learning that can generate a pixel-map containing semantic labels of the object visible at each pixel. Our approach uses a multi-scale deep convolutional neural network architecture and generates a pixel-map directly from an input image. We demonstrate the results of our experimentations with deep ConvNets on the NYUD2 dataset introduced in [1] that contains more than a thousand of annotated indoor scenes. We believe such advances in scene understanding will foster the research in the field of perception in robotics.

2 Related Work

Convolutional neural networks have been used recently for a number of applications such as image and video recognition [5], [6], bounding box object detection [7], key point prediction [8], and large-scale image classification task [4], [9], [10]. A Hypercolumn of activated features for each pixel in all the CNN units in an architecture can be used as a descriptor for simultaneous detection and segmentation task [11]. The works of authors in [12], [13] use deep learning CNN for the simultaneous classification of objects and finding good grasp for a robotic hand. The process of 3D object recognition devised by [14] uses a transfer learning approach to train CNNs on four separate data sets formed using RGBD information (splitting into R, G, B, D separate channels) and fuses their results to make a final prediction.

In recent years, there has been a significant use of convolutional neural network framework in the area of scene understanding by semantic labeling. Semantic segmentation is one of the important cues for scene understanding in [15], [16], [7]. Fully convolutional network architectures by Long *et al.* [17] for this task shows a state of the art

results on PASCAL VOC and NYUD2 datasets, adapting some contemporary classification networks. The multi-scale convolutional network architecture proposed in [18] uses the information from both shallow and coarse layers to predict surface normals, depth, and semantic labels. Also, some super-pixel based methods have been used for object recognition and segmentation of indoor RGBD scenes. In the work of [19], the authors formulate this task as a binary object-background segmentation and use an informative set of features and grouping cues for small regular super-pixels. Currently, most of the works in literature have achieved semantic segmentation in 2D RGBD scenes which can be transferred into a 3D reconstruction using Bayesian updates [20] and dense Conditional Random Fields (CRFs) [21] as proposed in [22].

Advances in 3D sensing and availability of low-cost 3D sensors like Kinect have made it possible to record depth information along with RGB images. As a result, the use of RGBD data for semantic segmentation task has received tremendous attention during past couple of years as depth provides an extra feature in the form of the geometry of the scene. Depth information has been used as an additional channel with RGB in works of [23], [19], [24], [22], [3] and [2]. Socher *et al.* proposed a combination of CNN (Convolutional Neural Net) and RNN (Recursive Neural Net) in [25] for learning features on RGBD images for 3D object classification. There have been various approaches to represent a depth image. The HHA encoding of depth image proposed in [3], [2] is used as geocentric features in combination with contour cues to achieve contour detection and semantic segmentation. On the other hand Hoft *et al.* in [24], use a simplified version of the histogram of oriented gradients (HOG) descriptors to create a histogram of oriented depth (HOD [26]) to represent depth channel from RGBD scenes. The current dataset benchmark suites like NYUD2 are limited in size presenting a critical bottleneck for further advancement in scene understanding including semantic segmentation, context reasoning [27], surface orientation and room layout estimation [28], [29]. SUN RGBD benchmark suite introduced by Song *et al.* [15] attempts to overcome this bottleneck.

3 Approach

Our scene understanding framework performs pixel-wise semantic classification of images from NYUD2 dataset according to four classes - “furniture”, “floor”, “structure”, and “props”. The input to this framework is RGBD (encoded form) data and expected output is the corresponding labeled 4-class ground truth image. We first describe our chosen way to represent the depth images as per [3] in *Section-3.1* and then our proposed deep ConvNet architecture for this task inspired from [4] in *Section-3.2*. We further report the training, evaluation and implementation details of our scene understanding framework in *Section-3.3–3.4*.

3.1 HHA Encoding of Depth

Depth images taken from 3D sensing devices contain the information about the geometry of a scene. Geocentric encoding of depth images in three-dimensional form is analogous to RGB channels, providing more rich representation about a scene geometry than simply using depth images. Such encoding is based on the fact that direction of gravity affects the shape of objects around us and hence it’s position in the scene from the floor. We

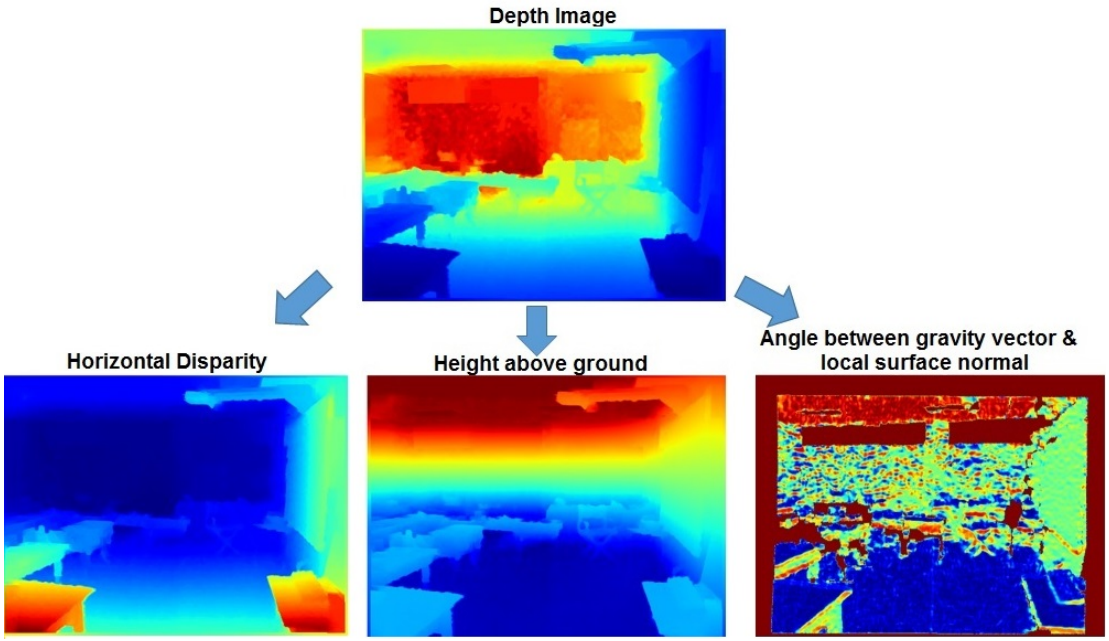


Figure 1: Geocentric encoding-HHA components of a depth image

adapt the method of [2] to obtain geocentric features, where each pixel in depth image is encoded with three channels namely – H-horizontal disparity, H-height above the ground, and A-angle the pixel's local surface normal makes with the calculated gravity direction at the pixel. Horizontal disparity helps to understand the closeness of an object from camera, height above the ground tells the possible positions of an object in the scene with respect to the ground plane and A-quantify the shape of an object in the scene. This way from a single depth image we are able to obtain 3-geocentric features. All channels are linearly scaled to map the obtained values across 0 to 255 range. In order to calculate A-channel we first use the following algorithm as proposed in [3] to calculate the direction of gravity. The algorithm starts with an initial estimate of gravity vector along Y-axis and iteratively improves the estimates via following 2 steps.

1. Using current estimate of gravity direction g_{i-1}, make hard-assignments of local surface normals in aligned set $\mathcal{N}_{||}$ and orthogonal set $\mathcal{N}_{\perp}$ such that

$$\mathcal{N}_{||} = \{\mathbf{n} : \Theta(\mathbf{n}, \mathbf{g_{i-1}}) < d \quad or \quad \Theta(\mathbf{n}, \mathbf{g_{i-1}}) > 180^\circ - d\}$$
$$\mathcal{N}_{\perp} = \{\mathbf{n} : 90^\circ - d \quad \Theta(\mathbf{n}, \mathbf{g_{i-1}}) < 90^\circ + d\}$$

   where $\mathcal{N}_{||}$ contains normals from points on floor and furniture tops, $\mathcal{N}_{\perp}$ contains normals from points on walls and d is threshold angle made by local surface normals with g_{i-1}. Stack the vectors in $\mathcal{N}_{||}$ and $\mathcal{N}_{\perp}$ to form matrix $\mathcal{N}_{||}$ and $\mathcal{N}_{\perp}$ respectively.

2. Estimation of new gravity vector g_i which is as aligned to normals in $\mathcal{N}_{||}$ set and as orthogonal to normals in $\mathcal{N}_{\perp}$ set as possible which corresponds to solving optimization problem of finding eigen-vector with the smallest eigen value of 3×3 matrix, $\mathcal{N}_{\perp}{\mathcal{N}_{\perp}}^t - \mathcal{N}_{||}{\mathcal{N}_{||}}^t$.
   The algorithm is run for first 5 iterations with $d = 45^\circ$ followed by another 5 iterations with $d = 15^\circ$ to obtain final value of gravity vector $\mathbf{g_f}$.

Once the gravity vector is computed the angle between $\mathbf{g_f}$ and local surface normal for each pixel is calculated. For calculating height above the ground, the lowest point in image is considered at the level of supporting the ground plane and height of each pixel above the ground is computed. Horizontal disparity is calculated using conventional method as difference between the images taken from left and right camera.

3.2 Architecture Development

Deep Convolutional Neural Networks of [10], [9], [4] have achieved a state of the art results on image classification task in Imagenet Challenge that contains around thousands of classes of objects. The VGG net of [4] has outperformed other ConvNets entries in *Imagenet Challenge 2014* in localization task. There are few works in literature that use deep convolutional neural networks for semantic segmentation like those of [17], [18] and [30]. We adapt VGG16 (having 16 weight layers) ConvNet a variant of VGG ConvNet, proposed by Simonyan & Zisserman in [4] for our purpose. Since this deep ConvNet is trained for image classification task, so we modified it for our semantic scene segmentation task in a two-step process. First we remove the last fully connected layers from VGG16 to make a network that we call - VGG16-FC presented in *Section-3.2.1* and lastly we modified this VGG16-FC architecture to obtain another network that we call - VGGM ConvNet presented in *Section-3.2.2*.

3.2.1 VGG16-FC architecture

In order to obtain VGG16-FC, we replaced all the last fully connected layers of VGG16 net with a single convolution layer that is based on "Network in Network" architecture of Lin *et al.* [31]. This layer performs convolution over an area of size 1×1 and has a total number of four filters which corresponds to four possible classes for each pixel - this is represented by the output layer in the architecture. We kept the filter parameters of the rest of convolutional layers unchanged. Each stack of convolutional layer like - *Conv(3,64)* | *Conv(3,64)*, is followed by a pooling layer, not shown in the figure. We use max-pooling over a 2×2 window, with stride 2. The notation *Conv(3,64)* - means a convolutional layer with *filter-size* - 3×3 and *number of filters - 64*. This architecture is presented in the *Figure 2*.

3.2.2 VGG-M architecture

This architecture is a modification of VGG16-FC. Similar to the works of [18], [32], we employ a multiscale deep ConvNet which first predicts a coarse global output using the VGG16-FC net, then refines it using a finer-scale local network. We first upsample the coarse prediction output of *Scale-2* to double the final output resolution of network and concatenate it with low-level predictions from shallow network of *scale-1* to get both high and low-level features – lastly we refine this prediction using a stack of three convolutional layers – each having *filter-size* - 3×3 and *number of filters - 64*. This is is our proposed architecture that we call VGG-M. As shown in [18] such network architecture is capable of learning more robust features. This architecture is shown in *Figure 3*.

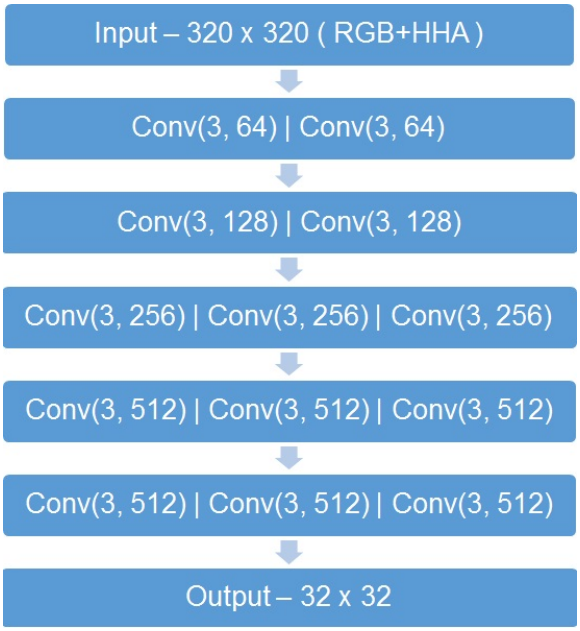


Figure 2: VGG16-FC ConvNet Configureations. Pooling layers not shown for brevity.

3.3 Training and Evaluation

Dataset

We have used NYUD2 dataset for training and evaluation of our proposed architecture. It has 1449 RGBD images taken from a Microsoft Kinect, containing pixel-wise annotated ground truth image sets with 4, 13 and 40 labels. We perform 4-class segmentation task that uses high level category labels namely - "floor", "furniture", "structure" and "props". The results on the standard split of 795 training images and 654 testing images are reported.

Training

We compute the mean value of pixels in RGB and HHA over the training set images and subtract it from each image as pre-processing operation. Since the number of images in NYUD2 dataset is very less, so the weight of convolutional layers were initialized with the weights from VGG16 ImageNet model that has been trained on millions of images. The weights of the layer in *Scale-1* of VGG-M was randomly initialized with biases set to zero initially. The images in the dataset were scaled to obtain a fixed-size 320×320 and a stack of RGB and HHA (calculated from depth image) components of an image concatenated in pixel-space were given as input along with ground truth provided as a target. Standard SGD (Stochastic Gradient Descent) is used for generating losses. It is carried out by using the method which optimizes the multinomial logistic regression objective with mini-batch gradient descent with momentum (based on back-propagation algorithm of [33]). The batch size was set to 10 and momentum to 0.9. The training was regularized by weight decay, using L_2 penalty multiplier set to $5\times\ 10^{-4}$. The learning rate was initially set to 10^{-3} and then decreased by a factor of 10 when the test set accuracy stopped improving. Overall, the learning rate was decreased 2 times, and the learning was stopped after 120 epochs. We also apply some random data transforms on input and ground truth to augment the training data.

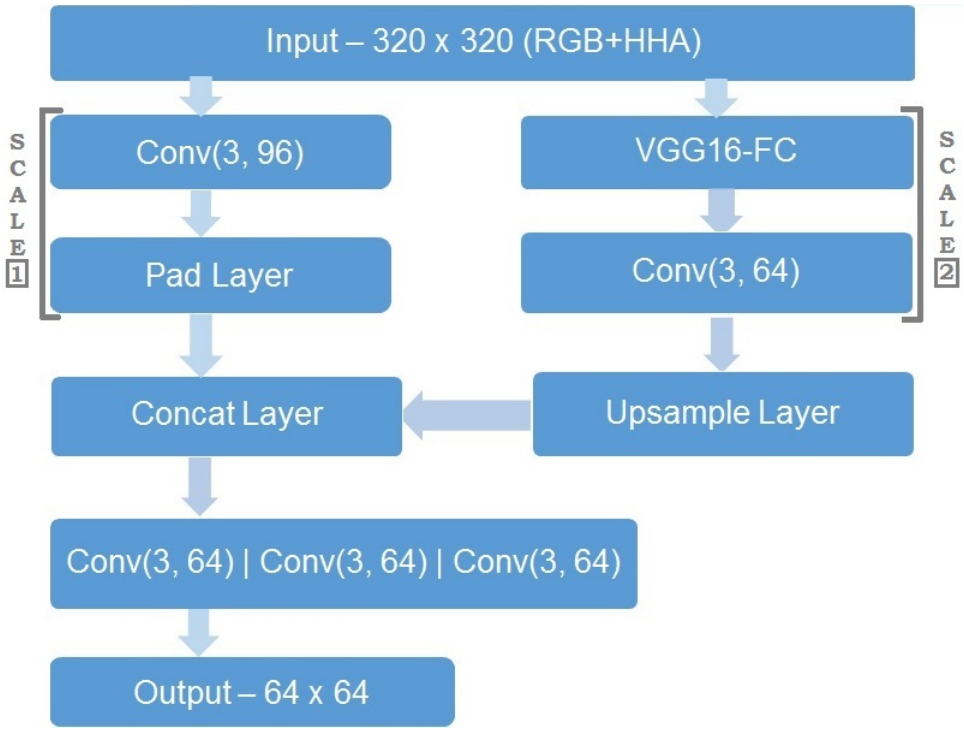


Figure 3: VGG-M ConvNet Configureations. Pooling layers not shown for brevity.

Evaluation
After training the network, the evaluation is performed on the standard 654 testing images of the dataset. First, these test images are scaled to match the fixed input size of ConvNet, then RGB and HHA components of each image are calculated and then given as input to the network in the form of a stack by concatenating them in pixel-space. This way the network is applied to obtain a final output image having dimension as per the network architecture being evaluated and having 4-class semantic labels.

3.4 Implementation Details

Our implementation is performed using publicly available Python library for deep learning Theano [34]. Caffe was used to read VGG16 Imagenet model. The three-dimensional HHA encoding of depth images is calculated using publicly available source code with slight modification to suit our need. On a system equipped with NVIDIA Tesla K40c, the training of the ConvNet VGG-M took 2-3 days.

4 Performance Experiments

We report the performance of our proposed architecture VGG-M on semantic labeling.

4.1 Accuracy Analysis

Pixel accuracy is calculated on the test set of NYUD2 dataset. During evaluation process, the predicted output of the network is compared with the corresponding ground truth

image and the number of correctly classified pixels is enumerated to generate an accuracy score given by

$$Accuracy\ Score = \frac{No.\ of\ correctly\ classified\ pixels}{Total\ No.\ of\ pixels\ in\ ground\ truth}$$

This way accuracy score for all the 654 testing images is calculated and then averaged to obtain an average accuracy score. Further, we use a *multiplier-100* to convert it into the percentage.

4.2 Semantic Labels

We finally apply our method - VGG-M architecture to semantic segmentation problem on the NYUD2 dataset. Since this data provide us with depth channel we use RGB and encoded Depth as input as described in *Section-3.3*. We evaluate our method for 4-Class segmentation task that uses high category labels – "floor", "structure ","furniture" and "props" as described in [16]. We also compare our method with recent prior proposed methods using the pixel accuracy metric as shown in *Table-1*. We outperform the prior methods of Couprie *et al.* and Khan *et al.*. In our future work, we hope to improve the system to bring its performance competitive to Gupta *et al.* and Eigen *et al.* Qualitative results are shown in *Figure-4*.

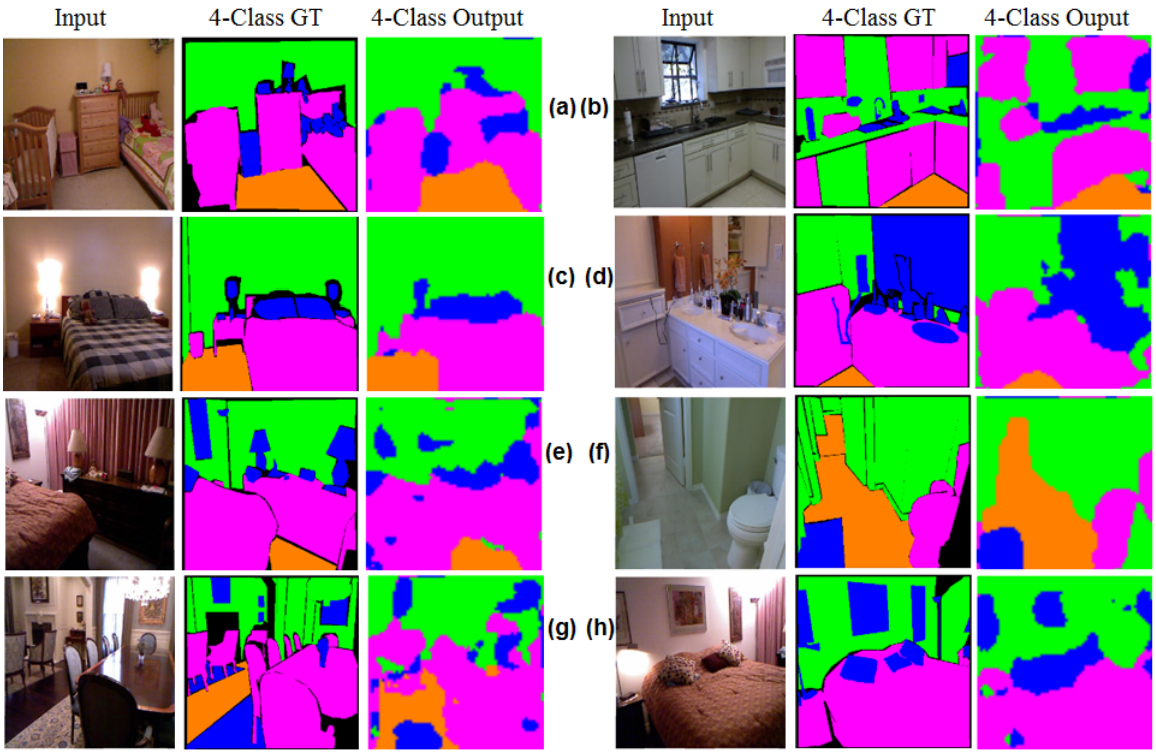


Figure 4: Semantic labeling results: For each half - we have rows with each row having Input, 4-Class ground truth, and 4-class labeling result. Note we use our proposed VGG-M architecture with RGB+D(HHA) as input.

4-Class Semantic Segmentation	
	Pixel Accuracy %
Couprie *et al.* [35]	64.5
Khan *et al.* [36]	69.2
Ours (VGG-M)	**70.45**
Gupta *et al.* [3]	78
Eigen *et al.* [18]	80.6

Table 1: Semantic Labeling on NYUD2 with 4 classes

5 Probe Experiments

5.1 Effect of Depth Input

The fact that only RGB image can be used as input to a network to learn semantic labels leads to the question: How important is the depth input with relative to RGB in semantic labeling task? To study this, we perform our experimentation with a shallow convolutional neural network as shown in *Figure 5* having few numbers of convolutional layers. The network takes 320×320 dimensional input and gives output of shape 32×32. We train and evaluate this network for 4-class semantic labeling task as per the outline described in *Section-3.3* (we don not initialize with any Imagenet weights) - 1)With RGB as input and 2) With RGBD as input, concatenating the depth with RGB in pixel space. We observed

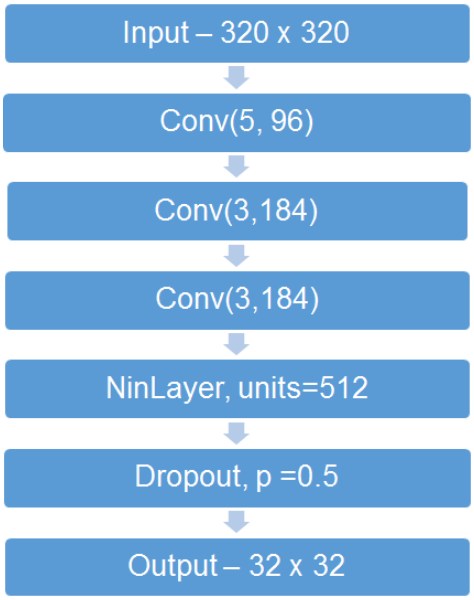


Figure 5: ConvNet to study the effect on the performance of a network when depth is used in input.

from *Figure 6*, while using only RGB data as input training loss of the network saturates at a $value > 1.0$ on the other hand when we introduce depth as an additional channel with RGB as input, it saturates at a $value < 1.0$, resulting in an appreciable decrease in losses. Also, the performance of the network improves by a significant margin on accuracy scale this is shown in *Table-2*. This implies that ConvNet is able to learn a more accurate

representation of indoor scenes in the form of correct semantic labels when we use depth information as an additional channel along with RGB, as depth provides an extra feature in the form of geometric representation of the scene.

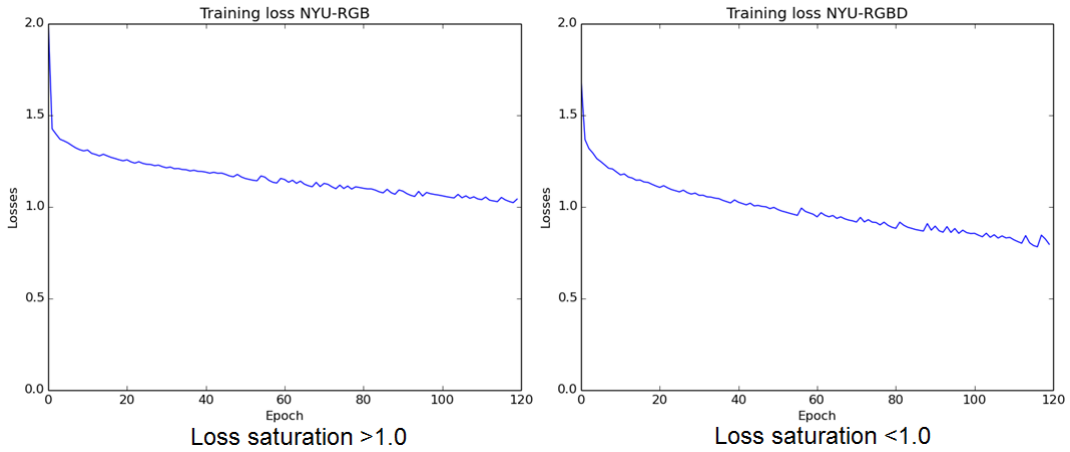


Figure 6: Training loss analysis when the input is 1) RGB and then 2)RGB+D. Note we use the network architecture of *Figure-5*.

Effect of Depth Input	
Input	Accuracy %
RGB only	62.29
RGB+Depth	65.02

Table 2: Comparison of performance of the network shown in *Figure-5* with different sets of inputs for 4-class semantic labeling task.

5.2 Contribution of Scale

In order to study the contribution of scale-1 to the performance of our proposed architecture VGG-M, we compare the performance of both the models - VGG16-FC (contains scale-2 only) and VGG-M (contains scale-1 and scale-2). We train and evaluate both networks for 4-class semantic labeling task as per the outline described in *Section-3.3* with RGB + D(HHA) given as input. While comparing the training losses of both the architectures we observe a progressive improvement in the decrease of losses as it plunges from always a $value > 0.7$ in the case of VGG16-FC to a $value < 0.2$ in the case of VGG-M and also the latter took less number of epochs to converge. This is shown in *Figure 7*. Performance measurement on accuracy metric highlights that VGG-M outperforms VGG16-FC with approximately 4.5% improvement in prediction accuracy, refer *Table 3*. We infer the introduction of another scale-1 in the case of VGG-M serves two purpose – 1)It captures low-level features from the image providing a global view which is combined with high-level features from the scale-2 output and 2)It also helps to double the resolution of the final output of the model providing a more fine-tuned labels.

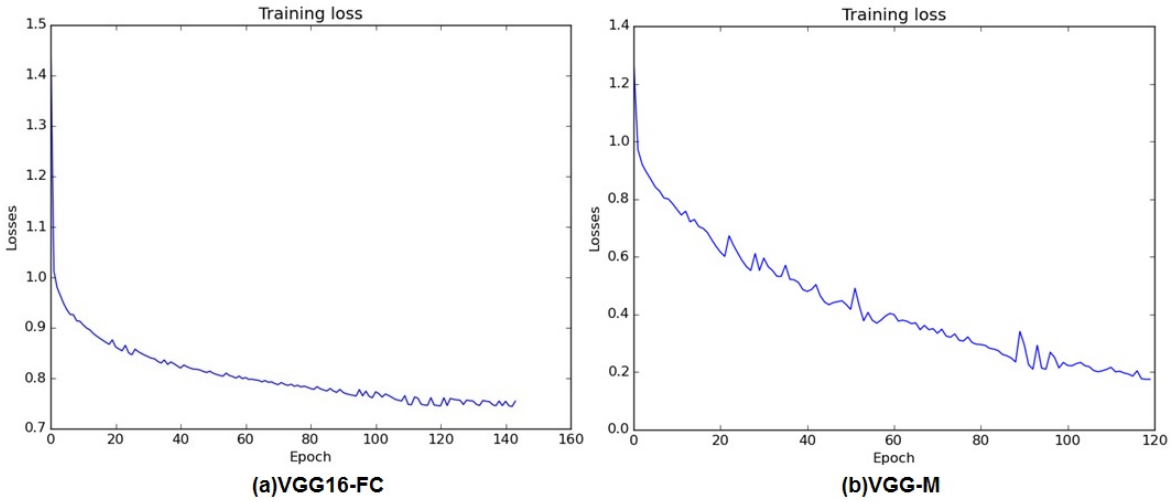


Figure 7: Training loss analysis of ConvNet VGG16-FC (having scale-2 only) & VGGM (having scale-1 and scale-2) to study the contribution of scale-1. Note input is RGB+D(HHA).

Contribution of Scale	
Architecture	Accuracy %
VGG16-FC (scale-2 only)	66.07
VGG-M (scale-1 + scale-2)	70.45

Table 3: Comparison of models having different scales for 4-class semantic labeling task.

6 Conclusion

We have proposed a multi-scale deep ConvNet - VGGM for our scene understanding task which performs semantic segmentation of indoor scenes from NYUD2 dataset. This multi-scale network can accurately predict semantic labels directly when a single RGB image along with it's recorded depth information are given as input. We exploit the geocentric encoding of a depth image (HHA) that splits the depth image into three geocentric feature map - providing a rich geometric view of a scene. This helps our model to learn more robust features about the scene aiding the semantic labeling task. We also studied that use of depth information along with RGB data greatly improves the training process as well as overall performance of a model. Also, a multi-scale architecture approach in our task certainly improves the prediction of semantic labels by providing a global view of image. Overall we infer that our findings aids the scene understanding. One drawback of our approach is that it is more prone to giving erroneous output when there is too much clutter in scene as shown in results in *Figure 4* (b), (d). We hope to overcome this in our future work.

7 Future Work

Since our current architecture fails in scenes having very high clutter, so in our future work we aim to remove this bottleneck and also improve the accuracy of predictions to

get more accurate labels. We plan to use the architecture as shown in *Figure 8* with the hypotheses that HHA encoding of a depth image have enough similar structures with RGB, and we hope to learn some unique features from RGB & HHA separately and fuse them to form a coarse prediction with further refinement with a local network.

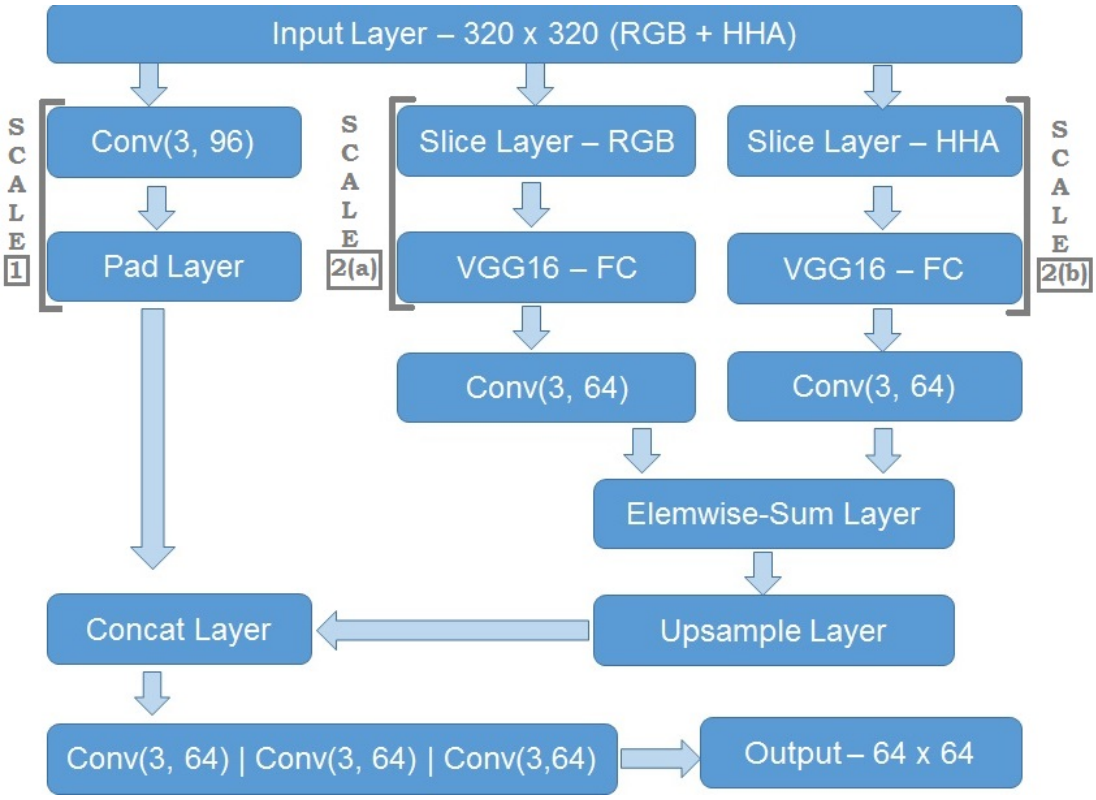


Figure 8: Architecture of future interest

References

[1] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 601–608. IEEE, 2011.

[2] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.

[3] Swastik Gupta, Pablo Arbelaez, and Jagannath Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 564–571. IEEE, 2013.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[5] Sudheendra Vijayanarasimhan, Prateek Jain, and Kristen Grauman. Far-sighted active learning on a budget for image and video recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3035–3042. IEEE, 2010.

[6] Rama Chellappa, Ashok Veeraraghavan, and Gaurav Aggarwal. Pattern recognition in video. In *Pattern Recognition and Machine Intelligence*, pages 11–20. Springer, 2005.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[8] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[11] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. *arXiv preprint arXiv:1411.5752*, 2014.

[12] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. *arXiv preprint arXiv:1412.3128*, 2014.

[13] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International gJournal of Robotics Research*, 34(4-5):705–724, 2015.

[14] Luís A Alexandre. 3d object recognition using convolutional neural networks with transfer learning between input channels. In *Proc. the 13th International Conference on Intelligent Autonomous Systems*, 2014.

[15] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015.

[16] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012*, pages 746–760. Springer, 2012.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[18] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *arXiv preprint arXiv:1411.4734*, 2014.

[19] Md Alimoor Reza and Jana Kosecka. Object recognition and segmentation in indoor scenes from rgb-d images. In *Robotics Science and Systems (RSS) conference-5th workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2014.

[20] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics (intelligent robotics and autonomous agents series). *Inteligent robotics and autonomous agents*, 2005.

[21] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *arXiv preprint arXiv:1210.5644*, 2012.

[22] Alexander Hermans, Georgios Floros, and Bastian Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2631–2638. IEEE, 2014.

[23] Max Schwarz, Hannes Schulz, and Sven Behnke. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features.

[24] Nico Höft, Hannes Schulz, and Sven Behnke. Fast semantic segmentation of rgb-d scenes with gpu-accelerated deep neural networks. In *KI 2014: Advances in Artificial Intelligence*, pages 80–85. Springer, 2014.

[25] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep learning for 3d object classification. In *Advances in Neural Information Processing Systems*, pages 665–673, 2012.

[26] Luciano Spinello and Kai O Arras. People detection in rgb-d data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3838–3843. IEEE, 2011.

[27] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1417–1424. IEEE, 2013.

[28] David F Fouhey, Arpan Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3392–3399. IEEE, 2013.

[29] David Ford Fouhey, Abhinav Gupta, and Martial Hebert. Unfolding an indoor origami world. In *Computer Vision–ECCV 2014*, pages 687–702. Springer, 2014.

[30] Holger R. Roth, Le Lu, Amal Farag, Hoo-Chang Shin, Jiamin Liu, Evrim Turkbey, and Ronald M. Summers. Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. *CoRR*, abs/1506.06448, 2015.

[31] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[32] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014.

[33] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[34] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[35] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[36] Salman Hameed Khan, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri. Geometry driven semantic labeling of indoor scenes. In *Computer Vision–ECCV 2014*, pages 679–694. Springer, 2014.