

CS229 Project:

3-D Image segmentation using Recursive Neural Networks (RNNs)

Vinayak Agarwal
Stanford University
vinayak1@stanford.edu

Anand Kamat Tarcar
Stanford University
anandkt@stanford.edu

Abstract

This goal of this project was to perform segmentation of indoor scene images, using not only the traditionally used 2 dimensional image features but also the 3 dimensional image features as obtained by a structured light sensor in the form of Microsoft Kinect. We explore the segmentation accuracies using some of most popular image descriptors for segmentation purposes such as Color Histograms, Scale Invariant Feature Transform (SIFT)[8], Histogram of Oriented Gradients (HoG)[9] on both the RGB and Depth images. In addition, we also explore how a trained Recursive Neural Network Model[2] can be used to segment an indoor scene with better accuracy than traditional methods like Conditional Random Fields (CRF)[13]. Each segment is represented with 3D features containing information about color, depth, gradients etc. The accuracy is determined on the NYU Labeled Dataset [3].

1. Introduction

Image Segmentation is a technique in Computer Vision that involves extracting meaningful information such as objects – both foreground and background from images. It has a wide range of applications such as segmenting out tumors using medical imaging modalities, biometric recognition systems, pedestrian detection systems etc.

It has been observed that natural scenes images have inherent recursive structure that can be exploited to identify units or segments and also discover how these units can be combined to form the root image[2]. In addition, in a separate study[3] it has been shown that adding a range of different representations for depth information leads to increase in segmentation accuracy as opposed to more traditional segmentation algorithms such as GrabCut[6]. So we explore how a Recursive Neural Network (RNN) Algorithm[2] predicts the tree structure that accurately presents how different segments can be combined to form the original scene with better segmentation accuracy than traditional methods like CRFs[13]. Features are extracted for each of these small segments, where each segment could be part or a whole segment in the final result. These features encode information about color, depth and gradients from depth map and RGB images. These features are then mapped to

some semantic space in which the RNN algorithm operates.

2. Previous Work/Background

Most of the segmentation algorithms are completely based on just the RGB color information, texture information of the (grayscale) image or involved some form of external user input such as marking of bounding box. For example, unsupervised clustering algorithms such as k-means[7] and expectation maximization cluster pixels that have similar colors in the RGB or Lab Color Space. GrabCut[6] is an efficient method to segment foreground from background in an image based on color models and energy minimization, but requires user input.

In addition to just the color values, some of the commonly used feature descriptors to represent an image/segment include SIFT[8], Dense SIFT and HoG[9]. Fei-Fei and Perona[4] showed that for the problem of scene classification, densely sampled keypoints provide higher accuracies than finding keypoints using corner detectors. This result is due to the fact that keypoint detectors such as Harris or Difference of Gaussians tend to be concentrate at high texture regions. However, in indoor scenes such as bedroom and kitchen, even regions with low textures such as bed, wall, ceiling, floor, shelf, door, table etc. might provide critical cues for segmentation. Also, keypoint detectors tend to fire on different parts of an image, even with small changes in the image. Usually the SIFT[8] features are further processed using Bag Of Features (BoF)[10] that standardizes its dimensionality to that of the number of cluster centers. In the BoF, features are first extracted from the images and then a feature dictionary called visual vocabulary is formed, similar to a codebook dictionary used for document analysis. This dictionary is built using large amounts of training data, and clustering techniques such as k-means/k-means++[7] or mean shift[5]. These features extracted from the training images and the query image are then binned into a histogram. This process of creating the histogram removes spatial relationships between the features. Although some argue that the spatial relationships are important for scene classification, sometimes it allows for a simpler model. BoF[10] is invariant to spatial translations of features, and demonstrates decent performance in image segmentation tasks. We used the k-means algorithm and Bag of Words

model to represent our SIFT features as well, as described in Section 4.

3. Dataset

The dataset that we use is NYU depth dataset used by [3]. It is comprised of 2283 images of indoor scenes as obtained from video frames of the RGB and Depth cameras from the Microsoft Kinect. The dataset has a class-labeled subset that acts as ground truth images. They have been preprocessed to fill in missing depth labels and the raw RGB, depth and accelerometer data as provided by the Kinect.

4. Our Algorithm



Figure1: Flowchart of our algorithm

4.1 Relabeling the Dataset

The dataset as obtained above in its raw form, has approximately 1296 different labels many of which are similar to one another (Ex ‘window blinder’ and ‘window cover’), some are erroneous (Ex. ‘whieboard’ instead of ‘whiteboard’, ‘winsoe’ instead of ‘window’) and others not totally mutually exclusive of each other (‘wooden shelf’, ‘wooden shelve’, and ‘wooden shelves’). In addition, some of the classes such as ‘zinc’, ‘well’ have very rare appearances in images. The data hence needs to be preprocessed to reduce the number of classes, where the ones chosen appear in most of the dataset images. Hence, top 12 classes were selected and the entire data was processed again to push anything that does not belong to these top classes to an arbitrary background class (which was the 13th class). Figure2 below shows the result of reducing the number of different classes in an image.

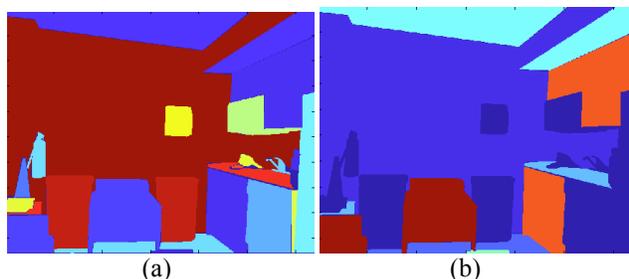


Figure2: (a) Original Image with all possible classes (21 different classes in this image) ;(b) Image with only top classes (8 different top classes in this image)

4.2 Over-Segmenting the Image

Highly segmented images are part of input to the learning algorithm. During the training phase the learning

parameters get trained to identify which segments from such images should be combined to form larger segments. This update of parameters is done based on maximizing the cost of ground truth trees while minimizing the cost of other possible tree structures for the scene image.

We used Edge Detection and Image Segmentation System (EDISON) algorithm [14] for producing these highly segmented images. We selected our parameters in a manner so as to restrict the number of segments that we get per image to an average of 100 while still getting high quality segments that attempt at distinguishing between different objects. Figure3 highlights the over segmentation result using EDISON segmentation[14], using the following mentioned parameters.



Figure3: (a) Original Image (b) Over segmented image using EDISON with parameters as : Minimum area = 600 pixels; Spatial Bandwidth = 10.0; Color Bandwidth = 4.0.

4.3 Feature Extraction:

The over-segmentation step is immediately followed by feature extraction for each of these small segments that will later be used by the RNN[2] algorithm to find similar segments. To train our RNN model we explored the following features:

A) Lab Color Histogram:

We first map the input image from RGB color space to Lab color space. We use Lab space because it has an expanded color gamut and most resembles the human perception of color as opposed to the RGB color space. Thereafter, for each segment in the post-EDISON segmented image, we compute the histogram in the 3-D Lab space with 10 bins in the ‘L’ direction and 12 bins in each of ‘a’ and ‘b’ directions. Lastly, we append the mean and standard deviation at the front of the histogram as features.

B) Scale Invariant Feature Transform (SIFT):

The traditional formulation of SIFT features involved two stages-keypoint detector and keypoint descriptor. However, unlike David Lowe’s SIFT[8] that uses difference of Gaussian to estimate the location of keypoints, SIFT used for scene classification samples keypoints in a regular grid fashion[4]. Regular grid sampling and a single scale is characteristic of the SIFT feature in scene classification community. Hence, we extract the SIFT features that are 128 dimensional independently from each of red, green and blue channels

of the image, resulting in an 384 (128x3) dimensional concatenated feature vector from the RGB image and another 384 dimensional feature vector from the Depth Image. The SIFT depth feature capture both large magnitude gradients caused by depth discontinuities, as well as small gradients that reveal surface orientation.

In the case of dense sampling of SIFT features, we observed that different segments in each image have different number of SIFT features within them, which resulted in the feature dimension not to be same for all the segments. So, we use the Bag of Words (BoW)[10] approach to fix the feature dimension to the number of cluster centers in the following two steps:

B1) Visual Vocabulary Generation – In order to generate the visual vocabulary, we found that performing k-Means[7] is very inefficient and extremely slow since the total number of features were very large viz. 2283x2394, where 2283 is the number of Images and 2394 are the number of SIFT[8] points per image, with each SIFT point being 384 dimensional for each of RGB and Depth Images. Since Hierarchical k-means[12] operates using hierarchical tree structure taking smaller steps in each level to find the top clusters, it is highly efficient and operates much faster than the flat k-means[7]. We use the VLFeat library[11] to perform Elkan’s Hierarchical k-means with a branching factor of 4 and create a total of 64 leaf clusters for RGB and Depth separately, totaling to 128 clusters. Before considering each feature for the clustering, we normalize it as follows:

$$\text{feature} = (\text{feature} - \min) / (\max - \min)$$

where min, max are the minimum and maximum values of the feature respectively.

B2) Bag of words model - Once we have the top 128 clusters we simply bin the SIFT features in each segment to a particular cluster center it is closest to using the Nearest Neighbor approach. In this technique, visual word frequency histogram uses the Vector Quantization (VQ) coding technique, where each keypoint maps to a unique dictionary codeword (i.e with a weight 1). This results in a 128 dimensional visual word frequency histogram for each segment.

(C) Histogram of Oriented Gradients (HoG):

HoG[9] features are very similar to SIFT[8] except that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. We use a window size of 3x3 and 9 number of bins in each cell.

4.4 Recursive Neural Network

RNN[2] is a framework that recursively combines similar segments to have the same class label, resulting in the final segmented image. Each image is preprocessed to produce an adjacency matrix that gives information about the neighboring segments for each segment. By its inherent property it is a symmetric matrix. Based on this,

for every combination of the smaller segments, RNN[2] computes a new score, new features and the class label for the possible combination. This score gives a measure of how well two adjacent segments can be combined to form a larger subunit. Based on this information, class labels adjacent segments are combined one by one to form the entire scene image.

RNN trains its parameters in four stages. The input to the algorithm is preprocessed dataset where for each segment of each image we find the correct mapping of features to semantic space and also the adjacency matrix as described above. The parameters for this phase are W^{sem} (nxm where n is the dimension of semantic feature (a_i) and m is the dimension of actual input feature).

$$a_i = f(W^{sem}F_i + b^{sem})$$

Based on this information, decisions would be made to combine smaller segments from the over-segmented image into larger segments. In each progressive step of the algorithm scores are calculated for each node added to the semantic tree constructed by combining leaf nodes that are essentially the super-pixels from the input image. At each step we only combine two super-pixels into a segment if the score of the parent node thus obtained is highest among all combinations at that level. When combining two sub-segments into a larger segment the new feature for the parent segment is found by using W (nx2n) matrix.

$$p_{(i,j)} = f(W[c_i; c_j] + b).$$

where p is the parent feature formed by combining child features c_i and c_j . Once parent segment feature is computed we calculate the score for that combination using W^{score} (1xn) as follows.

$$s_{(i,j)} = W^{score}p_{(i,j)}.$$

The weights decide whether two sub-units will be combined into a larger sub-unit based on how similar the two segments are in terms of class label or being parts of the same object in the image. To each RNN parent node a simple softmax layer is also added to predict its class label which is given as:

$$label_p = softmax(W^{label}p).$$

The algorithm thus is based on max-margin estimation and tries to maximize the score of the ground truth tree while minimizing the score of the incorrect trees for a given natural scene by penalizing it with respect to the score of the correct tree by a margin.

5. Results

We use 64 images in total (35 Training and 29 Testing) with a total running time of about 4-5 hours depending on

the feature dimension. Tabel-1 highlights the *pixel wise* accuracy on the NYU dataset [3].

Feature	Training (%)	Test (%)
Lab Color Histogram	69.11	44.11
RGB SIFT	71.54	52.07
RGB + Depth SIFT	77.36	38.47
RGB + Depth HoG	77.79	32.71
RGB HoG	--	39.56

Table1: Percentage accuracy for each of the features on the Training and Test Dataset.

Confusion Matrices

Since the background and wall occupy a large portion of most of the images, they tend to distort the confusion matrix. Hence following are the confusion matrices for the 11 classes excluding the background and wall.

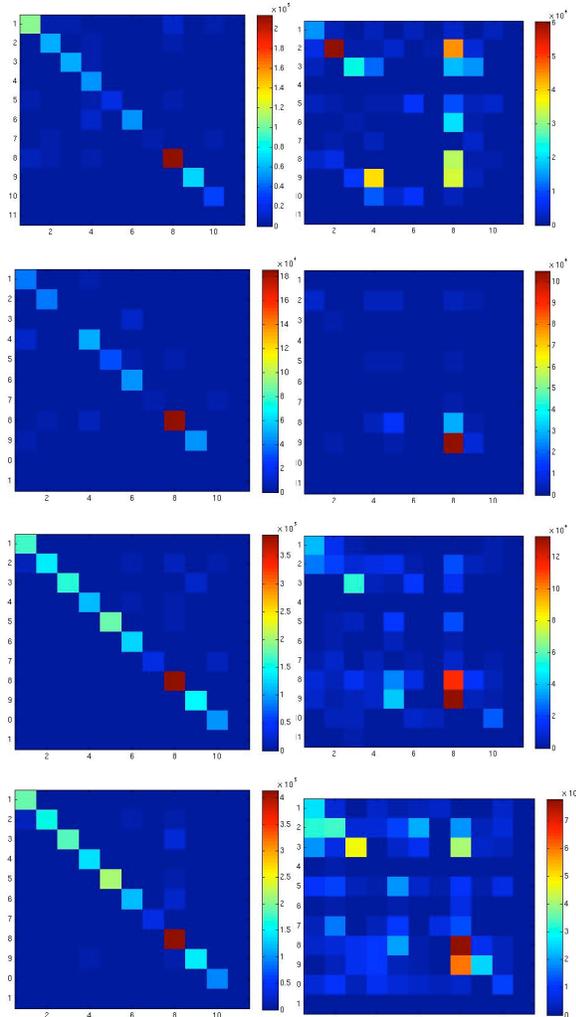


Figure-4 (1,1)-Lab Histogram (Training) ;(1,2)-Lab Histogram (Testing);(2,1) RGB SIFT (Training); (2,2) RGB SIFT (Testing) (3,1) RGB-D SIFT (Training);(3,2) RGB-D SIFT (Testing); (4,1) RGB-D HoG(Training) (4,2) RGB-D HoG (Testing);

All-Classes: Background, Wall, Table, Ceiling, Floor, Picture, Cabinet, Door, Chair, Sink, Book, Cloth, Faucet.

6. Analysis and Conclusion:

Some of the most interesting observations from this project are as follows:

(i) Indoor scenes like those in the provided dataset, have a large number of possible classes ~ 1296 , which we had to cut down to the top 13 classes to make it tractable. This resulted in images with a large number of background pixels or pixels belonging to most commonly occurring classes like wall or ceiling as opposed to others featuring in top 13 like faucet, cloth. This makes the model adversely biased towards these large subclasses and hence leads to large number of misclassifications. Perhaps subsampling from these large classes while training should improve the model and increase the accuracy.

(ii) RGB-SIFT seems to be performing the best amongst all the features considered, in terms of testing accuracy with a reasonably high training accuracy. We also believe that varying illumination and views across all images might be having a critical say in the final pixel wise accuracy.

(iii) The addition of depth SIFT to RGB-SIFT seems to be increasing the training accuracy by ~ 6 percentage points but reduces the testing accuracy. This is contrary to our initial belief that depth information always increases the percentage accuracy. However, this might as well be because we might not have chosen the best feature suitable for depth images. For the purposes of this project, we were limiting ourselves to SIFT and HoG on the depth images.

(iv) The testing accuracy for RGB-D HoG is very low (32.71%) but has an interesting high training accuracy. This might arise as HoG might be overfitting the training data, but is poorly generalized.

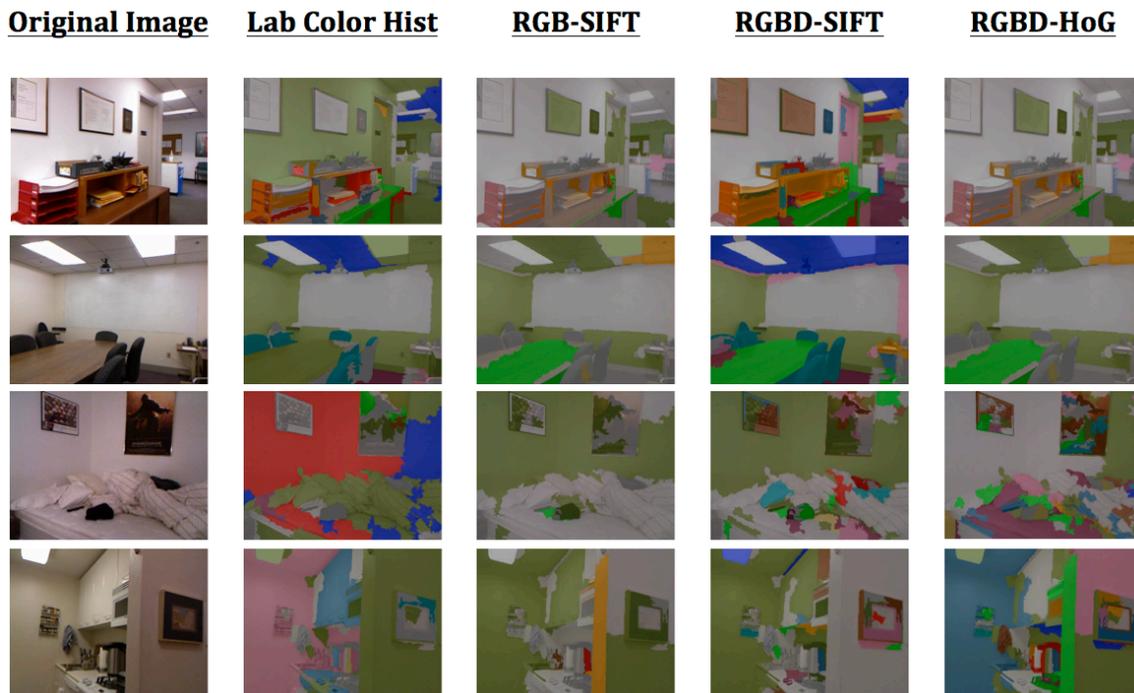
To conclude, in this paper, we primarily explore the performance of various feature descriptors and color histograms for the purposes of Segmentation using both Color and Depth images built on top of Recursive Neural Networks as the underlying framework.

7. Challenges

Due to high running time and limited computational resources (corn machines) and highly limited storage resources (needed for pre-computed SIFT features), we were unable to run on Corn clusters on full dataset, even after using /tmp/ directories to our benefit, as the administrators would clear our data when it exceeded a certain limit, which in most of the cases it did. Hence we restricted ourselves to smaller datasets to derive maximum learning conceptually, while leaving the computations on the full dataset for future undertakings.

8. References:

- [1] Gould, S., Fulton, R., and Koller, D. Decomposing a Scene into Geometric and Semantically Consistent Regions. In ICCV, 2009.
- [2] Socher R. et al. Parsing Natural Scenes and Natural Language using Recursive Neural Networks. In ICCV, 2011.
- [3] Silberman N., Fergus R. Indoor Scene Segmentation using a Structured Light Sensor. In ICCV, 2011.
- [4] Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In Proc. CVPR, 2005.
- [5] D. Comanicu, P. Meer: "Mean shift: A robust approach toward feature space analysis." *IEEE Trans. Pattern Anal. Machine Intell.*, 24, 603-619, May 2002.
- [6] C. Rother, V. Kolmogorov, and A. Blake, GrabCut: Interactive foreground extraction using iterated graph cuts, *ACM Trans. Graph.*, vol. 23, pp. 309–314, 2004.
- [7] MacQueen, J. B. (1967). "Some Methods for classification and Analysis of Multivariate Observations". 1. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281–297
- [8] Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157
- [9] Navneet Dalal and Bill Triggs "Histogram of Oriented Gradients for Human Detection". In Proceedings of IEEE Conference Computer Vision and Pattern Recognition , San Diego, USA, pages 886 - 893, June 2005.
- [10] Lewis, David (1998). "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval". Proceedings of ECML-98, 10th European Conference on Machine Learning. Chemnitz, DE: Springer Verlag, Heidelberg, DE. pp. 4–15.
- [11] @misc{vedaldi08vlfeat, Author = {A. Vedaldi and B. Fulkerson}, Title = {{VLFeat}: An Open and Portable Library of Computer Vision Algorithms}, Year = {2008}, Howpublished = {url{http://www.vlfeat.org/}}
- [12] Hamerly, G. and Elkan, C. (2002). "Alternatives to the k-means algorithm that find better clusterings". Proceedings of the eleventh international conference on Information and knowledge management (CIKM).
- [13] Lafferty, J., McCallum, A., Pereira, F. (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". Proc. 18th International Conf. on Machine Learning. Morgan Kaufmann. pp. 282–289.
- [14]. D. Comanicu, P. Meer: "Mean shift: A robust approach toward feature space analysis." *IEEE Trans. Pattern Anal. Machine Intell.*, 24, 603-619, May 2002.



8. Acknowledgement:

We would like to thank Prof. Andrew Ng for giving us the opportunity to work on this exciting and highly relevant problem. Thanks to Richard Socher, for his guidance throughout the project and for providing us with the codebase for the RNNs. This project was a great learning experience for both of us as we got to apply various machine learning techniques during the course of the project.