VISUAL SCENE UNDERSTANDING THROUGH SEMANTIC SEGMENTATION

by

Gautam Singh
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

_________________________________________  Dr. Jana Košecká, Dissertation Director

_________________________________________  Dr. Zoran Duric, Committee Member

_________________________________________  Dr. Huzefa Rangwala, Committee Member

_________________________________________  Dr. Anthony Stefanidis, Committee Member

_________________________________________  Dr. Sanjeev Setia, Department Chair

_________________________________________  Dr. Kenneth S. Ball, Dean, The Volgenau School
of Information Technology and Engineering

Date: ________________________________  Fall Semester 2014
George Mason University
Fairfax, VA
Visual Scene Understanding through Semantic Segmentation

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

By

Gautam Singh
Master of Science
George Mason University, 2013
Bachelor of Technology
International Institute of Information Technology, Hyderabad, 2007

Director: Dr. Jana Košecká, Professor
Department of Computer Science

Fall Semester 2014
George Mason University
Fairfax, VA
Dedication

I dedicate this dissertation to my parents.
Acknowledgments

I am grateful to many people for helping me get to this point. Many thanks to my advisor, Jana Košecká for all the help and support that she provided to help me achieve all that I’ve achieved during my Ph.D studies at George Mason. Despite her many other commitments, she has always been there whenever I needed her help for understanding computer vision, thinking about problems, writing papers or preparing talks. I’ve learned a tremendous amount from her during my time here and I’m eternally grateful and in debt for her guidance and patience along the way.

I would like to thank the members of my dissertation committee. Prof. Zoran Duric introduced me to computer vision and was the motivator who gave me the initial push to research in this great field of ours. Prof. Huzefa Rangwala provided me very useful insight which helped improve the quality of my work and the conclusions that I provide here. Prof. Anthony Stefanidis, with his questions, motivated me towards thinking about the greater picture in the context of my research work.

I also owe a lot to friends at GMU and elsewhere. I would like to thank all the members who have been associated with Jana’s lab for making my time here enjoyable - Stefano Scheggi, Ana Murillo, Xing Zhou, Damla Arifoglu, Alimoor Reza, Shenghui Zhou, George Georgakis, Arsalan Mousavian, Ali Bagheri Khaligh and Sanaz Rajbi. A special mention to George with whom I discussed virtually anything and everything ranging from European football to the powers of different entities in Indian mythology (hopefully he remembers the ten headed demon Ravan). Members of the Shehu Lab - Kevin Molloy, Brian Olson, Daniel Veltri and Irina Hashmi - made the office a fun place to be at. Kevin, particularly, taught me a lot about life in America. Friends outside Mason were like a family to me (and constantly fed me delicious food) - Nalini Vishnoi, Anish Mitra, Anindya Nath, Aveek Ganguly, Asha Rani, Anveshi Charuvaka, Shaeq Khan, Daniel Kling, Indrajit Das, Nakul Kumar and Ankit Mittal. Anish and Nalini joined Mason at the same time as me and all of us successfully graduated together.

Lastly but most importantly, my biggest thanks go to my parents and my brother. They have showered me constant love and been a source of inspiration and strength, my father with his quiet encouragement and advice about life in general, my mother for believing in me and being with me all this time and my brother with his own way of support. Without their love and support, this work would not have been possible.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>Abstract</td>
<td>xii</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Semantic Segmentation</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Contributions</td>
<td>9</td>
</tr>
<tr>
<td>2 Related Work</td>
<td>14</td>
</tr>
<tr>
<td>3 Topological Mapping and Scene Classification using Semantic Labeling</td>
<td>20</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Background</td>
<td>20</td>
</tr>
<tr>
<td>3.3 Approach</td>
<td>22</td>
</tr>
<tr>
<td>3.3.1 Semantic Label Descriptor</td>
<td>27</td>
</tr>
<tr>
<td>3.4 Experiments</td>
<td>28</td>
</tr>
<tr>
<td>3.4.1 Clustering Topology</td>
<td>28</td>
</tr>
<tr>
<td>3.4.2 Intersection Classification</td>
<td>35</td>
</tr>
<tr>
<td>3.5 Discussion</td>
<td>37</td>
</tr>
<tr>
<td>4 Non-parametric Semantic Segmentation</td>
<td>39</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>39</td>
</tr>
<tr>
<td>4.2 Background</td>
<td>39</td>
</tr>
<tr>
<td>4.3 Approach</td>
<td>40</td>
</tr>
<tr>
<td>4.3.1 Superpixels and features</td>
<td>41</td>
</tr>
<tr>
<td>4.3.2 Appearance Likelihood</td>
<td>42</td>
</tr>
<tr>
<td>4.3.3 Weighted k-NN</td>
<td>46</td>
</tr>
<tr>
<td>4.3.4 Semantic Contextual Retrieval</td>
<td>48</td>
</tr>
<tr>
<td>4.4 Experimental Results</td>
<td>50</td>
</tr>
<tr>
<td>4.4.1 Semantic segmentation results</td>
<td>50</td>
</tr>
<tr>
<td>4.4.2 Scene Categorization</td>
<td>61</td>
</tr>
<tr>
<td>4.4.3 Analysis of Retrieval Set</td>
<td>63</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>List of superpixel features.</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Per pixel and per class accuracy on 320 StreetView image dataset</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparison of category level accuracy on 320 StreetView image dataset</td>
<td>26</td>
</tr>
<tr>
<td>4.1</td>
<td>List of watershed superpixel features</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Global image features used for construction of retrieval set.</td>
<td>46</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of using different global features for building the retrieval set for the WKNN-MRF method. The features used are G - GIST, S - Dense SIFT histogram, R - RGB Color histogram.</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Semantic labeling performance on the SiftFlow dataset. AdaptNN [16] is based on SuperParsing [90] with the addition of a weighted $k$-NN classifier and superpixel label histograms for context.</td>
<td>55</td>
</tr>
<tr>
<td>4.5</td>
<td>Results for cross validation over the SiftFlow dataset.</td>
<td>56</td>
</tr>
<tr>
<td>4.6</td>
<td>Semantic segmentation accuracy for less frequent categories in the SiftFlow dataset.</td>
<td>56</td>
</tr>
<tr>
<td>4.7</td>
<td>Semantic labeling performance on the SUN09 dataset</td>
<td>57</td>
</tr>
<tr>
<td>4.8</td>
<td>Performance on the Google StreetView dataset</td>
<td>60</td>
</tr>
<tr>
<td>4.9</td>
<td>Performance on the Stanford background dataset</td>
<td>61</td>
</tr>
<tr>
<td>4.10</td>
<td>Scene categorization performance on subset of SUN-Attribute dataset. The set of ground truth attributes was modified to remove object attributes which were in common with the semantic categories of SUN09.</td>
<td>62</td>
</tr>
<tr>
<td>4.11</td>
<td>Ranking performance for different features. We use NDCG for different retrieval set sizes to measure the performance. Relevance between query and retrieved image computed as count of attributes shared between them.</td>
<td>66</td>
</tr>
<tr>
<td>5.1</td>
<td>Details for datasets used in our introspection experiments.</td>
<td>77</td>
</tr>
<tr>
<td>5.2</td>
<td>Results for familiar semantic category labeling on the SUN09 dataset using SiftFlow dataset as source for training instances. UKNN denotes uniform weight $k$-NN classifier while WKNN is the weighted $k$-NN. Strange KNN selects the least strange label to assign to a superpixel.</td>
<td>78</td>
</tr>
</tbody>
</table>
5.3 Semantic labeling results on SUN09 using Stanford background dataset. . . . . . . 78
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A scene which can be annotated with a wide variety of semantic descriptions. Image from Stanford Background dataset [26]</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>(a) In scene categorization, a single semantic category is associated with an image. (b) It is also possible to view a scene as a composition of sub-scenes i.e. scenes within a scene.</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Attributes are human designated names for different properties observable in a scene.</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Object detection looks at predicting the presence of objects in a scene with bounding boxes.</td>
<td>6</td>
</tr>
<tr>
<td>1.5</td>
<td>Semantic segmentation provides a semantic label for each pixel in the image.</td>
<td>7</td>
</tr>
<tr>
<td>1.6</td>
<td>Overview of Chapter 3</td>
<td>10</td>
</tr>
<tr>
<td>1.7</td>
<td>Overview of Chapter 4</td>
<td>11</td>
</tr>
<tr>
<td>1.8</td>
<td>Overview of Chapter 5</td>
<td>12</td>
</tr>
<tr>
<td>2.1</td>
<td>Examples for the choice of image sites used by approaches for semantic segmentation.</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Google Maps visualization of the StreetView dataset.</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>A panoramic piecewise perspective image used in our experiments; composed of four parts - left, front, right and back views</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>(a) Image from StreetView Dataset (b) Segmentation obtained using the graph based method of [20]. Superpixel boundaries marked by red color (c) Semantic labeling result for the given segmentation. Color code for labels in top row.</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Examples of semantic labeling of Streetview images. The top row shows a side view while the bottom row visualizes a frontal view. From left to right, (a) Input image (b) Ground truth annotation (c) Predicted labeling</td>
<td>27</td>
</tr>
<tr>
<td>3.5</td>
<td>Process of computation of descriptor for a location. Here $f_L, f_F, f_R, f_B$ are the semantic label descriptors for the left, front, right and back views respectively.</td>
<td>29</td>
</tr>
<tr>
<td>3.6</td>
<td>Visualization of the $k$-means clustering for the entire StreetView sequence. Different colors distinguish the cluster assignments for individual locations (best viewed in color).</td>
<td>30</td>
</tr>
</tbody>
</table>
3.7 Cluster visualization for the non-highway portion of the StreetView sequence. The highway section has been removed in this visualization. Note the black color assigned to the highway in Figure 3.6 is missing except for an area next to the riverfront which is similar to the highway and lacks buildings on either side.

3.8 Visualization of the average frontal view for each cluster (shown for 6 clusters). Different clusters capture semantic structures.

3.9 Cluster examples. Each column shows the average frontal image for a cluster in the top row followed by some sample frontal views from locations assigned to that cluster.

3.10 Cluster assignment match rate between revisited locations and the previously visited location nearest to them.

3.11 Top: Probability maps for each label occurring at a pixel at non-intersection side images. Bottom: Probability maps for each label occurring at a pixel at intersection side images. Red indicates a high probability while blue indicates a low probability.

3.12 Results for intersection recognition. Locations marked with green icons were predicted as intersections by our system.

4.1 An overview of our non-parametric approach for semantic labeling of images.

4.2 Example of watershed segmentation. (a) An input image (b) Watershed superpixels.

4.3 Impact of using an MRF for smoothing the labeling output. The data-term is the output based on using most likely label from the \( k \)-NN likelihood while the MRF uses data term with color intensity based smoothness. Color coding for semantic labels is provided at the top of the figure.


4.5 Impact of feature relevance on SiftFlow test set (best viewed in color). U and W are per-pixel accuracies for the UKNN-MRF and WKNN-MRF methods respectively. In example (b), higher weight for color component helped segment trees and in (c), SIFT feature channel has higher weight leading to better labeling of the building region.

4.6 Examples illustrating impact of refined retrieval set on SiftFlow test set (best viewed in color). W and R are per-pixel accuracies for WKNN-MRF and WAKNN-MRF respectively. See text for more details.
4.7 Comparison of our results on SiftFlow dataset against SuperParsing [90]. T and R are per-pixel accuracies for SuperParsing and WAKNN-MRF methods respectively. (a) shows utility of using our watershed superpixels which segmented the pole pixels. In (b) and (d), rarer classes like staircase, window, cars are labeled correctly while SuperParsing labels it as building.

4.8 We perform semantic labeling on unlabeled images of SUN-Attribute dataset. The semantic label descriptor from the resultant labeling is used to train scene category SVMs.

4.9 Example of images which received high scores from the scene category SVMs for (a) Beach (b) Dining Room (c) Street categories.

5.1 Example of the SLIC segmentation. (a) An input image (b) Corresponding superpixels.

5.2 Example uncertainty outputs (best viewed in color). (a) Query image (b) Predicted semantic segmentation (c) Strangeness based uncertainty. Darker intensity pixel implies higher uncertainty of belonging to the known semantic categories.

5.3 Comparison of uncertainty measures for confidence ranking in the SUN09 dataset using SiftFlow dataset. The y-axis denotes the percent of void pixels in the images present in a particular ranking subset (each of size 500 images) when using an uncertainty measure e.g. there are 93.1% void pixels in images ranked 1-500 using WKNN based strangeness. Higher percent of void pixels in lower ranks implies a better performance.

5.4 Evaluation of uncertainty measures on SUN09 using Stanford as source dataset. The evaluation procedure is the same as Figure 5.3.

5.5 Visualization of confidence ranking of SUN09 images in order of uncertainty based on WKNN strangeness with SiftFlow as source dataset. Under each image, we provide its rank.

5.6 Qualitative results on SUN09 using SiftFlow dataset. Entries N and W denote the image level uncertainty score when using the NEP and WKNN strangeness measures respectively. The confidence rank corresponding to the uncertainty measure is provided in parentheses.
The problem of visual scene understanding entails recognizing the semantic constituents of a scene and the complex interactions that occur between them. Development of algorithms for semantic segmentation, which requires the simultaneous segmentation of an image into regions and the classification of these regions into semantic categories, is at the heart of this problem. This dissertation presents methods that provide improvements to the state of the art in semantic segmentation of images and investigates the use of the obtained semantic segmentation output for related image retrieval and classification tasks. We present a method for non-parametric semantic segmentation of images which can effectively work on image datasets with a large number of categories. The method exploits query time feature channel relevance and also introduces the semantic label descriptor for improving the semantic segmentation output by retrieving images which share semantically similar spatial layouts. We further demonstrate how to associate accurate confidences with the resulting semantic segmentation through the use of the strangeness measure. We show how this measure can be applied for confidence ranking of unlabeled images and associate high uncertainty scores with images containing unfamiliar semantic categories. We then demonstrate the use of semantic segmentation output for additional tasks such as scene categorization, learning related semantic
concepts and content based image retrieval.
Chapter 1: Introduction

The fundamental problem of visual scene understanding can be viewed as the process of extracting semantic information from an image. As humans, when provided with a scene, we do not just look at its color or texture properties. We utilize our prior knowledge about the world to reason about the presence of objects in a scene, their spatial locations, the interactions between these scene constituents and the effect of the layout of the scene and any complex activities that maybe part of the scene. Humans have no difficulty with these tasks and can associate semantic information with the scene at different levels. This thesis seeks to replicate some of that functionality and develop representations of images which can efficiently enable association of semantic information with the image given some training examples of previously seen semantic concepts. The goal is to obtain a representation which can be used effectively in a wide variety of applications like content-based image retrieval [97], autonomous driving [25], robotic exploration of environments [4] and aid for the visually impaired [18].

There are a variety of ways to associate semantic information with a scene. As an example, consider Figure 1.1. It can be described as an outdoor scene, more specifically as an urban city. However, that information is not the only way to describe it. The scene has multiple objects in it like cars, bus and pedestrians. There are other categories like buildings, sky, road and crosswalk in the background. As human beings, we are also able to specify the exact spatial location of the different categories in the scene. The object categories can be classified as moving or non-moving entities. Knowing the spatial layout of the categories in the scene helps us further reason about open and closed spaces in the scene. As can be observed, many of these properties are linked to each other. Inferring all these properties remains a challenging task and different approaches have been proposed to gather semantic information about a scene.

To improve the understanding of scenes, a variety of tasks have been defined in the community and
they can typically be formulated as image labeling problems where the task is to assign labels to unobserved hidden random variables which correspond to different properties of a scene. These tasks differ in the kind of required human supervision and the effort needed to provide these human annotations.

One such task involves scene categorization which tries to predict a single semantic label associated with the image of a scene e.g. library, bedroom or kitchen. Notable progress has been achieved in scene categorization using bag of features models [10] and their spatially aware counterparts [48]. Recent work [103] has also looked at finer-level scene categorization where the scene can be seen as a composition of multiple sub-scenes which are defined as scenes within scenes e.g. restaurants, parking lots inside a street scene. An example of this is depicted in Figure 1.2. Scene categorization typically divides environments into functional and semantic groups e.g. beach vs kitchen. Recent work has also proposed models for associating visual attributes which are properties observable
in the image and have a human designated names like natural, man-made, wet, dry (example in Figure 1.3). These descriptions have served as cues for multiple tasks like scene recognition [69], object recognition [19] and describing unfamiliar objects [46, 47].

While scene categorization assigns a single semantic category to an image, a further understanding of the scene can be obtained by analyzing what objects are present and where are they located. For example, for an autonomous robot navigating in a building, it is useful to know if the current environment is a kitchen or a laboratory. However, it is also useful to detect the presence or absence of objects and their spatial locations since such information can provide cues for tasks like navigation and object tracking. Another extensively studied task is that of object detection where the goal is to identify the presence and location of object categories in an image e.g. Figure 1.4. Numerous object detection processing pipelines have been developed in the literature and they limit semantic information to different object categories and a single background category i.e. categories like building, tree, sea are treated as a single background category. Object detection approaches have been found to be effective for frequently occurring objects classes like faces [96], pedestrians [11] partly due to large number of training examples and relatively low intra-class variance.

Object detection localizes objects using bounding boxes which often over-approximates the boundaries of the object. A more richer representation is obtained by the task of semantic segmentation where the goal is to assign every pixel in an image with a semantic category like building, tree, road as shown in Figure 1.5 and it obtains a more refined localization of the object in comparison to object detection by knowing exactly which set of pixels correspond to an object.

1.1 Semantic Segmentation

Semantic segmentation (used interchangeably with semantic labeling and scene parsing in the rest of this document) has been an active area of research in computer vision for a substantial period [66, 67]. It has found applications in a wide variety of problems including content-based image search [98], object detection [21] and traffic understanding [17]. The training data for semantic segmentation requires images to be fully labeled i.e. each pixel needs to be labeled with a semantic
Figure 1.2: (a) In scene categorization, a single semantic category is associated with an image (b) It is also possible to view a scene as a composition of sub-scenes i.e. scenes within a scene.
category and providing human annotation is a laborious task [76].

Formally, semantic segmentation is defined as follows: Given an image, the desired output is a per pixel labeling. Semantic labeling can be formulated on different kind of image sites which correspond to the basic image element that will be labeled. The simplest possible sites to label in an image are its pixels [80]. Due to ambiguous information at the pixel level and for efficiency purposes, proposed approaches in the literature have used segments [24] obtained from bottom-up segmentation methods [20, 61, 78] or a group of segments [73]. Given an image, the output is the labeling \( L = (l_1, l_2, \ldots, l_S)^T \) with each site \( s_i \) in the image assigned a unique label, \( l_i \in \{1, 2, \ldots, L\} \) where \( L \) is the number of semantic categories and \( S \) is the total number of sites in the image. The posterior probability of a labeling \( L \) given the corresponding observed appearance feature vectors \( A = [a_1, a_2, \ldots, a_S] \) can be expressed as:

\[
P(L|A) = \frac{P(A|L) P(L)}{P(A)}. \tag{1.1}
\]
Figure 1.4: Object detection looks at predicting the presence of objects in a scene with bounding boxes.

The most probable or Maximum a Posteriori (MAP) labeling is defined as,

$$\arg\max_L P(L|A) = \arg\max_L P(A|L) P(L). \quad (1.2)$$

where $P(A|L)$ is the observation likelihood and $P(L)$ is the joint prior. In order to compute the observation likelihood, a Naive Bayes assumption of independence between appearance features given the labels is made and it yields

$$P(A|L) \approx \prod_{i=1}^{S} P(a_i|l_i). \quad (1.3)$$
Figure 1.5: Semantic segmentation provides a semantic label for each pixel in the image.

where \( P(a_i|l_i) \) models the correlation between the appearance and semantic label for image site \( s_i \). This is computed using classifiers like support vector machines [24], boosting classifiers [80] or \( k \)-nearest neighbor (\( k \)-NN) classifiers [82] on the features of the image sites. A way to model the joint prior \( P(L) \) is to define it as function over the labels of neighboring image sites as

\[
P(L) \approx \exp \left( \sum_{(i,j) \in E} g(i, j) \right),
\]

where the pairwise affinity function \( g(i, j) \) measures compatibility between neighbors \( s_i \) and \( s_j \) and the set \( E \) contains all neighboring pairs. The intuition behind this is to encourage neighboring sites to share the same labels. Given these formulations for the observation likelihood and the joint prior, one way to obtain the labeling in Eq. (1.2) is to rewrite it in log space and the image labeling
problem reduces to a minimization problem:

$$\text{argmin}_L \left( \sum_{i=1}^S E_{\text{data}}(s_i, l_i) + \lambda \sum_{(i,j) \in E} E_{\text{pair}}(l_i, l_j) \right),$$

(1.5)

where $E_{\text{data}}(s_i, l_i) = -\log P(a_i | l_i)$ is called the data term, $E_{\text{pair}}(l_i, l_j) = g(i, j)$ is the pairwise term defining compatibility between a pair of neighboring image sites and scalar $\lambda$ is the weight for this term. Under this formulation, the data term and pairwise term defined above denote unary and binary potential terms in a pairwise Markov Random Field (MRF) which is a probabilistic graphical model. The use of the MRF is motivated by the observation that while there is an exponential number of possible spatial arrangements of objects in an image, natural scenes are not a random collection of pixels and have a structure to them. Using a probabilistic graphical model like MRFs helps incorporate the conditional dependencies between the labels and solve for the whole labeling jointly. The inference in the MRF i.e. a search for a MAP assignment is performed using publicly available solvers [6, 7, 101]. Finally, the accuracy of a semantic labeling output $L$ is computed as:

$$\text{accuracy}(L) = \frac{\sum_{i=1}^S 1\{l_i = l_i^*\}}{S}$$

(1.6)

where $l_i^*$ is the ground truth label of pixel $x_i$ and $1\{\cdot\}$ is the indicator function.

There are online interfaces [76] available to use human annotators to provide the semantic labeling for an image. However, it is a very laborious task for humans leading to added expenses and with the rapidly increasing volume of images, there is an increasing need to develop methods for semantic segmentation of images. Through this dissertation, we present methods which improve upon state of art methods for the semantic labeling of images. However, the semantic labeling of the scene is not the only desired goal. We also use semantic labeling as a representation for images which can facilitate other tasks. The contributions of this dissertation are summarized below:
1.2 Contributions

The contributions of this thesis are improvements in the accuracy of the methods for semantic labeling and using it as a representation that facilitates different tasks like topological mapping of urban environments, street scene classification as intersections or non-intersections and content-based image search. Our work addresses different challenges and is summarized below:

- In the first part of this dissertation (Chapter 3), we adopt an existing approach to perform the semantic labeling of an image over a single bottom-up segmentation of the image and introduce the *semantic label descriptor* which summarizes the semantic labeling of a scene. Here, image superpixels [75] are described by discriminative features which have recently shown success for the computation of the geometric surface layout of a scene [31]. For computing the label likelihood for individual superpixels, discriminative boosting classifiers are utilized as they provide the advantage of automatically selecting the most informative features for a semantic category’s classification. Experiments with this representation are performed on street scenes and show comparative performance to other state of art semantic labeling methods. With the broader goal of scene understanding in mind, we compact the semantic labeling of a scene in the informative *semantic label descriptor* which is a spatial pyramid [48] based summary of the semantic labeling output. The semantic label descriptor is used to obtain a semantic model of the urban environment by inducing its topology and is also applicable to learning scene concepts like intersection versus non-intersections. Figure 1.6 provides an overview of our first contribution.

- In the second part (Chapter 4) of this dissertation, we present a non-parametric approach for semantic labeling of images based on a simple representation that provides improvements in accuracy in comparison to the state of the art approaches. In comparison, in Chapter 3, we presented an approach which trains a discriminative classifier for each semantic category. However, these approaches can be computationally infeasible as more categories and instances are added to increase the size of the training dataset. Recently, with the emergence of large
Figure 1.6: Overview of Chapter 3
databases, non-parametric models which are data driven have shown success in several computer vision applications. Our non-parametric method is formulated over small patches and these are described by three simple features which characterize their color, texture and location properties. The labeling is performed in a $k$-nearest neighbor ($k$-NN) framework which exploits a query time feature relevance computation method [15] for non-uniform weighing of the feature channels to improve the semantic labeling. Furthermore, we introduce a method for generating a retrieval set of images for nearest neighbor computation which includes not just visually similar but semantically similar images also. This is achieved through the use of the previously mentioned semantic label descriptor and obtains further improvements in semantic labeling performance. Figure 1.7 summarizes the presented approach. We follow the intuition that the semantic label descriptor implicitly divides scene categories and also present results for using it as a representation for scene categorization and image retrieval. The evaluation shows a correlation between the tasks of semantic segmentation with scene categorization and scene attribute annotation.
• The approaches presented in Chapters 3 and 4 operate under the closed world assumption of the dataset containing a fixed number of semantic categories. However, this is not true in a real world setting, as it is possible that unfamiliar categories are encountered in the unseen images. To help improve the understanding of unfamiliar semantic categories, we operate in a non-parametric framework and use a transductive confidence measure computed from the current semantic labeling of image regions of the unlabeled data (Chapter 5). For this purpose, we first analyze the unlabeled image data and predict image regions as known and unknown semantic categories. This is done through the computation of a $k$-NN based strangeness measure [71] which corresponds to the uncertainty of an instance belonging to a known semantic category with respect to the category’s labeled instances. The confidences computed for regions of unlabeled images are utilized to provide a confidence ranking of images which could be utilized to discover previously unseen categories. An overview of the presented approach is provided by Figure 1.8.

![Source Dataset of Labeled Instances](image)

<table>
<thead>
<tr>
<th>Input Image</th>
<th>Predicted Labeling</th>
<th>Labeling Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Image 1" /></td>
<td><img src="image2" alt="Predicted Labeling 1" /></td>
<td><img src="image3" alt="Confidence 1" /></td>
</tr>
<tr>
<td><img src="image4" alt="Image 2" /></td>
<td><img src="image5" alt="Predicted Labeling 2" /></td>
<td><img src="image6" alt="Confidence 2" /></td>
</tr>
<tr>
<td><img src="image7" alt="Image 3" /></td>
<td><img src="image8" alt="Predicted Labeling 3" /></td>
<td><img src="image9" alt="Confidence 3" /></td>
</tr>
<tr>
<td><img src="image10" alt="Image 4" /></td>
<td><img src="image11" alt="Predicted Labeling 4" /></td>
<td><img src="image12" alt="Confidence 4" /></td>
</tr>
<tr>
<td><img src="image13" alt="Image 5" /></td>
<td><img src="image14" alt="Predicted Labeling 5" /></td>
<td><img src="image15" alt="Confidence 5" /></td>
</tr>
<tr>
<td><img src="image16" alt="Image 6" /></td>
<td><img src="image17" alt="Predicted Labeling 6" /></td>
<td><img src="image18" alt="Confidence 6" /></td>
</tr>
<tr>
<td><img src="image19" alt="Image 7" /></td>
<td><img src="image20" alt="Predicted Labeling 7" /></td>
<td><img src="image21" alt="Confidence 7" /></td>
</tr>
<tr>
<td><img src="image22" alt="Image 8" /></td>
<td><img src="image23" alt="Predicted Labeling 8" /></td>
<td><img src="image24" alt="Confidence 8" /></td>
</tr>
<tr>
<td><img src="image25" alt="Image 9" /></td>
<td><img src="image26" alt="Predicted Labeling 9" /></td>
<td><img src="image27" alt="Confidence 9" /></td>
</tr>
<tr>
<td><img src="image28" alt="Image 10" /></td>
<td><img src="image29" alt="Predicted Labeling 10" /></td>
<td><img src="image30" alt="Confidence 10" /></td>
</tr>
</tbody>
</table>

Figure 1.8: Overview of Chapter 5
The rest of the dissertation is organized as follows. In Chapter 2, we briefly review research related to the work presented in this dissertation. Chapter 3 presents our system for semantic labeling of street scenes and using the resultant semantic labeling for topological mapping and semantic concept learning in urban environments. In Chapter 4, we transition to a data driven non-parametric approach which is shown to work effectively for semantic segmentation using small patches and simple features. As illustrated in Chapter 3, in Chapter 4, we present additional experiments that further show the utility of the resultant semantic labeling output for different tasks, namely, improving semantic labeling itself and for scene categorization and content based image retrieval. In Chapter 5, we move away from the closed world assumption of fixed number of categories and present results for associating confidences with semantic labeling which can help discover previously unseen categories through confidence ranking of an unlabeled dataset. The thesis is concluded in Chapter 6 with a discussion of potential future directions.
Chapter 2: Related Work

In this chapter, we briefly review literature related to the work presented in this dissertation. More detailed background references follow in the individual chapters. In this dissertation, we look at the problem of semantic segmentation and tasks that can be facilitated using semantic segmentation. In Chapter 1, we introduced the formulation for semantic segmentation in Eq. (1.1). We showed that the problem of finding the labeling output \( L = (l_1, l_2, \ldots, l_S)\top \) can be formulated as Maximum a Posteriori (MAP) estimate of \( P(L|A) \) where \( P(L|A) \) represents the posterior probability of a labeling \( L \) given the observed appearance features \( A = [a_1, a_2, \ldots, a_S] \) for the \( S \) sites of an image. The labeling problem for semantic segmentation requires the assignment of labels to multiple sites and these labels are typically conditionally dependent on each other e.g. two spatially connected sites in an image often share the same label. A way to deal with these dependencies which influence the MAP estimation is to incorporate them in a graphical model where each node of the graph corresponds to an individual image site and the set of edges \( E \) connecting the nodes model the conditional dependencies between the sites. This inducing of a graphical structure between image sites permits one to reformulate the desired posterior through an energy function

\[
\begin{align*}
\text{argmin}_{L} & \left( \sum_{i=1}^{S} E_{\text{data}}(s_i, l_i) + \lambda \sum_{(i,j) \in E} E_{\text{pair}}(l_i, l_j) + \ldots \right) \\
& \text{(2.1)}
\end{align*}
\]

where the first term \( E_{\text{data}}(s_i, l_i) \) called the data term defines the likelihood of site \( s_i \) taking the label \( l_i \) and the subsequent terms can be used to encode dependencies of different orders between image sites as defined by the edge structure of the graph. In Eq. (2.1), we have shown one example of such a term which is of order two and is a pairwise function defined over a pair of neighboring image sites.
This formulation as an energy function provides a generalized framework for the semantic segmentation problem where approaches proposed for semantic segmentation differ on the key ingredients in it. The key ingredients to solving semantic segmentation are the choice of image sites to label, the type of features used to characterize them, methods for computing local label evidence at the sites (these three compose the part which focuses on the data term in Eq. (2.1) above) and methods which may integrate this information with other terms e.g. the pairwise term above. We now briefly discuss these essential ingredients for semantic segmentation.

**Image Sites and Features** The simplest image site possible to label in an image are its pixels [80]. However, if for example, one considers an image of size 500x500, the number of image sites to label in the graph is 250,000 which greatly increases the size of the induced graph and limits the ability to model complex dependencies in an efficient manner which led to a different choice of image sites in the literature. The work of [79] looks at using regular sized patches to perform semantic segmentation. The use of such patches leads to a decrease in the number of nodes in the graph but the patches do not necessarily correspond to the boundaries of the semantic categories present in the scene affecting the semantic segmentation performance. Approaches [24, 90] have been proposed to use large image regions called superpixels which are generated from unsupervised bottom-up segmentation methods [1, 20]. These methods use a single oversegmentation of the image and characterize the large regions with a rich set of features and their use has displayed success on various benchmark datasets. An issue with such methods is that they rely on a single oversegmentation while each bottom-up segmentation method is known to have its share of pros and cons and it is highly unlikely that they will always generate a segmentation with accurate boundaries for the semantic labels present in the image. Another issue is that the use of large superpixels leads to poor performance on the object and less frequent categories in a dataset. Methods have been proposed which look at performing semantic labeling by using multiple segmentations or a hierarchical segmentation of an image. Some of these [42, 73, 106] formulate the labeling problem as an energy optimization problem similar to the function presented in Eq. (2.1). Other methods use a layered approach with a hierarchy of segmentations where the output provided by one level in the segmentation hierarchy provides context to the subsequent levels [3, 64].
In this dissertation, we explore the use of both large [1, 20] and small superpixels [102]. The large superpixels are shown to have competitive performance for semantic segmentation of a small set of categories (Chapter 3). In comparison to frequently occurring background categories, objects and less frequent labels have shown poor performance when using large superpixels [90] and in this dissertation, we show improvements for their semantic segmentation formulated on small superpixels characterized by simple features in Chapter 4.

The image sites used for semantic labeling are described by different features characterizing color, texture, perspective cues like the work of [31], geometry [62] and structure from motion [88]. In this dissertation, we begin by showing competitive performance for labeling a small set of semantic categories by using a rich set of features characterizing appearance and geometry of large superpixels. In order to compute a complex set of features, large image sites are often required but the results suffer from the difficulty of obtaining accurate object boundaries for the bottom-up segmentation methods. Later, we show results for an approach formulated on small superpixels characterized by only their color, orientation and location properties for semantic labeling of a large number of semantic categories.
**Data Term and Higher Order Terms**  The appearance and geometry based features computed to characterize the image sites are then used to train classifiers which can model the mapping between the image site features and their semantic labels. This is typically done by training discriminative models like support vector machines [3], boosting classifiers [80] or using non-parametric models like \(k\)-NN classifiers [90]. In this dissertation, we begin (Chapter 3) by training boosting classifiers for a small set of semantic labels commonly found in street scenes. There, we select large superpixels [20] as the image sites to label and their appearance and geometric properties are characterized to produce state of art results of street scene parsing.

With the emergence of large datasets of images, data driven non-parametric approaches have found applications to many problems. Such approaches avoid training altogether and have found applications in geo-localization [30], object and scene recognition [92] and semantic segmentation [54,90]. The commonly proposed approaches for non-parametric semantic segmentation are based on large superpixels characterized by a wide variety of features e.g. a superpixel descriptor of more than 1900 dimensions is computed in the works of [16, 90]. As discussed previously, the use of large superpixels often leads to poor performance on objects and the less frequent categories. Therefore, in the second contribution of this dissertation (Chapter 4), we present a non-parametric \(k\)-NN classifier for labeling of scenes but in comparison to [16, 90], we formulate semantic segmentation on small superpixels. Moreover, instead of computing a high dimensional descriptor for the superpixel, we use three simple features and show competitive performance for semantic segmentation with this simple representation. A significant amount of research has been invested in learning the relevance of different features to tasks at hand and these typically fall under the domain of distance metric learning. This relevance between labels and features can be learned at a global level [105], at a per category level [100] and at a per instance level [23, 59]. We follow this intuition of relating individual feature channels to labels and adopt a query time global feature relevance method [15] for image superpixels that better correlates features to semantic categories. Our experiments illustrate the efficacy of using our simple representation for obtaining state of art accuracy for semantic segmentation with the additional effect of notable improvements in the less frequent and *things* categories which are poorly labeled by large superpixel based methods like [90].
Works in the literature also look at incorporating additional cues like object co-occurrence statistics [43, 73] which can include spatial offset based priors [27], scene-object co-occurrence statistics [106] or combine the results for semantic segmentation with shape based object detectors [44]. Many of these approaches utilize these additional cues by modifying the form of the energy function e.g. using higher order terms in Eq. (2.1) and then define optimization procedures suitable to address the complexity provided by such additional information. In this dissertation, we also utilize additional information computed as a summary of the initial semantic segmentation output produced by our system. This is done by summarizing the output in the semantic label descriptor and is used to produce a refined semantic segmentation output (Chapter 4). In comparison to methods like [27, 43, 44, 73], we do not require modifying the energy function. Our energy function is formulated over small superpixels and our choice of a higher order term in the energy is limited to the use of a simple pairwise function.

**Related Tasks**

Semantic segmentation provides a pixel level parsing of the scene. Since the labeling output provides the spatial extent of objects in a scene, semantic segmentation has been used to facilitate other image understanding tasks. It has been used in holistic scene understanding models [51, 106] which in addition to semantic parsing can associate other information with a scene like its scene category. It has been applied to provide priors for depth estimation [53, 99]. The predicted presence/absence of objects in a scene has been utilized for content based image retrieval approaches [50, 98]. However, in contrast to the work presented in this dissertation, these methods are based on utilizing object detectors instead of semantic segmentation.

In Chapters 3 and 4, we show the use of semantic segmentation output as a representation which can be used to associate other levels of semantic information with a scene. This is done by introducing the semantic label descriptor to summarize the spatial layout of semantic categories in a scene. In Chapter 3, we show the ability to induce a topological map over an urban environment sequence that separates locations based on their similarities over this descriptor. Then, in both Chapters 3 and 4, using the semantic label descriptor, we show the ability to learn semantic concepts based on the semantic layout of the scene with applications to scene categorization and image retrieval. In
Chapter 3, we learn the concept of intersections in urban environments in a supervised fashion while the work in Chapter 4 shows the ability to learn semantic concepts like scene categories without any supervision provided for them.

**Introspection and Confidence Estimation**  
Traditional approaches for supervised learning follow a closed world assumption of fixed number of labels and concern themselves with optimizing the accuracy of familiar label classification on test instances. In practical settings, one would like to use the obtained models on unlabeled data which contain novel previously unseen semantic categories. It is therefore important to quantify the performance of the existing models on this unlabeled data. This quantification is frequently done by associating a confidence with the predicted labeling and it has received a lot of attention in active learning problems. Research has focused on proposing algorithms to compute these uncertainties [32, 36, 37, 91].

There has been little work on uncertainty computation for semantic segmentation, with the work of [94] focusing on an approach for efficient active learning for semantic segmentation instead of accurate confidence estimation and in Chapter 5, we present an approach for quantifying the uncertainty of semantic labels associated with the image regions obtained by semantic segmentation. We use a $k$-NN classifier and compute a measure called *strangeness* [71] based on nearest neighbor distances. The strangeness characterizes an instance’s uncertainty with respect to its own label and can help provide a confidence for a classification decision. We analyze unlabeled data and show how the strangeness measure is effective for discovering image regions from previously unseen categories and doing so with high uncertainty scores. We also show how to aggregate the region uncertainties to provide a confidence ranking for an unlabeled set of images with respect to the known set of labels.
Chapter 3: Topological Mapping and Scene Classification using Semantic Labeling

3.1 Introduction

In this chapter, we present an approach to obtain a richer representation of scenes from urban environments using additional semantics. For this purpose, in the first stage, we compute the semantic labeling of street scenes by considering five semantic categories which are commonly observed in urban environments - building, road, sky, cars and trees. Using the generated semantic layout, we propose the informative *semantic label descriptor* that summarizes the semantic layout and demonstrate its capability to induce a topology over the environment which helps captures its spatial semantics at different levels. We further show the ability to learn semantic concepts over scenes using the proposed feature illustrating its extension to another scene understanding task - street scene classification as intersections. Part of the work described in this chapter was published in [81].

3.2 Background

Endowing models of environments with a richer representation is an active area of research. These semantic models can significantly improve navigation and decision making capabilities of autonomous machines and enhance human machine interaction. Towards this end, a variety of metric, topological and hybrid models have been proposed in the robotics community. Topological maps represent environments as graphs and may differ on how the nodes and the connections between them are defined [9, 52, 60]. One of the reasons for endowing environment models with a topological map is to attain a discrete representation of continuous space and enable more efficient planning. The seminal work of Kuipers [41] presented the idea of a spatial semantic hierarchy where different
types of representations of the environment are related in a hierarchical manner. This has been instantiated in the context of indoors environments, where several methods for building topological maps from metric representations of space were explored [5]. While grouping neighboring places with similar appearance has been pursued for developing a topology [39], more recently, trends in mapping focus on endowing the environments in addition to geometry and topology, with additional semantics. The semantic labels have been either associated with individual locations [87], such as kitchen, corridor, printer room or individual image regions [70].

In our work, the semantic labels are associated with the individual pixels of an image. We utilize the semantic segmentation of scenes to induce a topological representation of the environment. Characterizing the semantic layout with an informative feature, we are able to gather evidence of different semantic concepts which can be used for more refined semantic labeling and scene classification as demonstrated by the ability to recognize intersections in streets.

**Overview of Semantic Segmentation**

With the development of methods for integration of object detection techniques, with various contextual cues and top down information as well as advancements in inference algorithms used to compute the optimal labeling, semantic segmentation has been an active research problem in computer vision. A majority of the current approaches are formulated using a graphical model over image sites which can capture the dependencies in local neighborhoods with conditional random fields [45] (CRFs) the most popular variant. In the work of TextonBoost [80], pixel level label evidence is gathered by combining appearance, shape and context based features. One of their main contributions was the introduction of texture-layout features which are simply counts of quantized texture descriptors at varying offsets from a pixel. These are combined with location based potentials and inference for the random field is performed using graph-cuts [6, 7]. These have been extended to use superpixels [26] which are computationally more efficient for inference purposes. Since a single unsupervised segmentation may be erroneous, researchers in [42] proposed the use of higher order CRFs in a hierarchical framework which allowed the integration of features at different
levels (pixels and superpixels) to help overcome such errors.

3.3 Approach

In the first stage of the approach, the goal is to compute the semantic layout of street scenes. For the street scene imagery, we use StreetView™ panoramas acquired by a 360° field of view Lady-Bug multi-camera system. The sequence consists of 12,000 panoramas acquired from a run in an urban environment and was previously used in loop detection experiments [65, 84]. A Google Maps visualization of the sequence is provided in Figure 3.1.

![Google Maps visualization of the StreetView dataset.](image)

A single panorama is obtained by warping the radially undistorted perspective images onto the sphere assuming one virtual optical center. The sphere is back-projected into a quadrangular prism to get a piecewise perspective panoramic image (see Figure 3.2). Our panorama is composed of four perspective images covering 360° horizontally and 127° vertically. The system includes a
top camera as well, but it is discarded as it does not provide much information. The panorama is represented by 4 views (front, left, back and right) each covering 90° horizontal FOV as seen in Figure 3.2. We discard the bottom part of all views which contains parts of the vehicle acquiring the panoramas.

![Figure 3.2: A panoramic piecewise perspective image used in our experiments; composed of four parts - left, front, right and back views](image)

The proposed approach for semantic labeling is based on a single bottom-up segmentation of the image where the superpixels are characterized with a variety of features including color, texture, location and perspective cues. The labeling is performed using boosting classifiers which automatically compute feature relevance in the trained classifier. The semantic labels we consider are five commonly occurring semantic categories in street scenes - ground, sky, building, car, tree. The proposed semantic segmentation approach is closely related to [31] and [90]. As the elementary image site which we will try to label, we choose the superpixels obtained by color based over segmentation scheme proposed in [20]. This segmentation algorithm typically generates large irregular regions of different sizes (see Figure 3.3a).

Since we are interested in learning the coarse semantic layout of the urban environment, we use both geometric as well as appearance features to capture the statistics of individual regions. The choice of features has been adopted from [31] where each superpixel is characterized by location and shape (position of the centroid, relative position, number of pixels and area in the image), color
Figure 3.3: (a) Image from StreetView Dataset (b) Segmentation obtained using the graph based method of [20]. Superpixel boundaries marked by red color (c) Semantic labeling result for the given segmentation. Color code for labels in top row.

(color histograms of rgb and hsv values and saturation value), texture (mean absolute response of the filter bank of 15 filters and histogram of maximum responses) and perspective cues computed from long linear segments and lines aligned with different vanishing points. We use the publicly available code provided by authors of [31] for computing these features. In addition to the above features, we endow each superpixel region with a histogram of SIFT descriptors computed densely at each image location and quantized into 100 clusters. The entire feature vector is of 194 dimensions. Table 3.1 summarizes the features computed for each image region.

In order to compute the label likelihood - $P(a_i|l_i)$ in Eq. (1.3) - we use discriminative boosting classifiers [77]. Within the boosting framework, we use decision trees as the weak learners since they automatically provide feature selection. We learn separate classifiers for each of the five classes and this is done in a one vs. all fashion. During testing, the separate classifiers are run on the individual feature vectors of the superpixels of an image and output confidence scores. The class with the maximum confidence score is assigned to be the superpixel’s label. In our implementation,
<table>
<thead>
<tr>
<th>Table 3.1: List of superpixel features.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Color</strong></td>
</tr>
<tr>
<td>RGB color mean</td>
</tr>
<tr>
<td>HSV values</td>
</tr>
<tr>
<td>Hue histogram (5 bins)</td>
</tr>
<tr>
<td>Saturation histogram (3 bins)</td>
</tr>
<tr>
<td><strong>Texture</strong></td>
</tr>
<tr>
<td>Histogram over LM filter [49] response</td>
</tr>
<tr>
<td>Mean absolute LM filter response</td>
</tr>
<tr>
<td><strong>Location and Shape</strong></td>
</tr>
<tr>
<td>Mean normalized x and y</td>
</tr>
<tr>
<td>Normalized x and y, 10th and 90th percentile</td>
</tr>
<tr>
<td>Normalized y w.r.t. estimated horizon</td>
</tr>
<tr>
<td>Segment location w.r.t. horizon</td>
</tr>
<tr>
<td>Normalized superpixel area</td>
</tr>
<tr>
<td><strong>Perspective Cues</strong></td>
</tr>
<tr>
<td>Lines: Number of line pixels</td>
</tr>
<tr>
<td>Lines: Percent of (nearly) parallel lines</td>
</tr>
<tr>
<td>Line Intersections: 8 bin histogram over orientations</td>
</tr>
<tr>
<td>Line Intersections: Percent right/above image center</td>
</tr>
<tr>
<td>Vanishing Points: Number of line pixels with vertical VP membership</td>
</tr>
<tr>
<td>Vanishing Points: Percent of line pixels with vertical VP membership</td>
</tr>
<tr>
<td>Vanishing Points: Number of line pixels with horizontal VP membership</td>
</tr>
<tr>
<td>Vanishing Points: Position of vertical VP w.r.t. segment center</td>
</tr>
<tr>
<td><strong>Gradient Histogram</strong></td>
</tr>
<tr>
<td>Dense SIFT [57] histogram with 100 clusters</td>
</tr>
</tbody>
</table>

each strong classifier has 15 decision trees and each of the decision trees has 6 nodes. An example of the obtained semantic layout is shown in Figure 3.3c.

We annotated a dataset of 320 side and 90 frontal views where each pixel of an image is assigned one of the five classes or *void* if it does not fall into any of the categories. Two separate models are learned, one using the dataset of side views and the other using the frontal views. This is because while the classes may have similar appearance across side or frontal views (e.g. trees are generally green in color), they may not necessarily share the same geometric properties in the two different views. As an example, in a frontal view the buildings are generally observed on the sides with the
ground/road in the middle while that is not the case in the side views where they appear in fronto-
parallel views. To evaluate the performance of the boosting classifier and compare it to state of the
art systems, we use the dataset of 320 side views. The classifier was trained using one half of the
dataset for training and the other half is used for testing.

Table 3.2: Per pixel and per class accuracy on 320 StreetView image dataset

<table>
<thead>
<tr>
<th>System</th>
<th>Global</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang-ECCV10 [108]</td>
<td>88.4</td>
<td>80.4</td>
</tr>
<tr>
<td>Zhang-CVPR11 [107]</td>
<td>93.2</td>
<td>73.1</td>
</tr>
<tr>
<td>Boosting</td>
<td>94.4</td>
<td>81</td>
</tr>
</tbody>
</table>

Table 3.3: Comparison of category level accuracy on 320 StreetView image dataset

<table>
<thead>
<tr>
<th>System</th>
<th>building</th>
<th>car</th>
<th>ground</th>
<th>sky</th>
<th>tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang-ECCV10 [108]</td>
<td>89.1</td>
<td>56.4</td>
<td>89.6</td>
<td>97.1</td>
<td>69.7</td>
</tr>
<tr>
<td>Zhang-CVPR11 [107]</td>
<td>95.3</td>
<td>40.5</td>
<td>96</td>
<td>92.5</td>
<td>41.4</td>
</tr>
<tr>
<td>Boosting</td>
<td>96.4</td>
<td>68.3</td>
<td>94.4</td>
<td>97.2</td>
<td>48.9</td>
</tr>
</tbody>
</table>

The results for the boosting classifier and its comparison to the approach of supervised label trans-
fer [108] and non-parametric scene parsing [107] methods on this dataset are provided in Table 3.2
and 3.3. It is observed that the boosting classifier outperforms the other state of the art systems
on this dataset and therefore, we use this classifier through all our experiments for the semantic
labeling of an image in this chapter. Some examples of the semantic labeling results can be found
in Figure 3.4. The illustrated examples include both frontal and side views.
3.3.1 Semantic Label Descriptor

The semantic labeling of an image provides a means of spatially aggregating the semantic information in the image. For example, the semantic labeling of a highway scene will typically be devoid of buildings while streets inside a city have high rise buildings. We propose to exploit the scene labeling to obtain a richer understanding of the urban environment. To summarize the semantic information in the labeled image, we introduce the semantic label descriptor. This descriptor captures the basic underlying structure of the image and can help divide images into sets of visually
and semantically similar images. This is done through encoding the spatial distribution of semantic categories in the image.

For a given image $I$, we divide $I$ into a uniform $n_k \times n_k$ grid. We compute the semantic labeling of the image and within each cell of the $n_k \times n_k$ grid, we compute the distribution for each of the five semantic categories using the number of individual pixels in that grid cell which have been assigned that class. This results in a five bin histogram for a single grid cell. The class distribution values for each cell are normalized so that they sum to one. The histograms for the $n_k^2$ grid cells are concatenated together resulting in a feature vector of length $5 \times n_k^2$. A high value for $n_k$ will capture the details of the layout more precisely but be prone to classification errors while a low value for $n_k$ would be less sensitive to errors in the labeling. In the experiments of this work, we use $n_k = 4$ resulting in a 80-dimensional semantic label descriptor.

### 3.4 Experiments

#### 3.4.1 Clustering Topology

The proposed semantic label descriptor is now used to cluster locations in the sequence based on their semantic similarity. While evaluating the performance of our boosting classifier (Table 3.2 and 3.3), we had used half of the 320 labeled images for training and the second half for testing. When computing the semantic layout of the entire sequence of 12,000 views, the classifier is trained using all 320 side views and run on the left and right side views of each location. The classifier trained using the 90 frontal views is run on the front and back view of each location. Locations for which the ground truth labels are available were excluded from the sequence labeling exercise. The resulting semantic layouts for the four views are then converted into the semantic label descriptor as described in Section 3.3.1. They are then concatenated together to form a location descriptor for each individual location. Since each individual semantic layout results in a 80 dimensional descriptor, the dimensionality of the location descriptor is 320 (using the four views). This process of generating a descriptor from its individual images that characterizes a location’s semantic structure.
Figure 3.5: Process of computation of descriptor for a location. Here $f_L, f_F, f_R, f_B$ are the semantic label descriptors for the left, front, right and back views respectively.

is visualized in Figure 3.5.

We then use the generated location descriptors for clustering the entire sequence. We perform k-means clustering and use cosine similarity between the descriptors instead of Euclidean distance. A color coded visualization of the clustering output over the StreetView sequence is shown in Figure 3.6.

One may note that the highway section in the bottom part of the sequence is assigned to a single cluster as the scene typically contains only sky and road with a few vehicles visible on the road. In Figure 3.7, we look in more detail at the non-highway portion of the sequence. It is interesting to note how cluster assignments often change at block intersections signifying a change in the semantic structure of the street scenes.
In Figure 3.8, we provide the average frontal view for a cluster when we set $k = 6$. It can be noted that the different clusters capture distinct semantic structures. For example, the top row has clusters for areas on highways or with buildings on only one side of the road. In the bottom row, there is a difference in the height of the buildings indicating that some areas have taller buildings than others. We provide sample frontal views assigned to a cluster characterizing such layouts in Figure 3.9. For example, column-(a) is a cluster corresponding to areas with buildings on one side of the road, column-(c) corresponds to highways, column-(d) has buildings on both sides of the street while column-(b) differs from column-(d) in the height of the buildings on the side.

**Clustering output at revisited locations**

We check the robustness of the clustering by analyzing the cluster assignments of revisited locations. Each location is provided with GPS coordinates specifying the latitude and longitude of that
Figure 3.7: Cluster visualization for the non-highway portion of the StreetView sequence. The highway section has been removed in this visualization. Note the black color assigned to the highway in Figure 3.6 is missing except for an area next to the river-front which is similar to the highway and lacks buildings on either side.
location. Using the GPS coordinates, the individual distance between all locations can be calculated. Any location which has a past location within a threshold distance of 10m is considered a revisited location. In order to avoid considering immediately preceding locations as revisits, we discard the previous 25 frames for a location so that views taken within short time of each other are not considered. Following this, we obtained a set of 3362 revisited locations. For each of these 3362 locations, we obtain its nearest neighbor location from the past. The cluster assignment for a revisited location and its closest past location are checked against each other. The matching rates for the set of revisited locations for different number of clusters is provided in Fig. 3.10. As can be seen, the cluster assignments maintain a matching rate of more than 75% for a large number of
Figure 3.9: Cluster examples. Each column shows the average frontal image for a cluster in the top row followed by some sample frontal views from locations assigned to that cluster.
clusters. This is crucial since, while there may be some changes at revisited location e.g. movement of pedestrians, the general semantic structure of the scene remains the same and hence, the semantic similarity ensures that revisited locations typically are assigned to the same cluster in the induced topology.

Figure 3.10: Cluster assignment match rate between revisited locations and the previously visited location nearest to them.
3.4.2 Intersection Classification

The semantic label descriptor introduced in the previous section was instrumental in grouping different urban regions together based on the presence and layout of different semantic categories in the scene. In this section we show how to infer additional semantic concepts from the attained image representation. In urban environments which can be described as networks of roads and intersections, it is useful to be able to classify a particular view as an intersection or not. The capability of detecting intersections often provides useful prior information of presence of additional semantic concepts, such as pedestrian crossings, stop lights, traffic lights etc. Intersections also correspond to locations where navigations decisions can be made and hence are of interest for automated driving systems.

Previous works explored scene classification using either global gist descriptor [68] or spatial pyramid matching [48] and considered more general scene categories like coast, mountain, forest, inside city and highway. In our setting we consider subordinate categories of intersection and non-intersection, which belong to urban scenes but vary in finer spatial semantic layout.

To recognize intersections, we compute an additional normalized histogram of the five semantic labels over the middle part of the image width for side views. This additional histogram is concatenated with the side view’s semantic label descriptor to yield a 85-dimensional descriptor and used to train a boosting classifier to classify the side views as intersections or non-intersections. This very simple approach is effective partly due to the 360 degree field of view and availability of the high quality of the semantic labels. The choice of integrating the label statistics from the middle of the side view is motivated by the distinguished appearance of intersections in inner city environments and also the fact that they typically appear at an angle from the main direction of travel. To visualize this intuition, we have computed for the side views (perpendicular to the direction of travel), for each pixel, the probability of a label occurring at that pixel at intersections and non-intersections in Figure 3.11. Based on this observation, an additional histogram is computed over 70% of the middle part of each side view.
Figure 3.11: Top: Probability maps for each label occurring at a pixel at non-intersection side images. Bottom: Probability maps for each label occurring at a pixel at intersection side images. Red indicates a high probability while blue indicates a low probability.
The 320 side views dataset was annotated for the intersection classifier experiments. Each of the 320 images is manually labeled as an intersection or a non-intersection. This resulted in a set of 250 non-intersection and 70 intersection views. The intersection descriptor is computed for all the 320 side views and another boosting classifier is trained using the resultant 320 descriptors. This boosting classifier has 5 decision trees and each of the decision trees has 4 nodes. This boosting classifier is now run on only the side views of the entire dataset. Locations which contributed images to the training of the intersection classifier were excluded from the test stage. If both the left and right side views of a location are classified as an intersection by the classifier, the location is categorized as an intersection. Otherwise the location is categorized as a non-intersection.

A visualization of this experiment can be seen in Figure 3.12. It can be observed that our intersection classifier successfully predicts intersection at many of the major intersections. A human annotator marked 79 unique areas of the sequence as intersections in the city. The intersection classifier correctly predicted an intersection for 63 of the 79 marked intersections for a recall rate of 79.7% indicating the effectiveness of our approach. A successful detection implies that at least two locations within 10m of an intersection were classified as an intersection by the classifier.

### 3.5 Discussion

In this chapter, we demonstrated an approach for semantic labeling of outdoor urban scenes formulated over superpixels obtained by a bottom-up segmentation method. We showed how the attained coarse semantic labels (building, sky, ground, trees, cars) and their spatial layout can be used to further understanding of street scenes. This was done by using the proposed semantic label descriptor for clustering locations based on their semantic similarity. An added utility of this representation was its ability to learn scene concepts as illustrated by successful recognition of intersections inside a city.
Figure 3.12: Results for intersection recognition. Locations marked with green icons were predicted as intersections by our system.
Chapter 4: Non-parametric Semantic Segmentation

4.1 Introduction

In this chapter, we present an approach for non-parametric semantic labeling of images. Non-parametric methods have shown notable success for semantic labeling recently and we formulate an approach over small patches characterized by simple features. The labeling is performed in a weighted $k$-NN framework where the weight computation has been adopted from [15]. We introduce a method which utilizes the semantic label descriptor (introduced in Chapter 3) computed from an initial semantic labeling of the image to obtain a refined retrieval set for nearest neighbor computation and an improved semantic segmentation output. The work presented in this chapter was previously published in [82, 83].

4.2 Background

With the increasing sizes of datasets and an increasing number of labels, the use of non-parametric approaches have shown notable progress in semantic segmentation [16,54,90,107]. They are appealing as they can utilize efficient approximate nearest neighbor search techniques e.g. $k$-d trees [63] and contextual cues. Context is often captured by a retrieval set of images visually similar to the query and methods developed for establishing matches between image regions (at pixel or superpixel level) for labeling the image. Using the method of SIFT Flow [55], pixel-wise correspondences are established between images for label transfer in [54]. Authors in [90] work at the superpixel-level and retrieve similar images using global image features which is followed by superpixel-level matching using local features and a Markov random field (MRF) to incorporate neighborhood context. The work of [90] was extended by [16] by training per superpixel per feature weights and also
by incorporating superpixel-level semantic context. A set of partially similar images is used in [107] by searching for matches for each region of the query image and then using the retrieval set for label transfer. A non-parametric method which avoids the construction of a retrieval set is [28] which instead addresses the problem of semantic labeling by building a graph of patch correspondences across image sets and transfers annotations to unlabeled images using the established correspondences. However the degree of the graph vertices is limited due to memory requirements for large datasets like SiftFlow [54].

Our method is closely related to the work of [16, 90] in that we also pursue a non-parametric approach, but differ in the choice of elementary regions, features, feature relevance learning and the method for computing the retrieval set for \( k \)-NN classification. The retrieval set is obtained in a feedback manner using a novel semantic label descriptor computed from the initial semantic segmentation. Similarly to [16], we follow the observation that a single global distance metric is often not sufficient for handling the large variations within a class and propose to compute weights for individual features channels. The weights in our case are computed at the test time to indicate the importance of color, gradient orientation vs location for individual superpixels. The computation of the feature relevance we adopt falls into a broad class of distance metric learning techniques which have been shown to be beneficial for many problems like image classification [22], object segmentation [59] and image annotation [29]. For a comprehensive survey on distance functions, we refer the reader to [74].

### 4.3 Approach

An overview of the proposed approach is shown in Figure 4.1. We follow the formulation presented in Eq. (1.1). In the presented approach, the image sites correspond to small watershed superpixels. The appearance likelihood - Eq. (1.3) - is computed using a \( k \)-NN based classifier (Section 4.3.2). The labeling by the \( k \)-NN classifier uses a weighted distance metric (Section 4.3.3) learned at query time. To further improve scene parsing, we utilize semantic context for refined retrieval (Section 4.3.4).
Figure 4.1: An overview of our non-parametric approach for semantic labeling of images.

4.3.1 Superpixels and features

For an image, we extract superpixels utilizing a segmentation method [102] where superpixel boundaries are obtained as watersheds on a negative absolute Laplacian image with LoG extremas as seeds. These blob-based superpixels are efficient to compute and naturally consistent with the boundaries with an example in Figure 4.2.

For each superpixel, we compute a 133-dimensional feature vector $a_i$ which captures statistics about color, gradient and superpixel location. The descriptors are summarized in Table 4.1. The SIFT descriptor for a superpixel is computed at a fixed scale and orientation using publicly available code [93].
Figure 4.2: Example of watershed segmentation. (a) An input image (b) Watershed superpixels.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Mean over pixels in Lab color space</td>
<td>3</td>
</tr>
<tr>
<td>Gradient Histogram</td>
<td>SIFT [57] at superpixel centroid</td>
<td>128</td>
</tr>
<tr>
<td>Location</td>
<td>Normalized x,y centroid coordinates</td>
<td>2</td>
</tr>
</tbody>
</table>

4.3.2 Appearance Likelihood

The individual label likelihood $P(a_i|l_j)$ for a superpixel $s_i$ is obtained using a $k$-NN method. Since a superpixel is uniquely represented by its feature vector, we use the symbols $s_i$ and $a_i$ interchangeably. For each class $l_j$ and every superpixel $s_i$ of the query image, we compute a label likelihood
score:
\[ L(a_i, l_j) = \frac{n(l_j, N_{ik})/n(l_j, G)}{n(l_j, N_{ik})/n(l_j, G)} \] (4.1)

where

- \( \bar{l}_j = L \setminus l_j \) is the set of all labels excluding \( l_j \);
- \( N_{ik} \) is a neighborhood around \( a_i \) with exactly \( k \) points in it;
- \( n(l_j, N_{ik}) \) is the number of superpixels of class \( l_j \) inside \( N_{ik} \);
- \( n(l_j, G) \) is the number of superpixels of class \( l_j \) in the retrieval set \( G \) (described later in Section 4.3.2).

We compute the normalized label likelihood score using the individual label likelihood:
\[ P(a_i | l_j) = \frac{L(a_i, l_j)}{\sum_{l_k=1}^{L} L(a_i, l_k)} \] (4.2)

A straightforward way to compute the neighborhood \( N_{ik} \) is to use the concatenated feature \( a_i \) (Section 4.3.1) and retrieve the \( k \) nearest points by computing distance to superpixels in \( G \). Such a retrieval can be efficiently performed by the use of approximate nearest neighbor methods like \( k \)-d trees [63].

**Inference**

The search for the optimum labeling is performed through inference in a pairwise MRF (Eq. (1.5)). We use the \( k \)-NN based likelihood computed in Eq. (4.2) for the data term. For the smoothness term, we chose a simple data driven prior which is based on color intensity difference. The pairwise
affinity function \( g(i, j) \) (Eq. (1.4)) is defined as

\[
g(i, j) = \begin{cases} 
1 - e, & \text{iff } l_i = l_j \\
\delta + e, & \text{otherwise},
\end{cases}
\] (4.3)

with \( e = \exp(-\|c_i - c_j\|^2 / 2\sigma^2) \), where \( c_i \) and \( c_j \) are 3-element vectors of mean colors expressed in the Lab color space for \( i \)-th and \( j \)-th superpixel, respectively. \( \sigma^2 \) and \( \delta \) are parameters set by the user. The above smoothness term is a combination of the Potts model (using constant penalty \( \delta \)) and a color difference based term \( e \). With this prior, we wish to keep same labels for neighboring superpixels but penalize an assignment of same labels if they differ in color. We perform the inference in the MRF, i.e. a search for a MAP assignment, using an efficient and fast publicly available \textsc{max-sum} solver [101].

In Figure 4.3, we provide examples of the impact of using the MRF for smoothing local superpixel likelihoods. The information at the local patch level can be noisy and using a MRF provides a more consistent labeling.

**Retrieval Set**

The computation of the appearance likelihood in Section 4.3.2 uses images from the training set. Instead of using the entire training set in the \( k \)-NN method, it is more useful to utilize a subset of images which are similar to the query image. For example, when trying to label a seaside image, it is more helpful if we search for the nearest neighbors in images of beaches and discard views from street scenes. We use overall scene appearance to find a smaller set of training images instead of using the entire training set. It helps discard images which are dissimilar to the query image and provides a scene-level context which can help improve the labeling performance. The retrieval subset will serve as the source of image annotations which will be used to label the query image. We compute three global image features for each image of the dataset as described in Table 4.2.
Figure 4.3: Impact of using an MRF for smoothing the labeling output. The data-term is the output based on using most likely label from the $k$-NN likelihood while the MRF uses data term with color intensity based smoothness. Color coding for semantic labels is provided at the top of the figure.
Table 4.2: Global image features used for construction of retrieval set.

<table>
<thead>
<tr>
<th>Global Image Feature</th>
<th>Description</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIST [68]</td>
<td>Response to filter at multiple scales</td>
<td>960</td>
</tr>
<tr>
<td>Dense SIFT [57] Histogram</td>
<td>3 level spatial pyramid with 200 words</td>
<td>4200</td>
</tr>
<tr>
<td>RGB Color Histogram</td>
<td>8 bins per color channel</td>
<td>24</td>
</tr>
</tbody>
</table>

All the images in the training set $T$ are ranked for each individual global image feature in ascending order of the Euclidean distance from the query image. We then add the individual feature ranks and re-rank the images of the training set based on the aggregate rank. Finally, we select a subset of images $T_g$ from the training set $T$ as the retrieval set. The superpixels of the images in set $T_g$ compose the set of training instances $G$ in Eq. (4.2).

This constitutes our baseline approach and is denoted UKNN-MRF in the experiments for the uniformly weighted $k$-NN. Its distinguishing characteristics are the use of small patch-like superpixels, simple features and approximate nearest neighbor methods for $k$-NN classification. We now describe in detail two of the main contributions in our work: a method for weighting feature channels non-uniformly and a strategy for refining the retrieval set.

### 4.3.3 Weighted $k$-NN

The baseline $k$-NN approach uses Euclidean distance to compute the neighborhood around the point. We propose to use a weighted $k$-NN method to compute the neighborhood of a query point. To compute a weighted distance between two superpixels $a_i$ and $a_j$, we split the feature vector into three feature channels of gradient orientation, color and location and first compute distances in individual feature spaces:

$$
d_{ij}^f = [d_{ij}^{color}, d_{ij}^{sift}, d_{ij}^{loc}]^\top
$$

(4.4)
where $d_{ij}^{color}, d_{ij}^{sift}, d_{ij}^{loc}$ are the Euclidean distances between the color, SIFT and location channels of the feature vectors $a_i$ and $a_j$ of the two superpixels respectively. We define a weighted distance between the two superpixels as

$$d_{ij}^{w} = w^T d_{ij}^{f}$$ (4.5)

where $w = [w_1, w_2, w_3] \in \mathbb{R}^3$ defines the weights for the individual feature distances. Using the weighted distance from Eq. (4.5), we can now obtain the neighborhood $N_{ik}$ around a superpixel by applying it to the feature distance vector $d_{ij}^{f}$ between $a_i$ and $a_j \in G$ to compute the label likelihood scores in Eq. (4.1). We now describe an approach to compute these weights.

**Weight computation**

With the varying nature of the retrieval set for individual query images, we use the locally adaptive metric approach of [15] for the weight computation. It is a query-based technique which uses a global metric to select neighbors for a test point which are then used to refine the feature weights. In our setting, the test points are the individual superpixels of the query image.

The goal is to estimate the relevance of a feature channel $i$ by evaluating its ability to predict class posterior probabilities locally at a query point. This is done by computing the expectation of the posterior $P(l_j|x)$ conditioned at a test point $x_0$ along feature channel $i$.

The ability of feature channel $i$ to predict $P(l_j|z)$ at $x_i = z_i$ is defined as

$$r_{i}(z) = \sum_{l_{j}=1}^{L} \frac{(P(l_j|z) - \bar{P}(l_j|x_i = z_i))^{2}}{\bar{P}(l_j|x_i = z_i)}$$ (4.6)

Intuitively, the smaller the difference between $P(l_j|z)$ and $\bar{P}(l_j|x_i = z_i)$, the more information feature channel $i$ provides for predicting the class posterior probabilities locally at $z$. For the query
point $x_0$, the relevance for feature $i$ can be computed by averaging the $r_i(z)$’s in its neighborhood

$$\bar{r}_i(x_0) = \frac{1}{|N(x_0)|} \sum_{z \in N(x_0)} r_i(z) \quad (4.7)$$

where $N(x_0)$ denotes a neighborhood centered at $x_0$ (using the current feature weights) with $K_0$ points in it.

The relative relevance can then be computed as

$$w_i(x_0) = \frac{\exp(cR_i(x_0))}{\sum_{p=1}^{m} \exp(cR_p(x_0))} \quad (4.8)$$

where $m$ is the number of individual feature channels (three in our case), $c$ is a parameter which determines the influence of $\bar{r}_i$ (at $c = 0$, all three feature channels have equal weights) and $R_i(x_0) = \max_{p=1}^{m} \{\bar{r}_p(x_0)\} - \bar{r}_i(x_0)$. The quantities $P(l_j|z)$ and $\bar{P}(l_j|x_i = z_i)$ in Eq. (4.6) are estimated by considering neighborhoods centered at $z$. In the experiments section, this method evaluates the effect of the weight learning on the final classification and is denoted WKNN-MRF for the weighted $k$-NN.

### 4.3.4 Semantic Contextual Retrieval

#### Semantic Retrieval Set

In Section 3.3.1, we presented the semantic label descriptor which was used to cluster semantically similar street scenes. In this section, we examine its effectiveness in the stage of improving the retrieval set of nearest neighbor images. While utilizing superpixel level context for a refined labeling output has been explored in the past [16, 64], here we focus on scene level context. The construction of the semantic label descriptor in this section differs slightly from the representation proposed in
Section 3.3.1, specifically in the granularity as a coarser grid is utilized.

Given an image which has been labeled using the WKNN-MRF method, we consider a spatial pyramid of \( n \) levels over the labeled image. At level \( i \) in the pyramid, we divide image \( I \) into a uniform grid of \( d \times d \) cells where \( d = 2^{i-1} \). Within each grid cell, we compute the distribution for each of the \( L \) classes using the number of individual pixels in that grid cell which have been assigned that class. This results in a \( L \)-bin histogram for a single grid cell. The class distribution values for each cell are normalized so that they sum to one. The histograms for all the grid cells in the spatial pyramid are concatenated together resulting in an image feature \( f_{\text{seman}} \) of length \( L \times C \) where \( C = \sum_{i=1}^{n} 4^{i-1} \) is the total number of cells in the spatial pyramid.

A higher value for \( n \) will capture the details of the layout more precisely but be more prone to classification errors while a lower value for \( n \) would be less sensitive to errors in the labeling but does not encode the spatial position of the semantic categories as well. This approach of computing a semantic label-based descriptor is similar to [33]. However our method differs in the fact that we use a spatial pyramid over the labeled image instead of a single grid to encode the semantic label information and we do not include additional appearance information in the descriptor, because it has already been captured through other global image features (Section 4.3.2). Our method also differs from [16] who compute a superpixel-level semantic context descriptor as a normalized label histogram of neighboring regions.

Global image features (GIST, color histograms and spatial pyramid over SIFT) were used to build retrieval set \( T_g \) in Section 4.3.2. We now use the semantic label descriptor \( f_{\text{seman}} \) introduced above to help us refine the quality of the retrieval set by exploiting the semantic context.

For each image \( I_k \) in the training set, we perform leave-one-out-classification on the image using the WKNN-MRF approach. Using the resultant semantic image labeling, we generate its corresponding semantic label descriptor \( f_{\text{seman}}^k \). Similarly, for the query view \( I_q \), we label it using WKNN-MRF method and compute the corresponding semantic label descriptor. We generate a new set of ranking for the images in training set \( T \) based on the distance between their semantic label descriptor and
that of the query image. The ranking is computed in an ascending order of the semantic label descriptor distances. We can now use this ranking in isolation or combine it with the rankings for other global image feature types as was done in Section 4.3.2 to obtain the semantic retrieval set $T_s$. Using the new retrieval set $T_s$, we once again perform semantic labeling on the image by the process described in Section 4.3.2. This method is denoted as WLKNN-MRF in our experimental results. The WLKNN refers to a weighted $k$-NN using a retrieval set built using the semantic label descriptor only. We also experiment with using the semantic layout descriptor with all the other three global image features for the building of the retrieval set and denote this method WAKNN-MRF.

In Figure 4.4, we illustrate two examples of the impact of semantic context on the retrieval set of a query. For each example, the top row shows the image, initial retrieval set and initial labeling. The second row shows the ground truth labeling, the refined retrieval set and the final labeling. In the first example, the query is a mountain image with mountain, street and highway scenes in the initial retrieval set. This leads to prediction of road on mountains. However, the presence of sky and mountains in the initial labeling leads to discarding of the urban images thereby refining the retrieval set and improving the labeling accuracy by correcting the incorrect road segment prediction. The second example has some forest and countryside images in the initial retrieval set. The semantic context from the labeling obtained using this initial retrieval set leads to more urban images in the refined retrieval set and an improved labeling. Note how the addition of urban images leads to a better labeling of less frequent categories like doors, windows and staircase.

### 4.4 Experimental Results

#### 4.4.1 Semantic segmentation results

For the validation of our approach, we perform its evaluation by testing and comparing it with several state-of-the-art techniques on different datasets: SiftFlow [54], SUN09 [8], Google Street View [104] and Stanford Background [26]. The evaluation criterion for the methods is the per pixel accuracy (percentage of pixels correctly labeled) and per class accuracy (the average of semantic
Figure 4.4: Examples of Retrieval Set Refinement with Semantic Context. Query Images from SiftFlow dataset.
category accuracies).

For the Google Street View dataset, we selected 10% of the training set as the size of our retrieval set. In case of the other two datasets, we used a retrieval set of 75 images. In all our experiments, we set number of nearest neighbors $k = 9$ in Eq. (4.1) and $\lambda = 0.4$ in Eq. (1.5). We obtained these parameters by selecting a small subset of the training images as a validation set. Computation of the feature channel weights required an average of four minutes for a single query image. To help speed up the computation of the weights, we approximate the neighborhood construction of [15] through $k$-d trees [63]. For the query view, we index the individual features from the retrieval set in a $k$-d tree, constructing one $k$-d tree per feature channel. The neighborhood computation is then approximated using the set union of the $k$-NN from different feature channels. We carry out 5 iterations of the weight computation step in Eq. (4.8) adaptively changing the nearest neighbors in the weighted neighborhood space. While this approximates the weight computation, it affected our performance only slightly (a maximum decrease of 0.4% in per-pixel accuracy across the datasets) and helped reduce the time for weight computation for an image to 20 seconds. For an image, feature computation, $k$-NN likelihood computation and MRF inference took 1 second, 13 seconds and 0.5 second respectively. When reporting the performance, we used the following variants of our approach:

- UKNN-MRF: uniform weights for the features with retrieval set obtained by global image features
- WKNN-MRF: computed weights for the features with retrieval set obtained by global image features
- WLKNN-MRF: computed weights with retrieval set built using the semantic layout descriptor only
- WAKNN-MRF: computed weights with retrieval set built using a union of semantic layout descriptor and the three other global image features.

52
SiftFlow

SiftFlow is a large dataset of 2688 images with 33 semantic categories. [54] split the dataset into 2488 training images and 200 test images.

In Table 4.3, we present a comparison of using different global features for building the initial retrieval set. The results are for the standard 200 images test split of the SiftFlow dataset provided by [54]. The details for the features used are discussed in Table 4.2. The results are shown for the single best performing feature, the pair of features that returns the best output and finally, the output when using all three. The use of all three features gives the best performance and they are used in all of the remaining experiments in this section.

<table>
<thead>
<tr>
<th>Features Used</th>
<th>Per-Pixel</th>
<th>Per-Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>75.6</td>
<td>27.9</td>
</tr>
<tr>
<td>G + S</td>
<td>76.4</td>
<td>28.6</td>
</tr>
<tr>
<td>G + S + R</td>
<td>77.2</td>
<td>29.3</td>
</tr>
</tbody>
</table>

Table 4.4 compares the performance of our different methods against the state of art results on this dataset. Our weighted $k$-NN MRF performs on a comparable level on the per-pixel accuracy with the top methods. However, it still trails [16] for the per-class accuracy. The effect of using weighted $k$-NN instead of an unweighted $k$-NN classifier is illustrated in Figure 4.5.

When we incorporate semantic context to obtain a refined retrieval set, our system achieves the best performance for both per-pixel and per-class accuracies. The categories which saw an increase of more than 10% in the labeling accuracy after the use of semantic context include *field, car, river, plant, sidewalk, bridge, door, crosswalk*. These are categories which do not occur very frequently but achieved improved labeling with the context. For example, identifying road and highways helps
Figure 4.5: Impact of feature relevance on SiftFlow test set (best viewed in color). U and W are per-pixel accuracies for the UKNN-MRF and WKNN-MRF methods respectively. In example (b), higher weight for color component helped segment trees and in (c), SIFT feature channel has higher weight leading to better labeling of the building region.
Table 4.4: Semantic labeling performance on the SiftFlow dataset. AdaptNN [16] is based on SuperParsing [90] with the addition of a weighted $k$-NN classifier and superpixel label histograms for context.

<table>
<thead>
<tr>
<th>System</th>
<th>Per-Pixel</th>
<th>Per-Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu et al. [54]</td>
<td>76.7</td>
<td>-</td>
</tr>
<tr>
<td>SuperParsing [90]</td>
<td>76.9</td>
<td>29.4</td>
</tr>
<tr>
<td>AdaptNN [16]</td>
<td>77.1</td>
<td>32.5</td>
</tr>
<tr>
<td>UKNN-MRF</td>
<td>75.6</td>
<td>27.9</td>
</tr>
<tr>
<td>WKNN-MRF</td>
<td>77.2</td>
<td>29.3</td>
</tr>
<tr>
<td>WLKNN-MRF</td>
<td>78.5</td>
<td>32.0</td>
</tr>
<tr>
<td>WAKNN-MRF</td>
<td>79.2</td>
<td>33.8</td>
</tr>
</tbody>
</table>

label cars, sidewalk and crosswalk. Figure 4.6 shows examples of impact on semantic labeling as a refined set is used. Figure 4.6(a)-(b) are instances of semantic context improving the labeling as trees and mountains are predicted in the initial labeling. Figure 4.6(c)-(d) show improvement in labeling of cars and trees in street environments as road and buildings are predicted. Figure 4.6(e) is an instance of degradation in performance as the initial prediction suggested higher percentage of buildings in the scene. Overall, on the 200 image test set of SiftFlow, 135 (67.5% of test set) images saw an improvement with the refined retrieval set while the remaining saw a decrease. The drop occurred most frequently in images from forest and open country scenes where improvement in segmenting field and plant regions affected labeling of the more frequent (green) mountains and trees.

We also consider the generalization capability of our approach to ensure that it does not overfit to the standard train and test splits which are used as a benchmark for this dataset. For this purpose, we randomly generate 5 different splits composed of 2488 training images and 200 test images and evaluate our methods on these splits. The results for this cross validation step are presented in Table 4.5.

The results for the cross validation are similar to what was observed on the standard split of the dataset. The use of feature relevance for a weighted $k$-NN improves global accuracy of labeling by
Table 4.5: Results for cross validation over the SiftFlow dataset.

<table>
<thead>
<tr>
<th>Features Used</th>
<th>Per-Pixel</th>
<th>Per-Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKNN-MRF</td>
<td>75.3 ± 0.5</td>
<td>27.7 ± 0.3</td>
</tr>
<tr>
<td>WKNN-MRF</td>
<td>76.8 ± 0.3</td>
<td>29.2 ± 0.2</td>
</tr>
<tr>
<td>WLKNN-MRF</td>
<td>78.1 ± 0.3</td>
<td>31.8 ± 0.3</td>
</tr>
<tr>
<td>WAKNN-MRF</td>
<td>78.9 ± 0.2</td>
<td>33.3 ± 0.4</td>
</tr>
</tbody>
</table>

more than 1% and the use of semantic context improves both per-pixel and per-class accuracy by more than 2%.

A qualitative comparison of results using our approach to that of SuperParsing [90] is provided in Figure 4.7. The small watershed superpixels help label rarer labels like pole, window, staircase which are typically incorrectly classified by the system of [90] due to the bottom-up segmentation used by them. To summarize, our method performed better than their approach on 56 images by at least 5% (using the per-pixel accuracy metric) while it was worse on 34 images. On the remaining 110 test images, the two methods were comparatively equal.

Table 4.6: Semantic segmentation accuracy for less frequent categories in the SiftFlow dataset.

<table>
<thead>
<tr>
<th>Category</th>
<th>SuperParsing [90]</th>
<th>WAKNN-MRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>River</td>
<td>13</td>
<td>39.1</td>
</tr>
<tr>
<td>Plant</td>
<td>2</td>
<td>12.4</td>
</tr>
<tr>
<td>Grass</td>
<td>25</td>
<td>52.1</td>
</tr>
<tr>
<td>Sidewalk</td>
<td>31</td>
<td>37.3</td>
</tr>
<tr>
<td>Bridge</td>
<td>6</td>
<td>23</td>
</tr>
<tr>
<td>Door</td>
<td>17</td>
<td>35.1</td>
</tr>
<tr>
<td>Fence</td>
<td>15.7</td>
<td>25.9</td>
</tr>
<tr>
<td>Staircase</td>
<td>10</td>
<td>46.1</td>
</tr>
<tr>
<td>Sign</td>
<td>0</td>
<td>5.1</td>
</tr>
</tbody>
</table>

Table 4.6 provides the output of our method on the less frequent and things categories which show
an improvement in comparison to the output of SuperParsing [90] which uses large features characterized by more than 20 feature channels. This highlights the utility of using both the watershed superpixels as image sites to label and the semantic label descriptor which provides context for improved labeling of the less frequent labels.

SUN09

SUN09 [8] dataset has fully labeled per-pixel ground truth for a set of 107 semantic categories. In the experiments, the dataset was split into 4352 training images and 4310 test images. Table 4.7 reports the performance of our method on this dataset. Using the semantic context helped obtain an improvement of 3.6% compared to the WKNN-MRF method. In comparison to [89], we perform better on per-pixel accuracy but trail on per-class accuracy. It was observed that the per-pixel labeling accuracy of outdoor scenes was more than 11% better than indoor scenes highlighting the challenge of labeling indoor views using a retrieval set based approach.

Table 4.7: Semantic labeling performance on the SUN09 dataset

<table>
<thead>
<tr>
<th>System</th>
<th>Per-Pixel</th>
<th>Per-Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choi et al. [8]</td>
<td>33.0</td>
<td>10.6</td>
</tr>
<tr>
<td>ALE [43]</td>
<td>53.6</td>
<td>17.4</td>
</tr>
<tr>
<td>CascALE Expert [89]</td>
<td>49.3</td>
<td>16.7</td>
</tr>
<tr>
<td>CascALE Sharing [89]</td>
<td>52.8</td>
<td>15.2</td>
</tr>
<tr>
<td>WKNN-MRF</td>
<td>49.5</td>
<td>8.7</td>
</tr>
<tr>
<td>WAKNN-MRF</td>
<td>53.1</td>
<td>12.1</td>
</tr>
</tbody>
</table>

Similar to Table 4.5, we perform cross validation on this dataset. The per-pixel and per-class accuracy are 49.1±0.3 (8.4±0.2) and 52.7±0.3 (11.8±0.4) for the WKNN-MRF and WAKNN-MRF methods respectively highlighting the utility of semantic context.
Figure 4.6: Examples illustrating impact of refined retrieval set on SiftFlow test set (best viewed in color). W and R are per-pixel accuracies for WKNN-MRF and WAKNN-MRF respectively. See text for more details.
Figure 4.7: Comparison of our results on SiftFlow dataset against SuperParsing [90]. T and R are per-pixel accuracies for SuperParsing and WAKNN-MRF methods respectively. (a) shows utility of using our watershed superpixels which segmented the pole pixels. In (b) and (d), rarer classes like staircase, window, cars are labeled correctly while SuperParsing labels it as building.
**Google-StreetView**

The Google StreetView dataset (Section 3.3) contains 320 images selected from a set of 10,000 images captured in Pittsburgh. The dataset has five semantic categories: building, ground, tree, sky and car. The labeled data is equally split into a train and test set of 160 images. Table 4.8 contains the performance of our WAKNN-MRF approach against other methods. In comparison to the other methods, our performance was in the top-two for the per-pixel accuracy and for two semantic categories.

<table>
<thead>
<tr>
<th></th>
<th>Per-Pixel</th>
<th>Per-Class</th>
<th>Building</th>
<th>Car</th>
<th>Ground</th>
<th>Sky</th>
<th>Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang et al. [108]</td>
<td>88.4</td>
<td>80.4</td>
<td>89.1</td>
<td>56.4</td>
<td>89.6</td>
<td>97.1</td>
<td>69.7</td>
</tr>
<tr>
<td>Zhang et al. [107]</td>
<td>93.2</td>
<td>73.1</td>
<td>95.3</td>
<td>40.5</td>
<td>96.0</td>
<td>92.5</td>
<td>41.4</td>
</tr>
<tr>
<td>Singh et al. [81]</td>
<td>94.4</td>
<td>81.0</td>
<td>96.4</td>
<td>68.3</td>
<td>94.4</td>
<td>97.2</td>
<td>48.9</td>
</tr>
<tr>
<td>WAKNN-MRF</td>
<td>93.7</td>
<td>76.4</td>
<td>94.0</td>
<td>66.2</td>
<td>96.7</td>
<td>93.9</td>
<td>31.0</td>
</tr>
</tbody>
</table>

**Stanford-Background**

This dataset contains 715 images with two separate label sets; semantic and geometric. We conducted our experiments for predicting the semantic category only. The semantic classes include seven background classes and a generic foreground class. Experiments on this dataset are conducted over five different splits with each split containing 572 training images and 143 test images. Table 4.9 summarizes our performance on this dataset. The use of semantic context leads to an improvement of only 0.5% and our results trail other state of the art methods. The lack of significant improvement with the use of semantic context here can be explained by the nature of the dataset as more than 90% of the images contain 4 or more of the 8 semantic categories.

60
Table 4.9: Performance on the Stanford background dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Per-Pixel</th>
<th>Per-Class</th>
<th>Sky</th>
<th>Tree</th>
<th>Road</th>
<th>Grass</th>
<th>Water</th>
<th>Building</th>
<th>Mountain</th>
<th>Foreground</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel CRF [26]</td>
<td>74.3</td>
<td>66.6</td>
<td>93.9</td>
<td>67.1</td>
<td>90.3</td>
<td>83.3</td>
<td>55.4</td>
<td>71.4</td>
<td>9.3</td>
<td>62.2</td>
</tr>
<tr>
<td>Region Energy [26]</td>
<td>76.4</td>
<td>65.5</td>
<td>92.6</td>
<td>61.4</td>
<td>89.6</td>
<td>82.4</td>
<td>47.9</td>
<td>82.4</td>
<td>13.8</td>
<td>53.7</td>
</tr>
<tr>
<td>Leaf Level [64]</td>
<td>72.8</td>
<td>58.0</td>
<td>89.7</td>
<td>58.3</td>
<td>85.8</td>
<td>69.8</td>
<td>15.8</td>
<td>78.1</td>
<td>1.5</td>
<td>64.9</td>
</tr>
<tr>
<td>Hierarchy [64]</td>
<td>76.9</td>
<td>66.2</td>
<td>91.6</td>
<td>66.3</td>
<td>86.7</td>
<td>83.0</td>
<td>59.8</td>
<td>78.4</td>
<td>5.0</td>
<td>63.5</td>
</tr>
<tr>
<td>WKNN-MRF</td>
<td>73.6</td>
<td>61.2</td>
<td>92.5</td>
<td>67.1</td>
<td>91.5</td>
<td>74.5</td>
<td>45</td>
<td>77.6</td>
<td>2.8</td>
<td>38.7</td>
</tr>
<tr>
<td>WAKNN-MRF</td>
<td>74.1</td>
<td>62.2</td>
<td>92.2</td>
<td>65.0</td>
<td>91.0</td>
<td>74.6</td>
<td>45.8</td>
<td>78.2</td>
<td>6.7</td>
<td>44.1</td>
</tr>
</tbody>
</table>

4.4.2 Scene Categorization

In Chapter 3, we observed the efficacy of the semantic label descriptor for predicting the scene category of an image in a binary classification setting (intersection or non-intersection). We further investigate its utility for this task on a larger scene categorization dataset.

Our experiments for further evaluating the semantic label descriptor for scene classification are performed on a subset of the SUN-Attribute dataset [69]. The attribute dataset has 717 scene categories with 20 images for each category and we selected a subset of scene categories from this dataset. We choose all scene categories whose corresponding occurrences in the SUN09 dataset exceeded 20. Using this criteria, we obtained a set of 40 scene categories satisfying this threshold. Some example categories in this 40 category dataset include beach, casino indoor, hospital, kitchen, skyscraper.

We use the WKNN-MRF method to label the images of these categories in the attribute dataset where the training set is composed of the images corresponding to only these scene categories in the SUN09 dataset. Using the constraints on composing the datasets for evaluation, the subset from SUN09 dataset is composed of 1584 images while the subset from SUN-Attribute dataset has 800 images (40 scene categories, 20 images in each category).

We compute semantic label descriptors from the resultant labeling and use them to train linear
scene categorization SVMs as shown in Figure 4.8. The SVMs are trained by selecting half of the 20 images for a scene category with the second half used for evaluating the scene categorization classifiers.

Figure 4.8: We perform semantic labeling on unlabeled images of SUN-Attribute dataset. The semantic label descriptor from the resultant labeling is used to train scene category SVMs.

The results for the categorization are displayed in Table 4.10.

Table 4.10: Scene categorization performance on subset of SUN-Attribute dataset. The set of ground truth attributes was modified to remove object attributes which were in common with the semantic categories of SUN09.

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic label descriptor</td>
<td>30.25</td>
</tr>
<tr>
<td>Ground truth attributes</td>
<td>50.75</td>
</tr>
<tr>
<td>Combination</td>
<td>62.5</td>
</tr>
</tbody>
</table>

It is observed that when used in isolation, the semantic label descriptor does not perform as well as the ground truth attribute features. However, when we combine them with the attributes, we obtain an improvement of more than 11% in the categorization performance. This illustrates the utility of the proposed semantic label descriptor which is based on obtaining object level semantic information from a scene. Some images which received high scores from the scene category SVMs
using our representation are visualized in Figure 4.9.

4.4.3 Analysis of Retrieval Set

In Section 4.3.4, we presented an approach which utilizes the semantic labeling output of a scene to retrieve scenes which share similar layouts and then used the refined retrieval set to obtain an improved labeling output. This was validated by the state of art results presented in Section 4.4.1. Furthermore, in Section 4.4.2, we presented results for using the semantic label descriptor as a feature for scene categorization. The results showed an improvement when this representation was used in conjunction with scene attributes. It suggests that scene share similarities at multiple levels - semantic layouts, scene categories and attribute space. We are now interested in further analyzing the retrieval set and how relevant are the images that are retrieved when using visual and semantic similarity.

Image Ranking Analysis

For our experiments, we consider the subset of the SUN-Attribute [69] dataset that we used for scene categorization experiments in Section 4.4.2. The images in this subset were selected on the basis of sharing scene categories with the SUN09 [8] dataset. Using the selection criteria, we obtain a dataset of 800 images. For each image in the 800 image dataset, we compute global image features (previously used for retrieval set construction and listed in Table 4.2) and perform semantic labeling (Section 4.3) using the SUN09 dataset as the training dataset. From the resultant semantic labeling, we compute the spatial pyramid based semantic labeling descriptor.

We partition the dataset into two splits with 80% serving as a reference set and the remaining 20% as a test split. For each image in the test split, we perform a nearest neighbor retrieval over the reference set using visual features or visual features in conjunction with the semantic label descriptor where the labeling is performed using a subset of images from SUN09 (as was done in Section 4.4.2).

The images of the 800 image dataset are annotated at multiple levels - scene categories (one of
Figure 4.9: Example of images which received high scores from the scene category SVMs for (a) Beach (b) Dining Room (c) Street categories.
40 for our SUN-Attribute subset) and scene attributes. In the work of [69], scenes were described with the presence or absence of scene attributes which fall under four major categories (a) materials (e.g. cement, vegetation) (b) surface properties (e.g. rusty) (c) functions (e.g. cooking, playing) and (d) spatial envelope properties (e.g. enclosed, symmetric). In total, there are 102 attributes which include 38 material, 11 surface property, 36 function, and 17 spatial envelope attributes. This level of annotation provides an opportunity to compare the different features for the computed image ranking and their relevance to a query image.

For a given query, we define $\text{rel}_i$ as the relevance between the query and the $i$-th ranked image in the query’s retrieval set. Given the relevance scores between the query and retrieved images, we use the Normalized Discounted Cumulative Gain (NDCG) metric [35] to measure the performance of an image ranking approach. NDCG measures the performance of image ranking based on the relevance of image retrieved. It varies from 0 to 1 where a value of 1 represents the ideal ranking of the reference set images given a query.

The NDCG computation required computing the DCG first. The Discounted Cumulative Gain (DCG) at a rank position $p$ is defined as

$$ DCG_p = \sum_{i=1}^{p} \frac{\text{rel}_i}{\log(i+1)} $$ (4.9)

This is based on the premise that relevant results that are ranked lower should be penalized. Using Eq. (4.9), the NDCG at position $p$ is computed as

$$ NDCG_p = \frac{NDCG_p}{IDCG_p} $$ (4.10)

Here, $IDCG_p$ is obtained by sorting the reference set in descending order of their relevance to the query and computing the corresponding $DCG_p$ for this sorted list. Therefore, $IDCG_p$ represents the maximum possible DCG till position $p$. The NDCG for all the queries can be averaged to obtain a measure of the relevance of the ranked retrieval set to the query set.
In our experiments, we utilized the ground truth attributes for the images. For each query and an image in its retrieval set, we compute the relevance to the query as the number of attributes that the two images share i.e.

\[
rel_i = \sum_{j=1}^{n} 1\{a_j(q) == a_j(I_i)\}, \forall a_j(q) = 1
\]  

(4.11)

where \(1\{\cdot\}\) is the indicator function, \(n\) is the total number of attributes and variables \(a_j(q)\) and \(a_j(I_i)\) are binary valued based on the presence or absence of the \(j\)-th attribute in query \(q\) and the \(i\)-th ranked image \(I_i\). For this evaluation, images which share more attributes with a query are considered more relevant and the ideal retrieval set would give it a rank e.g. consider a scene from inside a city. It could be annotated with the scene attributes pavement, asphalt, concrete, man-made, sunny. For such a scene, a relevant retrieved image could be that of a street with buildings on a sunny day. To quantify the relevance of the retrieved images for such scenario, we computed the NDCG for different sizes of the retrieval set and the NDCG scores are presented in Table 4.11.

Table 4.11: Ranking performance for different features. We use NDCG for different retrieval set sizes to measure the performance. Relevance between query and retrieved image computed as count of attributes shared between them.

<table>
<thead>
<tr>
<th>Feature</th>
<th>NDCG@1</th>
<th>NDCG@5</th>
<th>NDCG@9</th>
<th>NDCG@15</th>
<th>NDCG@25</th>
<th>NDCG@35</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB Color Histogram</td>
<td>0.526</td>
<td>0.503</td>
<td>0.505</td>
<td>0.514</td>
<td>0.522</td>
<td>0.529</td>
</tr>
<tr>
<td>GIST [68]</td>
<td>0.536</td>
<td>0.543</td>
<td>0.549</td>
<td>0.550</td>
<td>0.558</td>
<td>0.566</td>
</tr>
<tr>
<td>Tiny Image [92]</td>
<td>0.434</td>
<td>0.447</td>
<td>0.452</td>
<td>0.464</td>
<td>0.482</td>
<td>0.494</td>
</tr>
<tr>
<td>Dense SIFT [57] Histogram</td>
<td>0.578</td>
<td>0.577</td>
<td>0.575</td>
<td>0.585</td>
<td>0.591</td>
<td>0.596</td>
</tr>
<tr>
<td>Visual + Semantic</td>
<td>0.594</td>
<td>0.569</td>
<td>0.580</td>
<td>0.594</td>
<td>0.603</td>
<td>0.611</td>
</tr>
</tbody>
</table>

We observe that amongst the visual features, the Dense SIFT histogram feature outperforms the other features in terms of retrieving relevant images that share attributes. When we combine the visual feature ranking with the similarity from the semantic label descriptor, the NDCG increases across all sizes. The increases varies from about 0.005 to 0.016. The categories of scenes which
typically showed an improvement with the use of semantic similarity with the visual similarity include computer rooms, dining rooms, bookstores, mountains and skyscrapers. Some of these categories were also part of the group of categories which performed better when using the semantic label descriptor for scene categorization in Section 4.4.2).

This improved performance further highlights the relationship between the multiple levels of information which can be associated with a scene. While our previous experiments showed a correlation between scene categories and the semantic structure of scenes (coded using the layout based semantic label descriptor), we are able to see an improvement in finding attribute-sharing images by analyzing their semantic labeling outputs.

4.5 Discussion

We have proposed an approach for non-parametric scene parsing using a $k$-NN method. We formulate our approach over small patches characterized by simple features which draws inspiration from biological vision. A locally adaptive distance metric is learned at query time to compute the relevance of individual feature channels. Using the initial labeling as a contextual cue for presence or absence of objects in the scene, we presented the semantic context descriptor which helped refine the quality of the retrieval set which is a key component of non-parametric methods for semantic labeling. The approach has been validated by experiments on several datasets and performance performance to state of the art techniques has been achieved. We also presented results for using our representation for the tasks for scene categorization and content image retrieval thereby correlating multiple scene understanding tasks with semantic segmentation as an image representation.
Chapter 5: Introspective Semantic Segmentation

5.1 Introduction

The work proposed in Chapters 3 and 4 operates under a closed world assumption of the dataset containing a fixed number of semantic labels. Splitting the dataset into training set and test set lends itself to the standard mode of evaluation of the trained classifiers. In practical settings, we would like to use the obtained models on unlabeled data which contain novel previously unseen semantic categories.

In this chapter, we present an approach for analyzing unannotated image data and discover regions in them which belong to previously unseen categories. For this, we adopt the aforementioned measure [71] which is based on $k$-nearest neighbor distances. The strangeness measure is computed for the instances of the labeled data and characterizes an instance’s uncertainty with respect to its own label. This is followed by the computation of the uncertainty for assigning a known semantic category to regions from unlabeled images by utilizing the strangeness measure again. The work presented in this chapter constitutes part of previously published work [85, 86].

5.2 Background

With an open universe i.e. datasets increasing in size and number of labels, the use of non-parametric approaches have shown notable progress for semantic segmentation and classification [16, 54, 90]. They are appealing as they do not need to be retrained as newer categories or images are added, can utilize efficient approximate nearest neighbor search techniques e.g. $k$-d trees [63] and contextual cues. Often, context is captured using a retrieval set of images which are similar to the
query and methods developed for establishing matches between image regions (at pixel or super-pixel level) for labeling the image. Authors in [90] formulate semantic labeling at the superpixel level. They retrieve similar images using global image features which is followed by superpixel-level matching using a wide variety of superpixel features and a Markov random field (MRF) to incorporate neighborhood context. This work was extended by [16] by training per superpixel per feature weights and also by incorporating superpixel level semantic context. Our approach for semantic segmentation is related to work of [16, 90] as we also pursue a non-parametric approach. However, we differ in the choice of features by also utilizing geometry based statistics which have displayed success in computing the geometric layout of scenes [31] and adopt a test time feature channel relevance learning method.

In our work, we are interested in quantifying the uncertainty of the predicted semantic labels. Different approaches have been proposed in the literature to associate classifier uncertainty on query data. An area where it has received significant attention is that of active learning where the task is to sequentially add labeled examples for training through human input on unlabeled samples but doing it with the added goal of minimizing the human effort. Hence, a critical component is to provide informative unlabeled samples for the human to label and classification uncertainty estimates are employed for selecting such samples. A commonly used uncertainty measure is entropy [32] which is an information-theoretic criterion computed over label likelihoods for an instance. An alternative measure is the best versus second best heuristic [36] computed as the difference between the probability values of the two labels having the highest estimated probability value with a low difference value indicating more uncertainty. This particular measure relates to the uncertainty between the most confusing labels in classification instead of using all the labels including ones which have low likelihoods as is done when computing entropy. In margin-based methods, the uncertainty of an instance has been characterized by its distance to the decision boundary between the classes [91]. An example which lies the closest to the boundary can be viewed as the one with the highest uncertainty. There has also been work [37] on utilizing Gaussian process classifiers for associating uncertainty based on the variance in the posterior. In our work, we propose to utilize the strangeness measure introduced by [71] to associate a confidence with the output of our
5.3 Semantic Segmentation Approach

In this section, we first review our non-parametric approach for semantic segmentation of images. It is based on the approach previously discussed in Section 4.3. We then describe the method for the computation of the strangeness measure in Section 5.4.

5.3.1 Problem Formulation

We follow the formulation presented in Eq. (1.1). In our approach, we formulate the semantic labeling of an image segmented into superpixels. The appearance likelihood $P(A|L)$ in Eq. (1.3) is obtained using a non-parametric $k$-NN method described in Section 5.3.3. As discussed in Section 1.1, many approaches to semantic segmentation model the joint prior $P(L)$ using a pairwise smoothness term in a Markov random field (MRF). However, we forgo the use of an MRF in our approach and utilize only the appearance likelihood for the semantic labeling of images.

5.3.2 Superpixels and features

To perform the semantic labeling of an image, we extract SLIC superpixels [1, 2] using the publicly available VLFeat library [93]. An example SLIC segmentation for an image is shown in Figure 5.1. SLIC superpixels obtained by this method are typically more regular shaped than the output of the graph based segmentation of [20].

To characterize the image superpixels, we use both geometric as well as appearance features to capture the statistics of individual regions. We adopt the same features where were previously used for describing the graph based segmentation [20] in Section 3.3. Most of the features were selected from the geometric layout estimation work of [31]. They provide information about color, texture, location, shape and perspective cues associated with a superpixel. The list of the superpixel features in provided in Table 3.1.
Figure 5.1: Example of the SLIC segmentation. (a) An input image (b) Corresponding superpixels.
5.3.3 Appearance Likelihood

We follow the approach in Section 4.3 which approximates the appearance likelihood with a Naive Bayes assumption yielding

\[ P(A|L) \approx \prod_{i=1}^{S} P(a_i|l_i). \] (5.1)

We use a non-parametric approach to obtain the individual label likelihood \( P(a_i|l_j) \) for a superpixel \( s_i \) which is obtained using a \( k \)-NN method. For each class \( l_j \) and every superpixel \( s_i \) of the query image, we compute a label likelihood score:

\[ \tau(a_i, l_j) = \frac{n(l_j, N_{ik})}{n(l_j, G)} / \frac{n(\bar{l}_j, N_{ik})}{n(\bar{l}_j, G)} \] (5.2)

where

- \( \bar{l}_j \) is the set of all labels excluding \( l_j \);
- \( N_{ik} \) is a neighbourhood around \( a_i \) with exactly \( k \) points in it;
- \( n(l_j, N_{ik}) \) is the number of superpixels of class \( l_j \) inside \( N_{ik} \);
- \( n(l_j, G) \) is the number of superpixels of class \( l_j \) in the entire dataset \( G \).

Note, the difference between Eq. (4.1) and Eq. (5.2) is that a retrieval set is not used for these experiments. We compute the normalized label likelihood score using the individual label likelihood:

\[ P(a_i|l_j) = \frac{\tau(a_i, l_j)}{L \sum_{l_k=1}^{L} \tau(a_i, l_k)}. \] (5.3)

Instead of using a basic \( k \)-NN approach that uses Euclidean distance over the concatenated region feature to compute the neighborhood around a point, we use the weighted \( k \)-NN method previously presented in Section 4.3.3. For computing a weighted distance between two image superpixels, we
split their individual feature vectors into five feature channels. These five features channels follow the grouping shown in Table 3.1 and contain information about (1) color (2) texture (3) location and size (4) perspective cues and (5) dense SIFT histogram. We obtain a vector of distances:

$$d_f = [d_{\text{color}}, d_{\text{tex}}, d_{\text{loc}}, d_{\text{line}}, d_{\text{sift}}]^\top$$

(5.4)

where \(d_{\text{color}}, d_{\text{tex}}, d_{\text{loc}}, d_{\text{line}}, d_{\text{sift}}\) are the Euclidean distances between the five feature channels (listed above) of the feature vectors of the two superpixels respectively. We define a weighted distance between the two superpixels as

$$d_w = w^\top d_f$$

(5.5)

where \(w \in \mathbb{R}^5\) defines the weights for the individual feature channel distances. Using the weighted distance from Eq. (5.5), we can now obtain the neighborhood around the query point. For the computation of the weights, we follow the method previously outlined in Section 4.3.3. We use the locally adaptive metric approach of [15] which is a query-based technique and computes a global metric at test time instead of pre-computed weights. We refer the reader to Section 4.3.3 for more details about the method for weight computation.

5.4 Strangeness Measure

While evaluating the semantic labeling output of an image is often the final goal of traditional approaches for semantic segmentation, in this work, we are also interested in computing the uncertainty of the semantic labels that we associate with an image. This is motivated by our desire to perform semantic segmentation of images with an introspective capability. In the previous section, we presented a non-parametric approach for the semantic labeling of an image. We now extend this method to help analyze a query image and identify image regions which instead of being associated with one of the known semantic categories can be characterized as unfamiliar. Our approach for this analysis is based on the concept of transduction in which an estimate about the properties of a
query point of interest is made directly from the training data as opposed to induction where first a
general rule is inferred from the training data and then applied to the query point.

Given a source set $T_s$ of fully annotated images, we segment the images of this set and compute
the features for the corresponding segments yielding the dataset of superpixels $G - (a, y)$ where $y$
is the segment label. For each segment $s_i$ in this dataset, an individual strangeness measure $\alpha_i$ is
computed

$$\alpha_i = \frac{\sum_{r=1}^{K} d^c_{ir}}{\sum_{r=1}^{K} d^{c_1}_{ir}}$$

where $c$ is the semantic label $y_i$ for instance $s_i$, $d^c_{ir}$ is the $r$-th shortest distance between $s_i$ and
an instance of class $c$, $d^{c_1}_{ir}$ is the $r$-th shortest distance between $s_i$ and an instance not belonging
to class $c$ and $K$ is the number of nearest neighbors considered for each sum. In our work, we
use the weighted distance of Eq. (5.5) for computing the nearest neighborhood around a point. The
strangeness measure is the ratio of sum of $K$ nearest distances from the same class to the sum of the
$K$ nearest distances from all other classes and it measures how “strange” an instance in question is
with respect to its semantic category. An example closer to other class instances in comparison to its
own class instances has higher strangeness and vice versa. To quantify the confidence of association
with a semantic category, we count the number of examples of the category in the dataset which have
a larger strangeness value and compute the p-value statistic proposed by [71]:

$$t_i = \sum_{r=1}^{K} \left\{ 1\{\alpha_i > \alpha_r\} \right\} / |c|$$

where $|c|$ is the number of instances in $G$ with the label $c$ and $1\{\cdot\}$ is the indicator function.
The value $t_i$ can be viewed as a measure of the probability of having instances in the class with
strangeness greater than or equal to that of $s_i$. Using Eq. (5.6) and Eq. (5.7), the strangeness and
p-values are computed for all the instances in $G$.

Now, given a set $T_u$ of query images on which we wish to investigate the introspective capacity of
our non-parametric approach for semantic labeling (with $D_u$ the segments corresponding to these query images), we wish to discover regions which do not belong to any of the semantic categories in $T_s$. In doing so, we also want to associate a measure of confidence for it to not belong to any of the familiar semantic categories. For this purpose, we utilize the strangeness measure from Eq. (5.6).

When computing the strangeness for instances in $G$, we already know the semantic label for the instance and the strangeness computation is straightforward. However, for the instances in $D_u$, we do not know the category of the instances. But note that our primary interest is determining if the instance belongs to any of the known semantic categories or not.

Therefore, we compute the strangeness and p-value for a query image region by assuming its putative label to be each of the known semantic categories $\{1, 2, \ldots, L\}$ one by one i.e. given a query instance $s_i$, we compute $\alpha^l_i$ and $t^l_i \forall l \in \{1, 2, \ldots, L\}$. The uncertainty for the region to belong to the known semantic categories is now defined as:

$$u_i = \min_{l=1}^L \left(1 - t^l_i\right)$$

(5.8)

Above, we compute the uncertainty of belonging to familiar semantic categories as a complement of $t^l_i$ as a lower $t^l_i$ corresponds to higher uncertainty for $s_i$ with respect to class $l$. The subsequent minimum function will select the class which is the least strange in comparison to $s_i$. Sample outputs for the strangeness based uncertainty are shown in Figure 5.2. In the figure, images have been labeled using the SiftFlow [54] dataset with semantic categories - sky, building, tree, mountain, road, sea, field. In the top row, a majority of the image regions have low uncertainty except for the road divider and the vehicle. The bottom row is an indoor scene where most of the image regions are associated with high uncertainty values.

### 5.5 Experiments

In this section, we present an evaluation of the approach proposed in this work. We first report the accuracy of our non-parametric approach for semantic segmentation. This is followed by an
evaluation of the strangeness measure for confidence based ranking of a set of query images.

5.5.1 Semantic Segmentation

We first evaluate the efficacy of the non-parametric approach for the problem of cross dataset semantic segmentation. Unlike the standard approach to semantic labeling where the evaluation is carried out by splitting a dataset into train and test sets, we consider source data from one benchmark set and evaluate models learned from it on another dataset. The motivation for doing so is to first establish the efficacy of the labeling approach for familiar semantic category segmentation before evaluating
any confidence based ranking. For this purpose, we consider three datasets of varying sizes which are commonly used by the research community for semantic labeling experiments. The details for the datasets are summarized in Table 5.1.

Table 5.1: Details for datasets used in our introspection experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Images</th>
<th>Categories</th>
<th>Scene Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford BG [26]</td>
<td>715</td>
<td>8</td>
<td>Outdoor</td>
</tr>
<tr>
<td>SiftFlow [54]</td>
<td>2688</td>
<td>33</td>
<td>Outdoor</td>
</tr>
<tr>
<td>SUN09 [8]</td>
<td>8662</td>
<td>107</td>
<td>Indoor + Outdoor</td>
</tr>
</tbody>
</table>

For the source dataset, we consider the two smaller datasets - Stanford and SiftFlow i.e. these constitute $T_s$. In both of these datasets, we select the seven most frequent background categories - *sky, building, tree, mountain, road, sea, field*. Evaluation of the non-parametric approach is carried out on the large scale SUN09 dataset. Pixels in SUN09 which do not belong to any of the seven categories are labeled *void*. As a baseline, we train boosting classifiers which have previously shown success in geometric layout computation [31] and semantic segmentation of urban environments [81]. Within the boosting framework, we use decision trees as the weak learners since they automatically provide feature selection. We learn separate classifiers for each of the seven semantic categories classes in a one vs. all fashion. Given a query image, the separate classifiers are run on the individual feature vectors of the superpixels of the image and output confidence scores. The class with the maximum confidence score is assigned to be a superpixel’s label. In our implementation, each strong classifier is composed of 25 decision trees with the tree size limited to 8 nodes. Table 5.2 and Table 5.3 report the performance of our approach against the baseline boosting method on the two different datasets. The evaluation criterion for the methods is the per pixel accuracy (percentage of pixels correctly labeled) and per category accuracy (the average of semantic category accuracies).

It can be observed that the results for our non-parametric approach and the boosting classifier are
Table 5.2: Results for familiar semantic category labeling on the SUN09 dataset using SiftFlow dataset as source for training instances. UKNN denotes uniform weight $k$-NN classifier while WKNN is the weighted $k$-NN. Strange KNN selects the least strange label to assign to a super-pixel.

<table>
<thead>
<tr>
<th>Labeling Method</th>
<th>Per Pixel</th>
<th>Per Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boosting</td>
<td>73.9</td>
<td>61.7</td>
</tr>
<tr>
<td>UKNN</td>
<td>70.4</td>
<td>58.5</td>
</tr>
<tr>
<td>WKNN</td>
<td>73.4</td>
<td>61.9</td>
</tr>
<tr>
<td>Strange-KNN</td>
<td>74.2</td>
<td>69.9</td>
</tr>
</tbody>
</table>

Table 5.3: Semantic labeling results on SUN09 using Stanford background dataset.

<table>
<thead>
<tr>
<th>Labeling Method</th>
<th>Per Pixel</th>
<th>Per Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boosting</td>
<td>68.0</td>
<td>59.4</td>
</tr>
<tr>
<td>UKNN</td>
<td>64.1</td>
<td>54.4</td>
</tr>
<tr>
<td>WKNN</td>
<td>67.6</td>
<td>58.9</td>
</tr>
<tr>
<td>Strange-KNN</td>
<td>68.5</td>
<td>60.9</td>
</tr>
</tbody>
</table>

While the uniformly weighted $k$-NN lags behind in comparison to the boosting output, utilizing feature channel relevance at query time leads to an improvement of more than 3% with both SiftFlow and Stanford datasets. The results for using SiftFlow are better than the output for Stanford due to the fact that SiftFlow and SUN09 share a few images as they are drawn from the large scale SUN database [103].

5.5.2 Confidence based Ranking

The next evaluation focuses on the introspective capacity of the proposed approach. Similar to the evaluation of the semantic segmentation, the introspection of the semantic labeling is also carried
out on the large scale SUN09 dataset. As mentioned previously in Table 5.1, both Stanford and SiftFlow are composed of outdoor scenes only while SUN09 consists of both outdoor and indoor scenes. For evaluation purposes, the category of the pixels which do not correspond to one of the seven aforementioned categories is set to the void label. With this processing, 3,321 images out of the 8,662 images in SUN09 dataset do not have a single pixel sharing a semantic label with the data from SiftFlow or Stanford datasets. Therefore, this provides an ample set of images on which we can evaluate our confidence based ranking method.

We compare the strangeness measure used in our approach to two baseline methods which have been previously utilized for computing classifier confidence in active learning experiments [32, 36]. For these, we utilize the boosting classifier evaluated in the previous section. Given a query image, for each region $s_i$, the boosting classifier provides the probability $p^l_i$ for assigning a semantic label $l$ to the region. Given this probability, we compute two metrics to characterize the uncertainty of labeling the region:

- **Normalized entropy of the boosting output (NEP)**

$$N_i = \sum_{l=1}^{L} -p^l_i \log (p^l_i) \log (L)$$  \hspace{1cm} (5.9)

Higher values for the normalized entropy implies more uncertainty in the labeling by the boosting classifier.

- **Best versus Second Best probability (BvSB)**

$$B_i = 1 - \left( p^l_{i1} - p^l_{i2} \right)$$  \hspace{1cm} (5.10)

where $p^l_{i1}$ and $p^l_{i2}$ are the highest two probability outputs from the boosting classifier. The lower the difference between $p^l_{i1}$ and $p^l_{i2}$, greater the uncertainty of labeling an instance.
The uncertainty measure presented in Eq. (5.8) is computed at the image region level. Using this uncertainty measure, we compute an image level uncertainty score. Given a query image $X_j$ composed of superpixels $\{s^j_1, s^j_2, \ldots, s^j_q\}$ where $q$ is the number of superpixels, the image level uncertainty of associating its regions with the familiar classes is computed as:

$$U(X_j) = \sum_{i=1}^{q} u_i \times \text{size}(s^j_i) \quad (5.11)$$

where $u_i$ is the uncertainty defined in Eq. (5.8) and $\text{size}(s^j_i)$ is the percent of image pixels corresponding to superpixel $s^j_i$ in $X_j$. When evaluating the NEP and BvSB methods, we substitute $u_i$ by $N_i$ and $B_i$ from Eq. (5.9) and Eq. (5.10) respectively.

Having obtained the image level uncertainty score, we sort the images of the query image set in a descending order. For any metric which is being evaluated for confidence based ranking, the goal is to obtain a higher number of images with unfamiliar categories in the (numerically) lower ranks e.g. in our evaluation, a metric performs better if an indoor scene of a bedroom with floor, walls and tables is assigned higher uncertainty (and subsequently lower rank) with respect to the familiar semantic categories than an outdoor scene of a beach. As part of this evaluation, we divide the resultant rankings into subsets and compute the percentage of void pixels present in each subset of the rankings. The results for the different measures when using SiftFlow and Stanford dataset are presented in Figure 5.3 and 5.4 respectively.

As can be observed, the normalized entropy and best vs second best probability difference measures based on probabilistic output of the boosting classifier perform inferiorly in comparison to the strangeness measure. In particular, we obtain a higher percentage of void pixels when using strangeness instead of normalized entropy or best vs second best probability in the images with high uncertainty scores e.g. when using SiftFlow on SUN09, images ranked 1-500 had 93.1% void pixels while NEP had 72.3%. As the ranks increase, there is a drop in the ratio of void pixels indicating that images with familiar categories are associated with lower uncertainty scores. The UKNN
Figure 5.3: Comparison of uncertainty measures for confidence ranking in the SUN09 dataset using SiftFlow dataset. The y-axis denotes the percent of void pixels in the images present in a particular ranking subset (each of size 500 images) when using an uncertainty measure e.g. there are 93.1% void pixels in images ranked 1-500 using WKNN based strangeness. Higher percent of void pixels in lower ranks implies a better performance.

...
Figure 5.4: Evaluation of uncertainty measures on SUN09 using Stanford as source dataset. The evaluation procedure is the same as Figure 5.3

descending order of their WKNN strangeness based uncertainty when using SiftFlow as the source dataset of instances. The image in the top left corner is ranked 1 while the image in right bottom corner is ranked 8501. It can be observed that the lower ranked sets are composed more from indoor scenes while the later ranks typically include outdoor scenes.

Some examples of the confidences associated with images are provided in Figure 5.6. Instances
Figure 5.5: Visualization of confidence ranking of SUN09 images in order of uncertainty based on WKNN strangeness with SiftFlow as source dataset. Under each image, we provide its rank.
(a)-(d) are examples of scenes where unfamiliar categories like *person, wall, floor* and *furniture* are associated with high certainty of being unfamiliar by WKNN strangeness but not necessarily by NEP. Example-(e) is an instance of an incorrect high uncertainty score by WKNN strangeness for the familiar categories of sky and building.

### 5.6 Discussion

We present a non-parametric approach for an introspective semantic segmentation of images. The approach is formulated over a single oversegmentation of an image which is labeled using a weighted $k$-NN approach where the weights are computed for individual feature channels at query time. The output of the non-parametric approach provides an introspective capability to analyze query image data by quantifying the uncertainty of semantic labels associated with the image regions. This is based on using the transductive *strangeness* measure which utilizes nearest neighbor distances. We presented results for confidence ranking of images from the large scale SUN09 dataset where our analysis shows that images with previously unseen categories present in them received high uncertainty scores.
<table>
<thead>
<tr>
<th>Image</th>
<th>Labeling</th>
<th>NEP</th>
<th>WKNN</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image_a" alt="Image" /></td>
<td><img src="labeling_a" alt="Labeling" /></td>
<td><img src="nep_a" alt="NEP" /></td>
<td><img src="wknn_a" alt="WKNN" /></td>
</tr>
<tr>
<td><img src="image_b" alt="Image" /></td>
<td><img src="labeling_b" alt="Labeling" /></td>
<td><img src="nep_b" alt="NEP" /></td>
<td><img src="wknn_b" alt="WKNN" /></td>
</tr>
<tr>
<td><img src="image_c" alt="Image" /></td>
<td><img src="labeling_c" alt="Labeling" /></td>
<td><img src="nep_c" alt="NEP" /></td>
<td><img src="wknn_c" alt="WKNN" /></td>
</tr>
<tr>
<td><img src="image_d" alt="Image" /></td>
<td><img src="labeling_d" alt="Labeling" /></td>
<td><img src="nep_d" alt="NEP" /></td>
<td><img src="wknn_d" alt="WKNN" /></td>
</tr>
<tr>
<td><img src="image_e" alt="Image" /></td>
<td><img src="labeling_e" alt="Labeling" /></td>
<td><img src="nep_e" alt="NEP" /></td>
<td><img src="wknn_e" alt="WKNN" /></td>
</tr>
</tbody>
</table>

N - 0.25(7011), W - 0.67(489)
N - 0.22(7660), W - 0.68(395)
N - 0.32(5175), W - 0.64(983)
N - 0.37(3553), W - 0.62(1398)
N - 0.33(4699), W - 0.65(918)

Figure 5.6: Qualitative results on SUN09 using SiftFlow dataset. Entries N and W denote the image level uncertainty score when using the NEP and WKNN strangeness measures respectively. The confidence rank corresponding to the uncertainty measure is provided in parentheses.
Chapter 6: Discussion

In this dissertation, we look at the problem of scene understanding in the context of semantic segmentation of images. A variety of tasks have been proposed by the computer vision community to help understand scenes and semantic segmentation provides a full pixel level parse of the image by annotating each pixel. In our work, we presented different methods, both parametric (boosting) and non-parametric ($k$-NN) for semantic labeling of images and used it as an image representation to address multiple tasks. In Chapter 3, we used off the shelf discriminative boosting classifiers to label street scenes. The labeling is performed on superpixels which were described by a rich set of features characterizing their appearance and geometry. The resultant labeling output is then summarized by the semantic label descriptor which is based on presence/absence of semantic categories in different spatial locations across the image. This descriptor is then used to induce a semantic topology over an urban environment where scenes sharing similar semantic structures are grouped together (Section 3.4.1). An added utility of the representation is the learning of related semantic concepts and as an example, we present results for detecting intersections in street sequences (Section 3.4.2).

With the growing nature of image databases with instances and categories added daily, methods which can effectively work an open universe setting would be highly valuable. Motivated by this goal, in Chapter 4, instead of training sophisticated parametric models, we described a non-parametric $k$-NN framework. In comparison to previously proposed methods, our approach was formulated over small superpixels described by simple features. This representation provides competitive results in comparison to others and we obtained significant improvements by utilizing the semantic label descriptor. The descriptor was used to refine the retrieval set of nearest neighbor images that serve as the source for labeling a query image and led to better semantic labeling with
particular improvement for the less frequent categories (Section 4.4.1). The refinement in the retrieval set occurs due to the learning of semantic concepts related to the category of the scene and the attributes present in it when we utilize the semantic label descriptor. This was validated by our analysis in Section 4.4.3 which showed a correlation between scene categories, scene attributes and the scene’s semantic labeling.

The traditional approaches to semantic labeling typically focus only on the accuracy of segmentation of known semantic categories in images. However, when the *open universe* nature of today’s databases, it is crucial that our models are not only accurate for the known set of categories but also be able to discover novel previously unseen semantic categories at the same time. This requires the ability to quantify the performance of existing models on unseen images. In Chapter 5, we presented a framework for non-parametric semantic labeling of images which can also associate an accurate estimate of the confidence of the labeling using the transductive *strangeness* measure. This measure is applied for confidence ranking of unlabeled images and we showed its efficacy for associating high uncertainty scores to images with unfamiliar categories (Section 5.5.2).

### 6.1 Future Directions

While we have been successful in improving the accuracy of semantic labeling methods and facilitating other scene understanding tasks, we have not achieved the grand of scene understanding by any means. In this section, we discuss possible directions for future work.

In Chapter 4, we presented a retrieval set based approach for semantic labeling of images. Currently, the retrieval set construction is straightforward - based on nearest neighbors computed with global image features. We observed an improvement in labeling with scene level context but as shown by the authors of SuperParsing [90], there is still further scope for improvement. In [90], it was shown that using a more accurate retrieval set based on selecting images sharing ground truth labels with a query, the parsing performance can be much improved. In the work of [58], it was shown that learning a per-exemplar distance metric greatly improves performance in tasks like object recognition.
and detection. Similar ideas could be pursued for learning per image distance metrics with supervision provided by their scene category and the semantic labels present in them. Another avenue to pursue to improve the quality of the retrieval set could be to use a better image feature e.g. learning the representation from a convolutional neural network [40].

In all the experiments presented in this dissertation, we assumed a flat hierarchy of categories where an image site is associated with one of these categories. Recently, there has been work in the domain of object recognition [12, 14] which has looked at exploiting a label hierarchy [13] which can associate labels from different levels of the hierarchy e.g. dogs, cats are animals while cars, trucks are vehicles. In the case of semantic labeling, instead of associating a single label from the set, an image site could be associated with a node in a hierarchy based on the evidence available. For example, a grass patch could be categorized as vegetation if there is insufficient evidence to disambiguate it as grass, trees or flowers. This level of supervision could also be extended to the computation of uncertainty.

In addition to exploiting a hierarchy of labels, another possible direction is to associate multiple labels with image sites. An example would be to associate both semantic and geometric labels with image regions e.g. buildings which are vertical. This kind of semantic information for a scene could also better associate with other information for the scene. To elaborate further, using the same example, a region in the scene can be labeled as a building. At the attribute level, a possible annotation for the scene could be skyscraper. This could provide an avenue for using a shared representation for learning multiple tasks [34, 56, 72, 95] e.g. semantic labeling of image regions and associating semantic attributes with them.

In Chapter 3, we presented experiments which performed a topological mapping of a street sequence. As part of our analysis, we observed that revisited locations are clustered together with a high rate. This suggests the possibility of using semantic structure of scenes for image retrieval in relation to the problem of geolocation. In the large scale geolocation experiment of [30], query images were localized on the basis of the retrieval set constructed using global image features like GIST [68], Tiny Images [92], and color histograms. However, their method only uses low level
image features and as shown in this dissertation, semantic labeling summarized by the semantic label descriptor helps characterize geographic locations. In Chapter 4, we showed an improvement in the retrieval set of nearest neighbors for scenes using semantic labeling as an image representation. A similar intuition could be applied to geolocation where semantic similarity can be used in addition to low-level image features for retrieving images from close-by geographic locations for a query image. Ideas which look at correlating collected images with semantic information besides geo-tags have recently started gaining attention of the community [38, 109] and we hope that the use of semantic representations can further improve this and related tasks.
Bibliography
Bibliography


96


Curriculum Vitae

Gautam Singh is a Ph.D. student in the Department of Computer Science at George Mason University, Fairfax, VA, USA. He received his Bachelor of Technology degree in Computer Science and Engineering from International Institute of Information Technology, Hyderabad, India in 2007 and a Masters in Computer Science from George Mason University. He will be completing his Ph.D. in the Fall-2014 semester. His research interests are in the areas of computer vision, machine learning and robotics with an application to scene understanding from images and/or videos. His research has applied these interests to problems like semantic parsing of scenes, scene categorization, content based image retrieval, topological mapping and location recognition in urban environments.