Road Detection and Semantic Segmentation without Strong Human Supervision

Ankit Laddha

CMU-RI-TR-16-37

July 29, 2016

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Prof. Martial Hebert
Prof. Kris Kitani
Mr. Ishan Misra

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

Keywords: Road Detection, Semantic Segmentation, Weak Supervision, Tag-Supervision, Self Supervised Learning, Deep Learning, Pixel Level Labeling, Structured Output, Convolutional Neural Networks, Fully Convolutional Networks

To my parents for always believing in me

Abstract

Recently, convolutional neural networks (CNNs) trained with strong human supervision have shown to achieve state of the art performance for both road detection and semantic segmentation. However, collecting strongly labeled data for both require detailed per-pixel annotations from humans which renders data annotation highly costly and time consuming. Therefore, in this work we propose methods to train a CNN for both of these tasks without using strong human supervision.

For road detection , we propose a two-step self-supervised method which does not require any human image annotation. Firstly, we automatically generate road annotations for training using OpenStreetMap, vehicle pose estimation sensors, and camera parameters. Next, we train a fully convolutional network (FCN) for road detection using these annotations. We show that we are able to generate reasonably accurate training annotations on KITTI data-set [14]. We achieve state-of-the-art performance among the methods which do not require human annotation effort.

For semantic segmentation, we use image-level tag annotations to learn a dense pixel-level prediction model. These tags indicate the presence or absence of various classes in an image. We propose a novel graph regularized multiple-instance multi-label (G-MIML) loss to train the FCN. The MIML loss encodes the constraints provided by image-level tags. The superpixel level graph over an image encodes the inherent label smoothness assumption. The proposed loss yields state-of-the-art performance on tag-supervised semantic segmentation on PASCAL VOC 2012 [12] data-set.

Acknowledgments

This thesis would not been possible without the guidance and help of several people who have contributed to it in one way or another.

First of all, I would like to thank my advisor Prof. Martial Hebert for his guidance and support. I am grateful for the opportunity to be able to develop and explore ideas during my stay at Carnegie Mellon University. His experience and knowledge has been invaluable in my work which has resulted in this written thesis.

I am also grateful to Mehmet K. Kocamaz and Luis E. Navarro-Serment for providing invaluable suggestions and advice. I am grateful for the long talks we had on various topics.

I would also like to thank Achal Arvind, Shreyansh Daftry and Ishani Chatterjee for their encouragement whenever I felt disheartened. Your support has helped me in some very difficult times.

Finally, I am thankful to Prof. Dhruv Batra for providing me the internship opportunity at Virginia Tech after graduation. My experiences there really prepared me for the masters program at CMU.

Contents

1 Introduction **1**
- 1.1 Contributions . . . 1
  - 1.1.1 Road Detection . . . 2
  - 1.1.2 Semantic Segmentation . . . 2

2 Map-Supervised Road Detection **3**
- 2.1 Introduction . . . 3
- 2.2 Related Work . . . 4
- 2.3 Approach Overview . . . 5
- 2.4 Automatic Road Annotation . . . 5
  - 2.4.1 Overview . . . 5
  - 2.4.2 Initial Labeling using OpenStreetMap (OSM) . . . 6
  - 2.4.3 Label Refinement . . . 6
  - 2.4.4 Determining Number of Clusters (K) . . . 8
- 2.5 Road Detection . . . 8
  - 2.5.1 Model . . . 8
  - 2.5.2 Training . . . 9
  - 2.5.3 Inference . . . 9
- 2.6 Experiments . . . 10
  - 2.6.1 Dataset . . . 10
  - 2.6.2 Analyzing Performance of Automatic Labeling Method . . . 10
  - 2.6.3 Analyzing Performance of the Road Detection . . . 13
- 2.7 Summary . . . 15

3 Tag-Supervised Semantic Segmentation **17**
- 3.1 Introduction . . . 17
- 3.2 Related Work . . . 18
- 3.3 Tag-Supervised Learning . . . 19
  - 3.3.1 Objective Function . . . 20
  - 3.3.2 Multiple Instance Multi-Label Loss . . . 20
  - 3.3.3 Similarity Graph based Regularization . . . 21
- 3.4 Implementation Details . . . 22
  - 3.4.1 Superpixels . . . 22
  - 3.4.2 Similarity Graph . . . 22

3.4.3 Network Architecture . . . 22
3.4.4 Training . . . 23
3.4.5 Inference . . . 23
3.5 Experiments . . . 24
3.5.1 Dataset . . . 24
3.5.2 Results . . . 24
3.6 Summary . . . 26

4 Conclusion **31**
4.1 Summary . . . 31
4.2 Future Work . . . 31

Bibliography **33**

List of Figures

2.1 Sample image with the annotated drivable road area . . . 3
2.2 Proposed Approach . . . 4
2.3 Errors in the initial road labeling . . . 7
2.4 Label Refinement Procedure . . . 7
2.5 Plots on *train set* to determine number of clusters . . . 10
2.6 Successful cases for the automatic labeling of images . . . 11
2.7 Failure cases for the automatic labeling of images . . . 12
2.8 Effect of different values of K on our proposed automatic labeling approach. . . . 13
2.9 Qualitative Examples of the Detected Road using the CNN trained with automatically labeled data on the *test set* . . . 14

3.1 Tag-Supervised semantic segmentation approach overview. . . . 18
3.2 Proposed training setup for tag-supervised semantic segmentation. . . . 23
3.3 Paired Graph showing accuracy difference for each image in the *val set* between our method and [35]. . . . 26
3.4 Example successful cases for our tag-supervised approach. . . . 27
3.5 Example failure cases for our tag-supervised approach. . . . 28

List of Tables

2.1 Quality of the machine generated labels. 11
2.2 Quality of machine generated labels after each step of the proposed algorithm. For details see 2.6.2. 12
2.3 Road detection performance of various methods on the *test set*. Note that, ↑ denotes higher is better and ↓ denotes lower is better. 13

3.1 Comparisons of various tag-supervised approaches on the *val set* using Aggregate Results. 25
3.2 Performance of each label in the *val set*. 25
3.3 Comparisons on *test set* using Aggregate Results. 27

Chapter 1

Introduction

A variety of tasks in computer vision require pixel-level prediction, for example, foreground segmentation, material segmentation, geometric scene understanding etc. For all of these tasks, the goal is to automatically assign each pixel x_i in an image with a predefined set of classes $c \in 1, \cdots, C$. Recently, convolutional neural network (CNN) based approaches [8, 30] have been shown to perform very well for these tasks. However, they require strong human labeled data in terms of task specific pixel-level class labels. This requirement restricts the applicability of these approaches because pixel labeling is a very costly and time consuming task for humans. Therefore, in this thesis we investigate the general question: Is it possible to train a CNN for pixel-level prediction tasks without strong human supervision?

In recent years, CNNs have revolutionized the field of computer vision. The seminal work of Krizhevsky et al. [24] has been the catalyst for this. They trained the network using a large amount of data, more than 1 million images, for a generic 1000-way object classification task. Their approach outperformed all the classical computer vision approaches on ILSVRC12 image classification challenge by a large margin. The classical methods used human engineered image representation whereas they used CNN to learn a hierarchical representation of images without requiring any human supervision for feature learning.

Later, [11, 42] showed that a network trained using a large amount of data for object recognition could be used as a generic feature extractor for other computer vision tasks. The output of intermediate layers was considered as features and used for various tasks such as scene classification, fine-grained recognition, object detection, image retrieval etc. The performance on all of these tasks using the intermediate representation from CNNs was shown to be superior than the hand crafted features. This observation led to the use of CNNs in many more computer vision tasks. For example, [8, 30] have shown that CNN based approaches outperform the classical approaches for pixel-level predictions. Due to the superior performance of CNNs on various computer vision tasks, we use them as the predictive modeling tool in this thesis.

1.1 Contributions

In this thesis, we propose approaches to to train CNNs for pixel-level prediction tasks without using task specific strong human supervision. Specifically, we tackle the tasks of road detection

and semantic segmentation in monocular images.

1.1.1 Road Detection

For the task of road detection we propose a self-supervised method that does not require any human road annotation effort in images. For this part, our main contributions could be summarized as follows:

- We propose a novel, scalable and cost effective method to automatically generate drivable road area annotations using localization sensors (GPS and IMU) on the vehicle and publicly available noisy OpenStreetMap[1] data.
- We train a CNN using these noisy labels for road detection and outperform all the methods which do not require human effort for image labeling on KITTI [14] data-set.

1.1.2 Semantic Segmentation

For the task of semantic segmentation we propose to use the human labeled image-level tag naming the objects present in the image to train a CNN. For this part, our main contributions could be summarized as follows:

- We propose a novel graph regularized multi-instance multi-label (G-MIML) loss to train a dense pixel-level prediction CNN using image-level tags.
- We perform extensive experimentation using the PASCAL VOC 2012 [12] data-set and show that our method outperforms previously proposed tag-supervised methods.

[1] https://www.openstreetmap.org

Chapter 2

Map-Supervised Road Detection

2.1 Introduction

Recognizing the obstacle-free road region to drive in front of the vehicle (e.g. Figure 2.1) is a very important information. It is essential for autonomous driving and useful for advanced driver assistance systems (ADAS). Researchers have used various sensors such as laser scanner [43], range scanner, stereo camera pair [49] and monocular camera [50] for this. In this chapter, we are interested in using images from a monocular camera to detect the collision-free road area.

The top performing methods [26, 32, 33] which use the publicly available KITTI benchmark [14] for the performance evaluation follow the human supervised learning paradigm. They collect images by driving a vehicle and ask humans to outline the drivable road area. These labeled images are used to train a classifier. This classifier is then used to predict free road space in images at test time.

Adapting these methods to new scenarios is hard because they require considerable human effort to produce new training annotations. If the labeled examples for training can be automatically generated, we can mitigate two major problems: scalability and cost. Therefore, in this chapter we are interested in addressing the following questions: Can we automatically generate the road annotations without any human intervention and use them to train a road classifier?

Our approach is inspired from a simple observation. A map of the area is essential for navigation while driving, even for humans. We use this necessary and already publicly available

Figure 2.1: Sample image with the annotated drivable road area. Note that out of the two parallel road in the image only one is labeled as drivable. Also, the road region occluded by the car is not labeled as drivable.

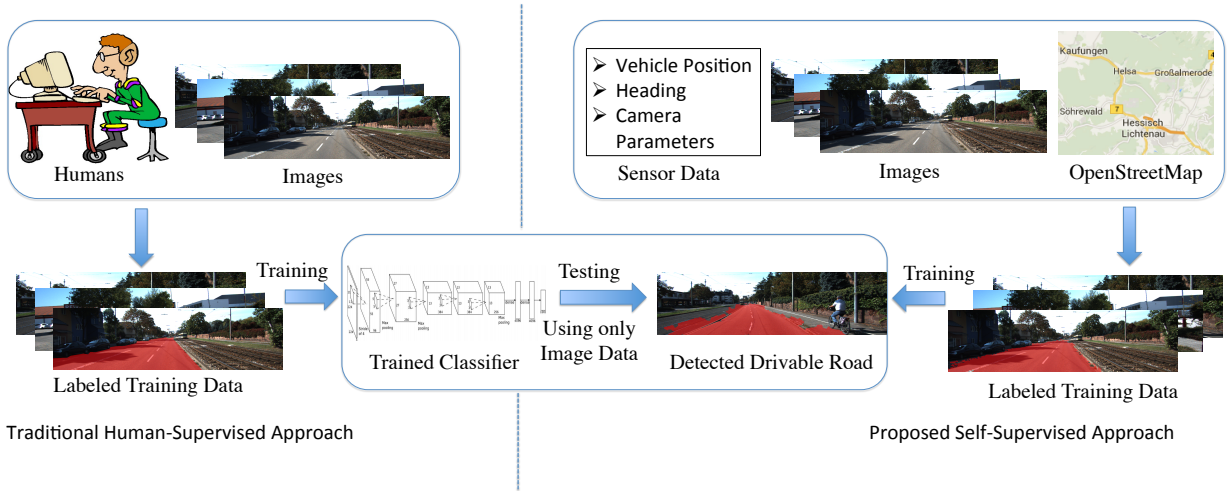


Figure 2.2: The pipeline of human-supervised and our proposed map-supervised road detection approaches. Human-supervised approaches use human to label the training data. Whereas, we use publicly available OpenStreetMap data, vehicle pose, camera parameters and pixel appearance features to label the training data. During testing we only use the image data.

information along with other localization sensor information to automatically label images for training. However, the map usually is very coarse and rife with errors. This problem is compounded by the presence of dynamic objects, such as the cars and pedestrians for which there is no information available in the map. Also, the localization sensors employed on the vehicle might be noisy.

The problems outlined above lead to errors in the labeling. We exploit the appearance features of the image to reduce the errors in the automatic annotation process. We then use these automatically generated annotation to train a Convolutional neural Network (CNN) model for road detection. During testing, we only use image information which allows us to generalize to areas without GPS or with poor signal quality. Figure 2.2 shows an overview of our approach and compares it to the traditional human-supervised paradigm.

2.2 Related Work

Monocular Road Detection: The majority of the image based drivable road detection approaches follow one of three paradigms: Human-supervised, Self-supervised, and Unsupervised.

The human-supervised approach is followed by [26, 32, 50]. They use human-generated annotations to learn a road model using powerful supervised machine learning algorithms. [26, 32] use Convolutional Neural Networks (CNN), and [50] uses a 1-D graphical model and mixture of Gaussians. These approaches are typically the best performing ones. However, they are costly and unscalable due to the human effort involved. In our approach, we also use a CNN, but we reduce the cost of training by removing humans in the image labeling step.

Unsupervised approaches use only a single image at any given time. These approaches work either by finding road borders using color and texture information [9, 21] or by building a color-

based model of the road points [2]. Road boundary detection based methods usually make assumptions about the shape of the road. Pixel based model methods assume that the center-bottom part of the road belongs to road. These approaches work well in highway scenes, but their performances leave a lot of room for improvements in urban areas.

Self-supervised algorithms [27, 37, 45] rely on the past predictions to adapt an existing model or train a model for the current image. They can be seen as operating between the human-supervised and unsupervised approaches. These methods employ very simple models because of the need to rapidly adapt the model on the fly. [45] uses color based template matching, [27] and [37] uses color based mixture of Gaussians. They also suffer from model drifts, so they require resetting the algorithm after some time. Our approach can be considered as a part of this category but we train a CNN in our offline training step.

Using Maps: Maps provide rich information about static man-made elements of the scene. However, this information is approximate and noisy. Therefore, [18, 48] used map as a prior in their scene labeling algorithms. They require human labeled images for training. [4] uses maps as prior for online road detection, but their algorithms also require some human annotations. [31] uses maps to label aerial images and require map information at the test time. Whereas, we use maps to label images taken from ground and our classifier does not need any map information during the testing.

Training with Machine Generated Labels: [3] uses structural information of a scene predicted by [17] for road detection. They first segment the image into horizontal and vertical surfaces and sky. They use these categories to aid their online road detection. In the experiments, we show that our approach compares favorably with theirs.

2.3 Approach Overview

Our goal is to recognize pixels denoting the drivable road area in a monocular image. As shown in Figure 2.2, our approach includes two steps:

- First, we automatically build a set of noisy labeled images using maps, localization sensor data and camera parameters. We reduce the annotation noise using pixel appearance features.
- In the following step, we train a Fully Convolutional Network using these automatically generated labels.

The main difference between the traditional supervised learning paradigm and our approach is that we use machine generated labels to train the classifier, thereby eliminating the human effort involved.

2.4 Automatic Road Annotation

2.4.1 Overview

Training a statistical model requires to have labeled instances which are representative for the distribution of data. We generate the training samples by exploiting information from maps in

the following steps:

- First, we use vehicle pose and maps to reconstruct the 3D scene around the vehicle.
- We then get an initial labeling of the images by projecting the reconstructed scene onto the image plane using a calibrated camera. This initial labeling is too noisy due to occlusions, and errors in the map data, vehicle pose and calibration parameters.
- In the third step, we refine this labeling based on pixel appearance to reduce the error.

We use the publicly available OpenStreetMap (OSM) as the source of map information.

2.4.2 Initial Labeling using OpenStreetMap (OSM)

OSM provides information about static structures existing in the scene. We use this data for two types of objects: Roads and buildings. We first reconstruct the 3D scene of a 100x100m^2 area around the vehicle. To populate the scene with roads and buildings, we use the GPS coordinates of the boundaries of buildings and coordinates of the center line of the roads. Other properties such as number of floors in the building, height of each floor, type of road, number of lanes, width of each lane etc are also extracted from the map database. However, since these are not always present, we make assumptions about them based on the geographic knowledge (e.g. residential roads have one lane and are 3m wide). Subsequently, we project this reconstructed scene onto the image plane using a calibrated camera.

This projection results in each of the pixels in the image being labeled with: road, building or none. The pixels labeled as none or building are combined to form the non-road class. The projected labels are very noisy mainly due to sensor errors, presence of dynamic objects, and erroneous or absent lane width data.

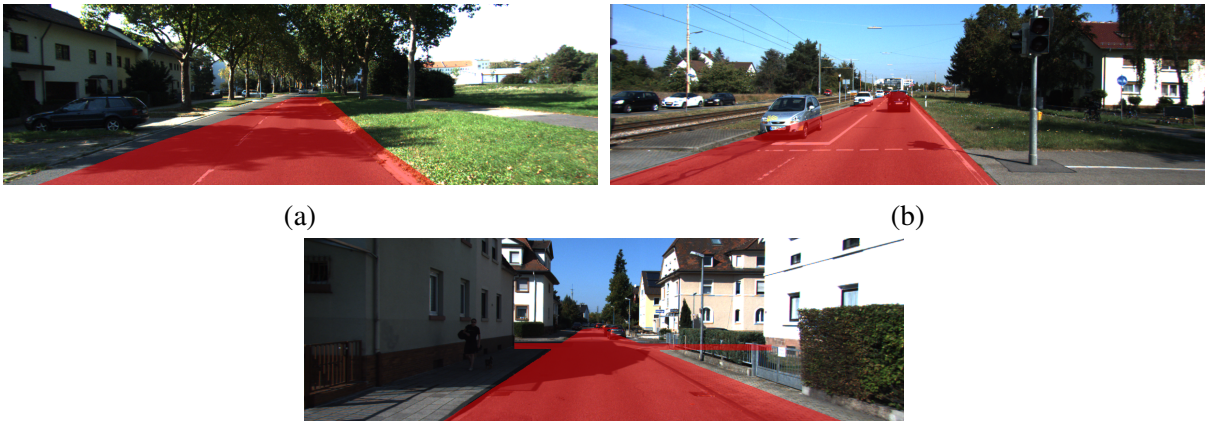
Approximate vehicle pose estimation causes errors in the relative position of the scene with respect to the vehicle. It causes the projection of labels to be displaced (Figure 2.3a) from their actual position. OSM provides information about the static elements in the scene, but it does not provide information about the dynamic objects, such as cars and pedestrians (Figure 2.3b). The map projection mislabels dynamic objects that occlude the road. Erroneous or absent lane width data causes over/under estimation of the extent of a road (Figure 2.3c).

2.4.3 Label Refinement

To reduce the noise in the initial labeling step, the images present in the training set are relabeled such that pixels with similar appearance are assigned to the same label.

We use the following approach: First, we cluster the pixels based on appearance and then we assign a label to each cluster based on the statistics of the pixels in that cluster. Assume we have K clusters, for each cluster i, nr_i denotes the number of road pixels, nnr_i denotes the number of non-road pixels after the initial labeling step.

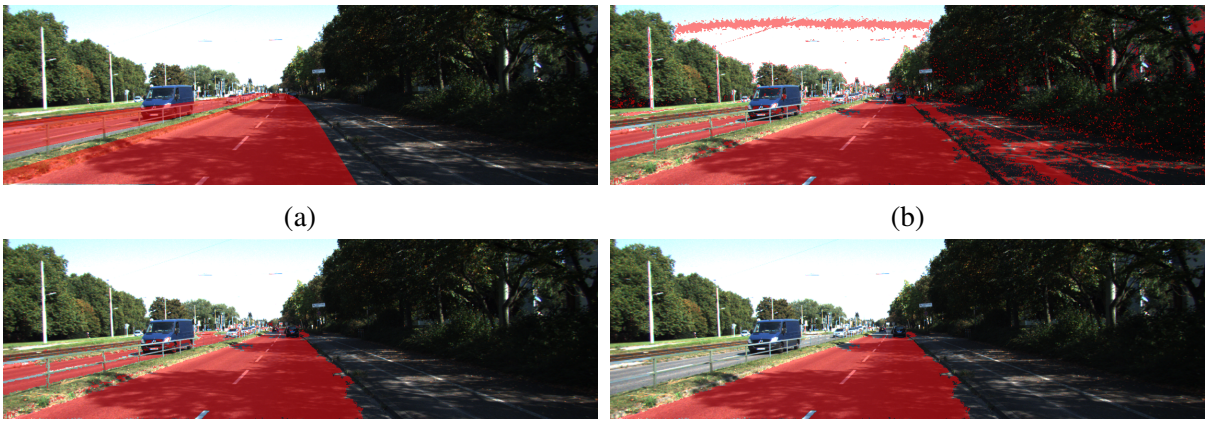
One way to label a cluster i is to define a ratio r_i such that $r_i = nr_i/nnr_i$ and label a cluster i road if $r_i \geq 1$ and non-road if $r_i < 1$. However, it will fail if the number of non-road pixels in a data set is much greater than road pixels, since most of the clusters will be assigned to non-road in this case. Therefore, in order to normalize the number of pixels for both classes, we modify



(a) (b)

(c)

Figure 2.3: Errors in the initial road labeling. Examples of errors: (a) due to sensor noise (road region is shifted to the right). (b) due to dynamic objects (mislabeled cars). (c) due to erroneous lane width data (mislabeled sidewalk).



(a) (b)

(c) (d)

Figure 2.4: Label Refinement Procedure: (a) Initial labeling using OpenStreetMap. (b) Label Refinement using K-Means. (c) Improved Labeling after restricting the allowed road pixels. (d) Final Labeling after removing the non-drivable parallel road.

the ratio r_i to r_i^{mod} as follows and use it:

$$r_i^{mod} = \frac{nr_i / \sum_{i=1}^{K} nr_i}{nnr_i / \sum_{i=1}^{K} nnr_i} \tag{2.1}$$

We use color features to represent the appearance of pixels. In particular, we use HSI and YCbCr color spaces. To achieve robustness from shadows, we only use the H and S channels from HSI space, and Cb and Cr channels from YCbCr space.

To reduce the false positives for the road class, we restrict the candidate road pixels. We divide the image into superpixels by thresholding the gPb [10] field, using a very low threshold value (0.01 in our case). We observe that the true road pixels are close to the OSM projected road pixels. Therefore, we consider a superpixel as road candidate if more than 10% of its area was predicted as road in initial labeling. We consider all the pixels present in the road candidate superpixel as road candidate pixels.

Some images contain multiple roads separated by a divider. We are looking for the drivable road area, so detecting any other road except our current road is a false positive. As drivable road area, we select the largest connected component in the image labeled as road. Figure 2.4 shows results after each step of label refinement for a sample image. It demonstrates that we are able to reduce the noise in the road annotations.

2.4.4 Determining Number of Clusters (K)

We use K-means clustering which requires the number of clusters (K) as input. We cannot use the standard cross validation approach: Select K which maximizes the validation set performance, because we do not have any human labeled ground truth data. We use the L method [41] to determine K. This method finds the knee of the K vs clustering evaluation metric graph by fitting a pair of straight lines to it. We use the Sum of Squared Error (SSE) as the evaluation metric.

Assume that there are n candidates $\{k_1, k_2, \cdots, k_n\}$ for K, such that $k_1 > k_2 > \cdots > k_n$. The L method determines the knee in the graph of K vs SSE as follows: For each $i = \{2, \cdots, n-2\}$, it splits the set of candidates into two parts $L_i = \{k_1, \cdots, k_i\}$ and $R_i = \{k_{i+1}, \cdots, k_n\}$. Then, it fits separate lines to the parts of graph belonging to these sets and calculates the Root Mean Squared Error (RMSE) for those lines. Lets denote RMSE for L_i and R_i as $rmse_{L_i}$ and $rmse_{R_i}$ respectively. The the total RMSE at pivot k_i ($rmse_{k_i}$) is a weighted sum of two RMSEs:

$$rmse_{k_i} = \frac{k_i - k_1}{k_n - k_1} rmse_{L_i} + \frac{k_n - k_{i+1}}{k_n - k_1} rmse_{R_i} \tag{2.2}$$

The knee point is the k_i which minimizes the $rmse_{k_i}$.

$$K = \underset{k_i}{\operatorname{argmin}}\, rmse_{k_i} \tag{2.3}$$

2.5 Road Detection

2.5.1 Model

We use a Fully Convolutional Network (FCN) for road detection. A FCN only contains convolutional layers so it could produce output at each pixel. Fully connected layers in a classification network can be easily converted into convolutional layers. Each node in the fully connected layer can treated as a filter whose spatial size is same as the number of edges incident on that node. For example the first fully connected layer in [44] has 4096 nodes and each node has 49 incident

edges. So this can be considered as a convolutional layer with 4096 filters with spatial size of 7x7.

We use the *Deeplab-LargeFOV* network proposed by [8] for our purpose. They modify and convert the 16-layer network classification network from [44] to a FCN. After converting the network of [44] to a fully convolutional one, the first fully connected layer has 4096 filters of large spatial size of 7x7. This becomes a computational bottleneck. Therefore, [8] reduces the number of filters to 1024 and spatial size to 3x3 by subsampling. Similarly, the size of second fully connected layer is also reduced to 1024 filters.

The FCN they get from directly converting the network from [44] produces a very coarse output with stride of 32. To increase the resolution [8] skip subsampling after the first three max-pooling layers in the network. This allows the network to produce an output with stride of 8. However, the remove of subsampling reduces the field-of-view of the network. So, to increase the field of view of the networks they use dilated convolutions [51]. Let's assume that h is a kernel of size $2r + 1 \times 2r + 1$ and g denotes the 2D matrix over which we want to use the convolution operator. Then the dilated convolution operator $*$ with dilation factor l is defined:

$$f_l(i,j) = g * h(i,j) = \sum_{m=-r}^{r} \sum_{n=-r}^{r} g(i - lm, j - ln)h(m,n) \tag{2.4}$$

In our case, we use a dilation factor of 2 for the last three convolutional layers on VGG-16 and a factor of 12 for the first fully connected layer. The final layer of our adaptation is a convolutional layer with 1x1 filters and 2 channels (one for road and another for non-road).

2.5.2 Training

We use the publicly available code[1] by [8] for implementing and training the network. We use soft-max loss at every pixel for back propagation. The network by [44] is already trained on image classification using ImageNet data [40]. We fine-tune the network with initial learning rate of 0.001 and batch size of 5. After every 5 epochs we multiply the learning rate by 0.1. We use a momentum of 0.9. For data augmentation during , we randomly mirror around the vertical axis and crop a 1217x353 dimensional patch from the image.

2.5.3 Inference

At test time, we pass the full image through the trained network to predict the probability of road label at each location in the image. However, the network produces the output with a stride of 8. So, we use bi-linear interpolation to increase its resolution to the image size.

[1] https://bitbucket.org/deeplab/deeplab-public/src

2.6 Experiments

2.6.1 Dataset

We use the KITTI data set [14, 15] to evaluate our approach. It contains a diverse set of road annotations of various different scenes taken in a span of various days. It consist of two sets: *train set* containing 289 image and *test set* containing 290 images.

We use the evaluation protocol defined by [14]: Results are evaluated in birds eye view (BEV) using per pixel metrics. We use Precision (PRE), Recall (REC) and F-measure (F) to evaluate the performance of automatic labeling. In addition to these metrics, we also use Average Precision (AP), False Positive Rate (FPR) and False Negative Rate (FNR) to evaluate road detection.

2.6.2 Analyzing Performance of Automatic Labeling Method

The evaluations in this section are done by comparing with the available human annotated road labels for *train set*.

Determining Number of Clusters K

The first step in our algorithm is to find the number of clusters (K) required for the clustering algorithm. Figure 2.5 shows the graph of K vs Sum of squared error (SSE). To find the knee of this graph, we plot the graph of K vs Total RMSE using the L method [41] (See Figure 2.5). Based on this, we select $K = 70$ since it has the minimum RMSE.

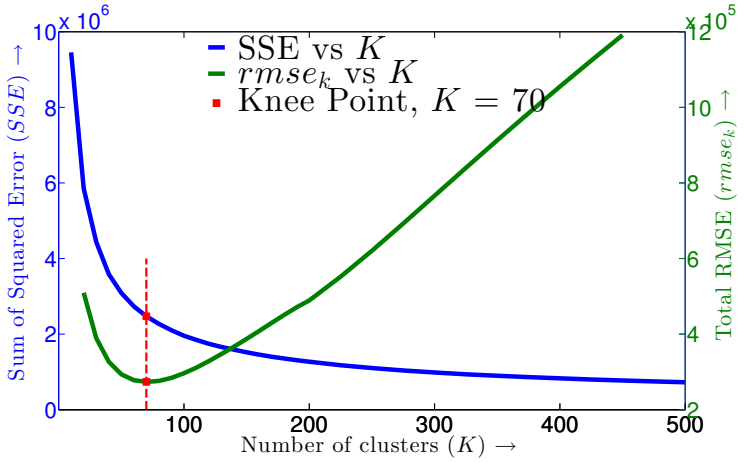


Figure 2.5: Plots on *train set* to determine number of clusters. We can see that knee of the K vs SSE graph occurs at $K = 70$ because the $rmse_k$ is minimum at that point.

Results

Table 2.1 shows that the label refinement step is able to reduce the noise in the initial projection of OpenStreetMap data.

Method	F	PRE	REC
Our Approach	**85.51**	84.16	**86.90**
Co-Labeling [5]	84.74	**88.02**	81.69
Map Projection	78.93	74.97	83.34

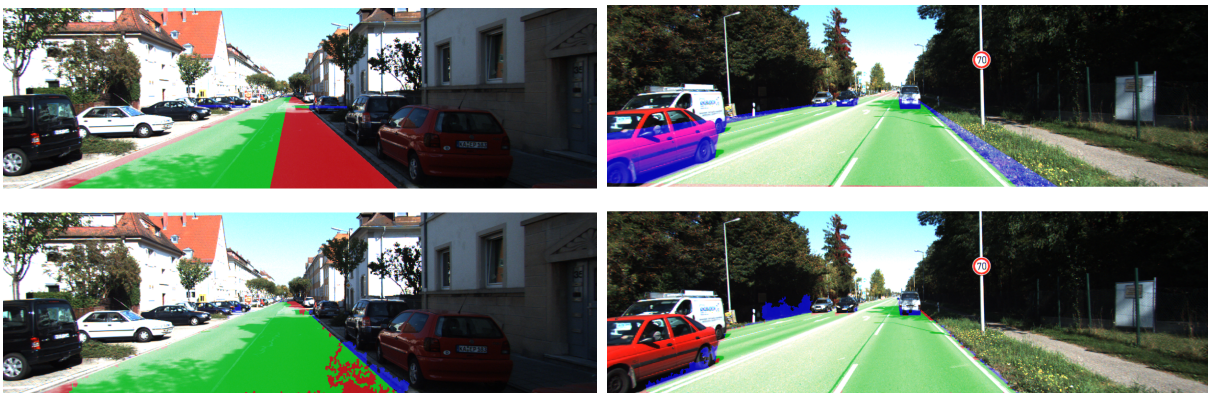
Table 2.1: Quality of the machine generated labels.

We also compare our label refinement method with the co-labeling approach of [5] which extends the fully connected conditional random field of [22] to label a large set of images simultaneously.

We use the initial labeling obtained by projecting the map data onto the image plane as the unary term. To set the pairwise term, we use the same appearance features as we used in our approach. We also select the largest connected component as the final drivable road region which is the same as our approach. We give the co-labeling approach a slight advantage by setting the model parameters such that they maximize the F-measure on *train set*. Both methods are based on the assumption that similar pixels should be assigned to the same label. We see that our method and co-labeling have similar performances. However, our method does not require any ground truth to select model parameters.

Qualitative Results

We show some successful cases where we are able to increase the quality of the initial annotations by map projection using appearance features in Figure 2.6a and 2.6b. Figure 2.7a and 2.7b shows some of the failure cases. The main reasons for the failures are: Extreme shadows, and



(a) (b)

Figure 2.6: Successful cases for the automatic labeling of images. Top road displays results after initial map projection and bottom row shows results after our label refinement approach. In (a), our approach is able to extend the road annotation to cover the whole road. In (b), we successfully removed the incorrectly labeled cars from the road class. The color coding is as follows: Green - True Positive, Red - False Negative, Blue - False Positive for road.

similar appearance of the road and sidewalk.

Ablative Analysis

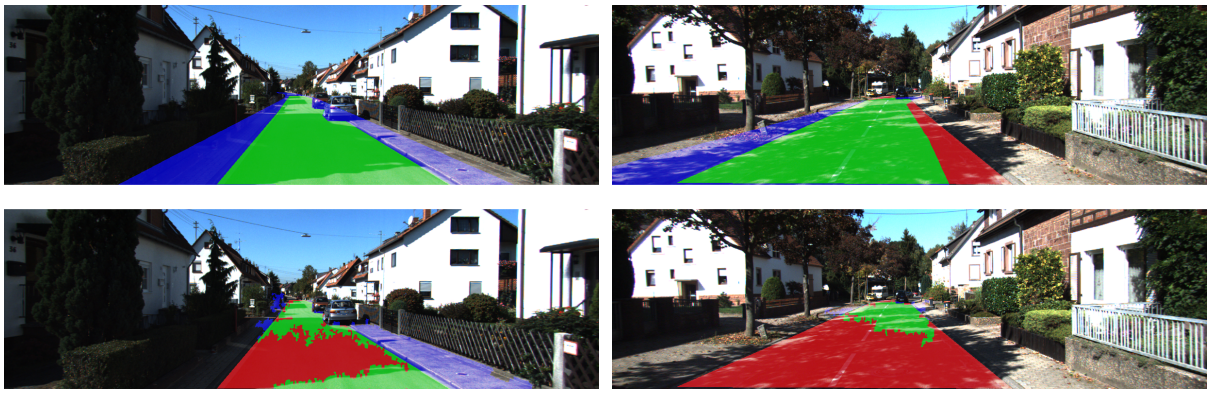
There are four steps in the algorithm: Projection of map data onto image plane (Map), label refinement using clustering (Cluster), restricting road candidate pixels using gPb (GPB) and finding the largest connected component (Component). Table 2.2 shows the results after each step. From this, we observe that the clustering step provides a boost in true positives. The number of false positives are reduced by computing the road candidate pixels and finding the largest connected component.

Step	F	PRE	REC
Map	78.93	74.97	83.34
Cluster	76.41	63.90	**95.01**
GPB	84.62	81.79	87.65
Component	**85.51**	**84.16**	86.90

Table 2.2: Quality of machine generated labels after each step of the proposed algorithm. For details see 2.6.2.

Effect of K

Figure 2.8 shows the performance of our automatic labeling approach with respect to the number of clusters K. We can see that for $K \geq 70$ the F-measure is very similar for all K. This indicates



(a) (b)

Figure 2.7: Failure cases for the automatic labeling of images. Top road displays results after initial map projection and bottom row shows results after our label refinement approach. These examples illustrate failure cases where our approach incorrectly removed most of the road pixels and kept the non-road pixels. The color coding is as follows: Green - True Positive, Red - False Negative, Blue - False Positive for road.

that after a specific value of K, which is in the order of 100, our approach is robust to the number of clusters.

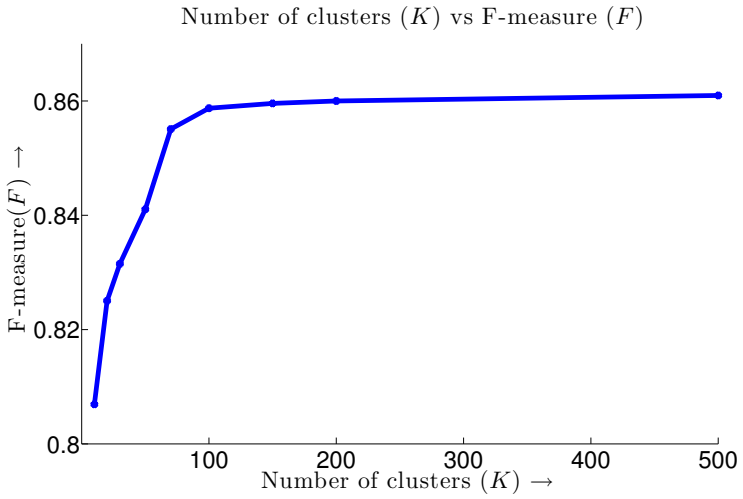


Figure 2.8: Effect of different values of K on our proposed automatic labeling approach.

Method	AP (↑)	MaxF (↑)	PRE (↑)	REC (↑)	FPR (↓)	FNR (↓)
Testing with Monocular Images						
Proposed - With Refinement	89.96	87.80	86.01	89.66	10.34	10.34
Proposed - Without Refinement	82.20	83.37	80.03	86.99	13.01	13.01
CN [3]	78.80	79.02	76.64	81.55	13.69	18.45
Testing with Other Sensors						
GRES3D+SELAS	86.86	85.09	82.27	88.10	10.46	11.90
RES3D+VELO [43]	78.34	86.58	82.63	90.92	10.43	9.08
Training with Human Annotations						
Fully Supervised (Proposed)	90.96	91.61	91.04	92.20	5.00	8.71

Table 2.3: Road detection performance of various methods on the *test set*. Note that, ↑ denotes higher is better and ↓ denotes lower is better.

2.6.3 Analyzing Performance of the Road Detection

We train the Fully Convolutional Network (FCN) using the images from *train set* and automatically generated road labels. We evaluate on the *test set*. During testing we only use monocular color images.

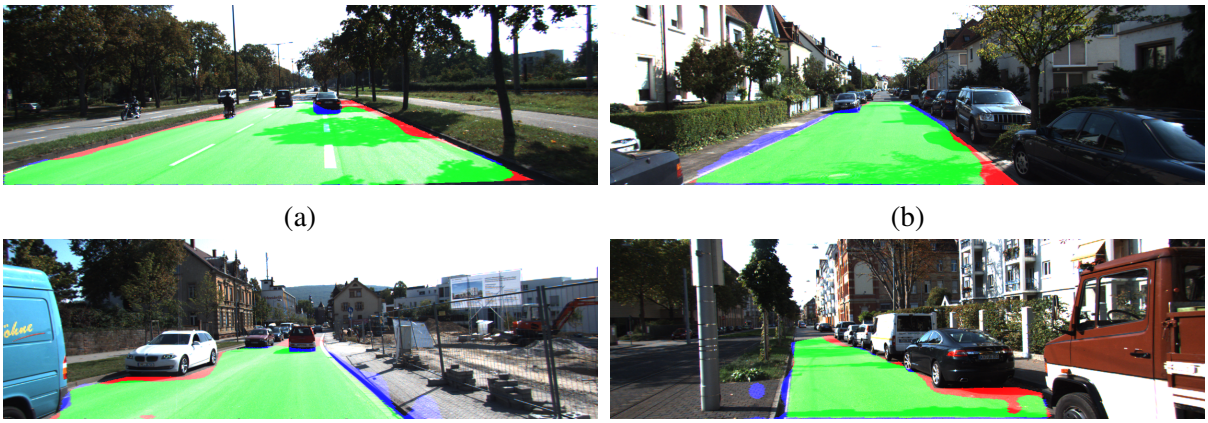
In Table 2.3, we can see that the FCN trained with refined labels (With Refinement) perform better than the one with labels we directly get from map projection (Without Refinement). This is because the refined labels are of better quality. We can also see that we outperform [3]. They

also train a CNN using machine generated labels and uses only monocular image for testing. However, their CNN is trained to predict the geometric structure (sky, vertical and horizontal) in the image which is them used to detect road. To build a road model for each image they assume that the middle bottom part of the image is road.

Next, we compare our approach with other methods which do not require human supervision but use other sensors. In Table 2.3, RES3D+VELO [43] uses laser data from Velodyne sensor for this task and GRE3D+SELAS uses a stereo pair. We achieve higher performance than these methods, thus establishing a new state of the art among approaches which do not require human supervision.

Finally, in Table 2.3 (Fully Supervised) we report the upper-bound performance of our classifier (FCN). In this case, the training is done with human annotated, ground truth labels. We can see that the FCN trained with automatically generated labels is able to achieve close to the upper-bound performance in terms of average precision and recall.

Qualitative Examples: Figure 2.9 shows some example road detection results on the *test set*. From these examples, We could see that the CNN trained using automatically labeled training data is able to detect roa



(a) (b)

(c) (d)

Figure 2.9: Qualitative Examples of the Detected Road using the CNN trained with automatically labeled data on the *test set*. The color coding is as follows: Green - True Positive, Red - False Negative, Blue - False Positive for road.

2.7 Summary

In this chapter, we presented a method to automatically annotate monocular images to train a classifier for road detection. We use OpenStreetMap, vehicle pose estimation and camera parameter to annotate the images with pixel level road/non-road labels. We show that we are able to generate good labeled data without any human annotation effort. However, we have difficulty in dealing with very strong shadows. This could be tackled by detecting shadow region in the image [25] and processing them separately.

We show that a CNN trained with automatically generated labeled data is similar to the same CNN trained with human labeled data. We also outperform the methods which use 3D data (laser or stereo) at test time. This is due to our ability to detect road at large distances which is not possible with 3D data because of noise at far away points.

Chapter 3

Tag-Supervised Semantic Segmentation

3.1 Introduction

The aim of semantic segmentation is to simultaneously recognize and segment various objects in images. It is one of the most challenging and fundamental task in computer vision. Recently, various convolutional neural network (CNN) based methods [8, 30] have been proposed to tackle this task. However, all of those require strong pixel-level human supervision in terms of pixel level boundary delineation of the boundaries of every object instance. The dependence on strongly labeled data makes scaling these algorithm difficult to a large number of classes.

Annotating images with pixel-level labels is an arduous and time consuming task. In [29] authors show that it takes 10-15x more time to generate pixel-level annotations for an image than only annotating the image with the name of objects present in it. They further show that workers on Amazon Mechanical Turk[1], a crowd sourcing platform, need to be trained to be able to generate pixel-level labels. Even with training, only 1 out of 3 workers were able to annotate images satisfactorily. Driven by these limitations, we are interested in addressing the following question: could we use a weaker form of supervision rather than full pixel-level supervision?

We use image-level tags as weak supervision to train a model for dense pixel-level prediction. Figure 3.1 shows the general idea of our approach. These tags provide the information about the presence and absence of classes in an image. Image-level tagging requires a fraction of time compared to full pixel-level annotations. However, tags only provide the name of the classes present in the image. They do not provide any information about the spatial extent or location of these classes in the image.

We use the multiple instance learning (MIL) framework [20] incorporate the image-level tag information. An image can be annotated with any number of classes which turns the problem into a multi-label classification problem at the image-level. Therefore, we propose to minimize a multi-label cross entropy loss instead of multi-class negative log likelihood. The traditional MIL framework does not take into account the structure of instances in a bag[2]. Whereas, an image is a structured object, with inherent label smoothness. To incorporate the smoothness property due to the image structure, we propose to create a similarity graph over the image and

[1] https://www.mturk.com/mturk/welcome

[2] An image is considered a bag of (super-)pixels instances

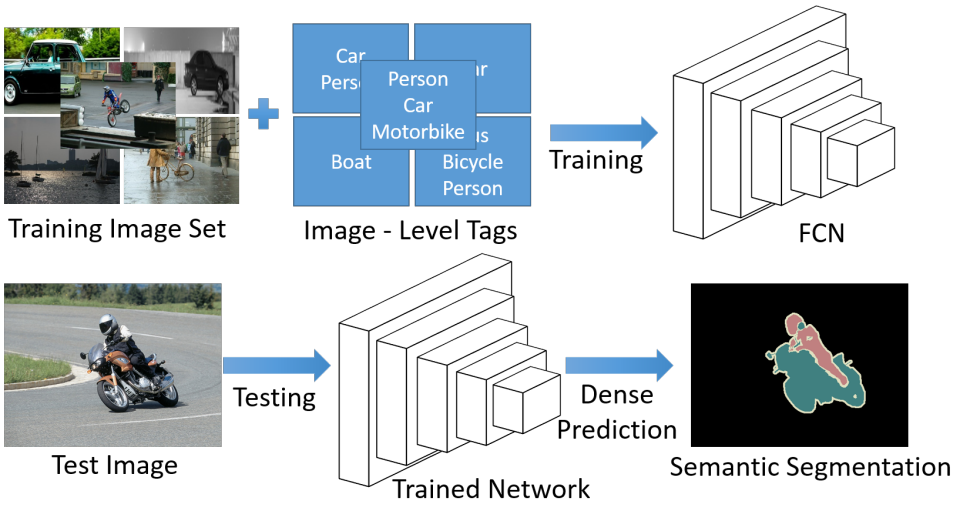


Figure 3.1: Tag-Supervised semantic segmentation approach overview.. We propose to train a CNN using image level tags. At test time the CNN predicts pixel-level semantic labels.

include a graph based regularization term with the cross-entropy loss. We formulate the training of a CNN as minimization of the proposed graph regularized multiple instance mulit-label (G-MIML) loss. We evaluate the proposed approach on PASCAL VOC 2012 [12] data-set and show that it outperforms previous state-of-art-methods.

3.2 Related Work

Full Supervision

Various top performing approaches for semantic segmentation train a CNN using strongly labeled data. Long et al. [30] first proposed fully convolutional network (FCN) by converting the fully connected layers for a classification network into convolutional layers. To increase the output resolution they use deconvolution layers and proposed a skip architecture which combines features from various layers in the network. In contrast to their approach, Chen at al. [8] proposed to increase the output resolution using dilated convolutions. To further improve the performance they proposed to post process the output using fully connected conditional random field (CRF) [23]. Later, Zheng et al. [53] proposed to jointly train the CRF and FCN by transforming the CRF inference procedure in to a recurrent neural network. This further improved the performance. However, the requirement of strong human supervision makes all of these methods unscalable. In this chapter, we train a CNN without using strong supervision. Therefore, making CNN training scalable for large number of classes and scenarios.

Tag Supervision

Vezhnevets et al. [47] first proposed to use image-level tags to train a pixel-level predictor. They used a Multiple Instance Learning (MIL) based loss to train a Random Forest. Whereas, we used

CNN which is a much better classifier. Also, for training they used strong human supervision for other pixel-level prediction task (geometric scene understanding).

Recently, Pinheiro et al. [38] and Pathak et al. [36] also used MIL framework for tag-supervised semantic segmentation. They use the Imagenet [40] data to train the classification layer of the network. Due to this their approach is only applicable if the target labels are a subset of imagenet labels. Whereas, our approach could be used with any target label set.

Similarly, Papandreou et al. [34] and Pathak et al. [35] have also proposed methods to train a FCN using tags. Both of these methods proposed an EM style iterative approach. In each iteration of their approach they first estimate the pixel labeling of each image using object size constraints and the tags. They then use these estimated pixel labels to train the FCN. In the experiments we show that our direct optimization method outperforms them.

Lastly, Pourian et al. [39] create a graph over the whole training data-set to infer the labels of each superpixel in the using the tags. At, test time for each superpixel in the test image, they transfer these inferred labels using an approach similar to weighted nearest neighbors. However, they require to store all the training data at the test time which makes their approach unsuitable for a very large amount of data. Whereas, in our case after training the CNN we do not need to store the training data.

MIL with Structured Data

Most classical MIL algorithms assume that the instances in the bag are independently and identically distributed. However, many times there exist rich dependency between the instances and ignoring them limits the performance of MIL algorithms. Therefore, Zhang et al. [52] proposed to include these rich dependence using a similarity graph. We use a similar approach as them but they train the classifier for bag level predictions whereas we train the classifier for instance level prediction.

3.3 Tag-Supervised Learning

Our goal is to train a dense pixel-level prediction model for semantic segmentation using only image-level tags. The tagged classes provide following constraints on the (super-)pixel labeling of an image. If an image is tagged with a certain class (e.g. dog), then at least one (super-)pixel in the image has to contain that class. All other classes except the tagged classes are assumed to be absent from the image. Therefore, none of the (super-)pixels in the image should be labeled with these classes. We use these constraints to propose a loss function for training a fully convolutional network.

Additionally, an image is a structured object with inherent label smoothness. The pixels with similar appearance in an image tend to have similar labels. Also, similar neighboring pixels tend to have similar labels. We use this smoothness assumption to regularize the training loss. We call the combined loss as graph regularized multiple instance multi-label (G-MIML) loss.

3.3.1 Objective Function

We over-segment an image into non-overlapping superpixels and use the standard assumption that all pixels in a superpixel belong to the same class. Let's denote an image $\mathbf{X}$ as a collection of n superpixels, $\mathbf{x}_i$, such that $\mathbf{X} = \bigcup_{i=1}^{n} \mathbf{x}_i$. Let's denote the number of predefined classes to label is C. We assume that each image $\mathbf{X}$ is associated with a binary label vector, $\mathbf{Y}$, of length C, such that $y_c = 1$ if the image is tagged with the class c, otherwise $y_c = 0$. We also assume that we are given a similarity graph with a $n \times n$ adjacency matrix $\mathbf{S}$. Let's denote $s_{i,j}$ as similarity between superpixels $\mathbf{x}_i$ and $\mathbf{x}_j$. The entries in the adjacency matrix $\mathbf{S}_{i,j} = 0$ if superpixels $\mathbf{x}_i$ and $\mathbf{x}_j$ are not connected and $\mathbf{S}_{i,j} = s_{i,j}$, if they are connected in the similarity graph.

Training a network for a particular task is formulated as optimizing its parameters, θ, by minimizing a training loss function over the whole training data-set, $\mathcal{L}(\theta)$, based on that task. Here, we describe the loss for one image. It could be extended to a whole data-set by summing over all the images. We define the G-MIML loss function (as consisting of two terms:

$$\mathcal{L}(\theta) = \underbrace{\mathcal{L}_{MIML}(\mathbf{Y};\theta)}_{\text{Tag based loss}} + \underbrace{\lambda \mathcal{L}_G(\mathbf{S};\theta)}_{\text{Graph based regularization}} \tag{3.1}$$

The first term is based on the tagged classes and the second term is based on the label smoothness assumption in an image.

3.3.2 Multiple Instance Multi-Label Loss

In the multiple instance learning (MIL) framework, the training set consists of labeled bags, each of which is a collection of unlabeled instances. Conventionally, MIL is used for binary classification with the assumption that a bag is labeled positive if at least one instance in the bag is positive, and a bag is negative if all the instances in it are negative. A classifier for predicting instance labels is trained by minimizing the bag-level negative binomial log-likelihood. The instance-level probabilities generated by the classifier are aggregated to produce bag-level probability.

We consider each image as a bag which is a collection of unlabeled superpixels. The constraints provided by image-level tags make MIL framework a natural choice for training the network. However, an image could be annotated by multiple classes due to we need to use a multi-label classification paradigm. This makes the standard binary MIL framework inapplicable. Therefore, we use the standard transformation of multi-label problem into multiple independent binary label problems [54].

Let's denote $p_{i,c}$ as the predicted probability of a class c at a superpixel $\mathbf{x}_i$. Also, let h denote the aggregation function to generate image-level probability, $p_c|\theta$ for a class c, such that $p_c|\theta = h(p_{1,c}|\theta, \cdots, p_{n,c}|\theta)$. We define multiple instance multi-label (MIML) loss, $\mathcal{L}_{MIML}(\mathbf{Y};\theta))$, as the average of the negative binomial log-likelihood for each individual class, such that:

$$\mathcal{L}_{MIML}(\mathbf{Y};\theta) = -\frac{1}{C}\sum_{c=1}^{C}\Big(y_c \log(p_c|\theta) + (1-y_c)\log(1-p_c|\theta)\Big) \tag{3.2}$$

Max Aggregation

A natural choice for the aggregation function, h, to generate image-level predictions is to use the max function, such that:

$$p_c|\theta = h(p_{1,c}|\theta, \cdots, p_{n,c}|\theta) = \max_{i \in 1,\cdots,n} p_{i,c}|\theta \tag{3.3}$$

This aggregation would encourage the network to increase the probability of the superpixel which is deemed most important for image-level classification. However, in the experiments we saw that using this aggregation leads to a the optimization getting stuck at predicting all of the superpixels as background.

Log-Sum-Exp Aggregation

Due to the difficulty with the max aggregation function we used a smooth and convex approximation of the max function called *Log-Sum-Exp* (LSE) [7, 38]:

$$p_c|\theta = h(p_{1,c}|\theta, \cdots, p_{n,c}|\theta) = \frac{1}{r} \log \left[\frac{1}{n} \sum_{i=1}^{n} \exp(r\, p_{i,c}|\theta) \right] \tag{3.4}$$

The hyper-parameter r controls the smoothness of the approximation. High values of r make the approximation closer to max function. Whereas, low values make the approximation closer to average function.

3.3.3 Similarity Graph based Regularization

The above MIL framework treats each superpixel in an image independently. Thus, it ignores the structure present in the image. We propose to regularize the MIML loss using a similarity graph to include the image structure information during training.

We assume that we are given a similarity graph $G = (\mathbf{V}, \mathbf{E})$ over each image. Each vertex v_i in the graph denotes a superpixel $\mathbf{x_i}$ and the weight $s_{i,j}$ on each edge denotes the similarity between the superpixels $\mathbf{x}_i$ and $\mathbf{x}_j$. We use this graph, G, to define the regularization term as:

$$\mathcal{L}_G(\mathbf{S}; \theta) = \frac{1}{C} \sum_{c=1}^{C} \left[\frac{1}{|\mathbf{E}|} \sum_{(i,j) \in \mathbf{E}} s_{i,j} \left(p_{i,c}|\theta - p_{j,c}|\theta \right)^2 \right] \tag{3.5}$$

This term have been used in various standard semi supervised learning algorithms [13]. This term encourages the network to predict similar label distribution at superpixels with high similarity. In experiments, we show that this graph based regularization helps to improve the performance of semantic labeling.

3.4 Implementation Details

3.4.1 Superpixels

We use [46] for superpixel segmentation of an image. We set the parameters such that we get an average of 300 superpixels per image.

3.4.2 Similarity Graph

We use the standard bag-of-words representation for superpixels to create the similarity graph. We extract dense SIFT features over all the training images using VLFeat[3]. We then use K-means to quantize these features into a D word visual codebook. We use normalized histogram of the visual codebook as the representation $\mathbf{x_i}$ for each superpixel. We define the visual similarity, $s_{i,j}$, between two superpixels $\mathbf{x}_i$ and $\mathbf{x}_j$ as:

$$s_{i,j} = e^{-\nabla(\mathbf{x}_i, \mathbf{x}_j)} \tag{3.6}$$

Where, $\nabla(\mathbf{x}_i, \mathbf{x}_j)$ denotes the distance between the superpixels i and j. The distance between the histogram representation is measured using Hellinger distance, which is defines as:

$$\nabla^2(\mathbf{x}_i, \mathbf{x}_j) = 1 - \sum_{d=1}^{D} \sqrt{\mathbf{x}_{i,d}\mathbf{x}_{j,d}} \tag{3.7}$$

This distance has been extensively used in image retrieval [6]. Let's denote $\mathcal{N}_k(\mathbf{x}_i)$ as the set of k-nearest neighbors in the visual space and $\mathcal{T}(\mathbf{x}_i)$ denotes the spatial neighbors of the superpixel i. We define the adjacency matrix $\mathbf{S}$ as:

$$\mathbf{S}_{i,j} = \begin{cases} s_{i,j} & \mathbf{x}_j \in T(\mathbf{x}_i) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

3.4.3 Network Architecture

We use the same architecture as mentioned in Section 2.5.1 with one change. Based on the observation in [8] that smaller field-of-view tend to give better result for weak supervision, we reduce the dilation factor of the first fully connected layer to 8 from 12. This change reduces the FOV of the network from 224×224 to 160×160.

Figure 3.2 shows the training setup for our approach. We take four inputs during training: a set of images and their associated tags, superpixel segmentation of and the superpixel-level similarity graph for each image. The FCN we use produces the output with a stride of 8 so we up-sample the output using bilinear interpolation. Then we do average pooling of the scores over each superpixel to get superpixel scores. We then use softmax function to convert the scores to probability. These probabilities are then used ti calculate both terms in the loss function. Note that at the test time, we only need the trained FCN.

[3] http://www.vlfeat.org/overview/dsift.html

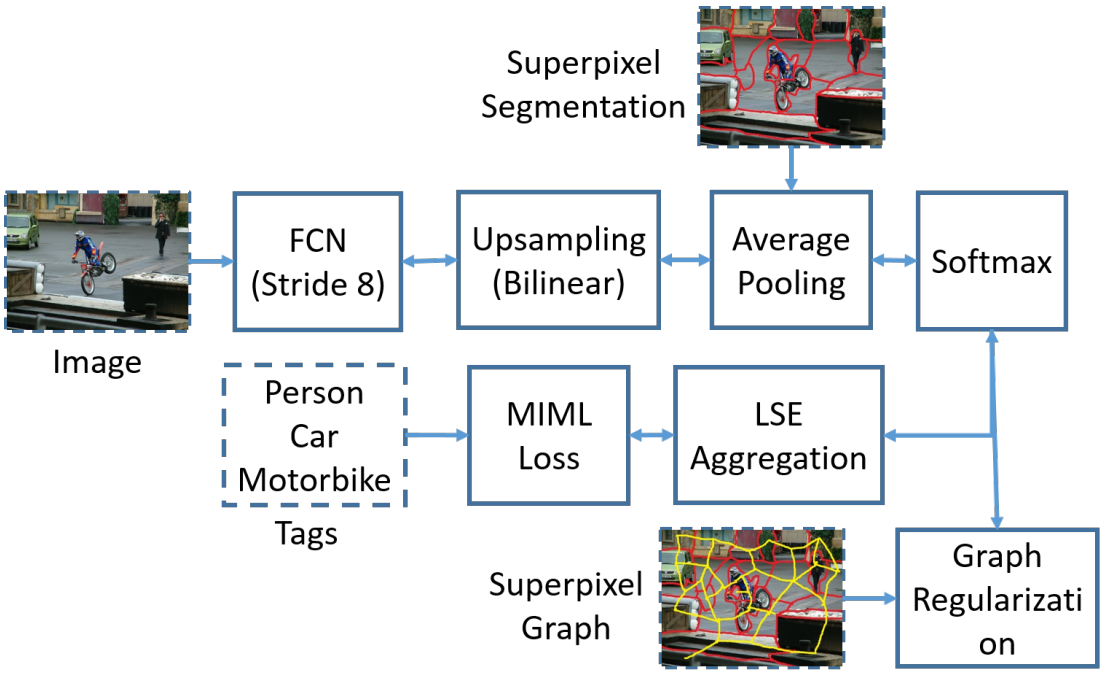


Figure 3.2: Proposed training setup for tag-supervised semantic segmentation.

3.4.4 Training

We use the publicly available code[4] by [19] for implementing and training the network. We initialize all the convolutional and fully connected layers using the imagenet trained network from [44]. The last classification layer is random initialized. We fine-tune the network with initial learning rate of 0.001 for 12 epocs and then with the learning rate of 0.0001 for 6 more epocs. We only fine-tune the fully connected layers and the classification layer. We use the batch size of 25. We re-size every image to 321×321. For data augmentation, we randomly mirror around the vertical axis.

3.4.5 Inference

At test time, we pass the full image through the trained network to predict the probability of all the classes at each location in the image. However, the network produces the output with a stride of 8. So, we use bi-linear interpolation to increase its resolution to the image size. We also do the CRF post processing as suggested in [8] with the default parameters present in the code from [23]. Note that at the test time we do not need any superpixel segmentation or superpixel graph.

[4] https://github.com/BVLC/caffe

3.5 Experiments

3.5.1 Dataset

We extensively evaluate our approach on the PASCAL VOC 2012 [12] data-set to evaluate our approach. We compare our approach to various previously proposed methods using tag supervision and pixel level supervision. The data-set contains a wide variety of images with labels for 20 object classes such as Car, Bus, Cat, Boat etc. It also had a background class which includes all the pixels not belonging to any of the 20 object categories. So, in total there are 21 labels.

We use the combination of images from the official *train set* of PASCAL VOC2012 segmentation task and the image annotated by [16] for training. We optimize the hyper-parameters, r and λ, on subset of 100 images from the *train set*. These images are assumed to contain pixel-level labels. In all the following experiments we set $r = 7$ and $\lambda = 10$. We report results on the *val set* and *test set*. We use the intersection-over-union criteria described in [12] to report performance for each label. To report aggregate performance we average results over all the labels.

3.5.2 Results

Validation Set

The *val set* of PASCAL VOC 2012 contains 1449 images. The ground truth pixel-level segmentation for each of these images are public available. We report performance by comparing our predictions with these available ground truth annotations.

Table 3.1 shows aggregate results of our approach and comparisons with other tag-supervised approaches. We report results with and without the post-processing CRF. Performance without the CRF post-processing shows the benefit of training with our novel loss function. We also include the data-sets used for training for each algorithm for a fair comparison. First, we see that use of the graph regularization, which incorporates the image structure, provides a significant boost in performance than only using tag information. Thus affirming that the regularization helps in reducing the ambiguity in the MIL problem. Next, we see that we are able to outperform all the previous approaches in both cases: with and without post-processing. Note that we outperform [38] and [36] which also use MIL framework based loss function. However, they use a much larger set of images from Imagenet to train the classification layer.

Table 3.2 shows results for each label present in PASCAL VOC 2012. First, we see that using the graph regularization improves performance for all the labels. Second, we observe that we achieve better performance than [35] on 13 out of 21 labels. Third, the bottom 5 performing labels are: Chair, Bicycle, Potted Plant, Sofa and Dining Table. All of these classes are either indoor objects or very intricate objects with fine details. These are also the worst performing categories for strongly supervised methods [8, 30].

Figure 3.3 shows the paired graph of accuracy difference for each image in the *valset* between the our method and CCNN [35]. We use the per-pixel accuracy measure to create this graph as intersection-over-union is a document level performance measure. We see that our method is able to improve the accuracy of around 60% of the images.

Figure 3.4 shows some qualitative examples of success results of our approach. We see that

Method	Data for training	w/o CRF	w/ CRF
Ours	VOC12	**34.3**	**37.6**
Ours w/o Graph Regularization	VOC12	29.8	32.3
CCNN [35]	VOC12	33.3	35.3
EM-Adapt[5] [34]	VOC12	32.0	33.8
MIL-FCN [36]	VOC12 + Imagenet	24.9	–
MIL Base w/ ILP [38]	Imagenet	32.6	–

Table 3.1: Comparisons of various tag-supervised approaches on the *val set* using Aggregate Results.

Label	CCNN [35]		Ours w/o Graph		Ours	
	w/o CRF	w/ CRF	w/o CRF	w/ CRF	w/o CRF	w/ CRF
Background	66.3	68.5	66.8	70.1	71.5	**74.9**
Airplane	24.6	25.5	28.4	30.8	31.5	**34.4**
Bicycle	17.2	18.0	15.8	18.0	16.6	**18.5**
Bird	24.3	25.4	26.2	31.4	34.4	**42.2**
Boat	19.5	20.2	23.4	26.6	24.0	**26.8**
Bottle	34.4	**36.3**	21.5	23.4	26.7	31.0
Bus	45.6	**46.8**	38.8	38.1	43.8	43.0
Car	44.3	**47.1**	35.2	39.0	40.2	44.9
Cat	44.7	48.0	44.1	47.3	51.2	**54.8**
Chair	14.4	**15.8**	11.4	12.8	13.0	15.0
Cow	33.8	**37.9**	27.3	30.1	31.3	34.5
Dining Table	21.4	21.0	22.8	25.3	24.1	**26.4**
Dog	40.8	44.5	36.2	39.3	43.0	**46.7**
Horse	31.6	34.5	28.4	31.4	34.2	**36.6**
Motorbike	42.8	46.2	37.6	41.7	43.5	**48.7**
Person	39.1	40.7	33.1	35.5	43.3	**48.6**
Potted Plant	28.8	**30.4**	18.3	21.4	21.9	25.6
Sheep	32.3	36.3	30.4	33.6	39.0	**44.1**
Sofa	21.5	22.2	19.6	20.9	22.5	**25.0**
Train	37.4	**38.8**	38.1	37.6	38.4	37.9
TV/Monitor	34.4	**36.9**	22.9	24.4	27.3	29.6

Table 3.2: Performance of each label in the *val set*.

the graph regularization term helps in recovering the shape of the objects. Figure 3.5 shows some qualitative examples of failure cases of our approach. These confirm the observations from quantitative evaluation. The first two rows show that our algorithm has difficulty in segmenting

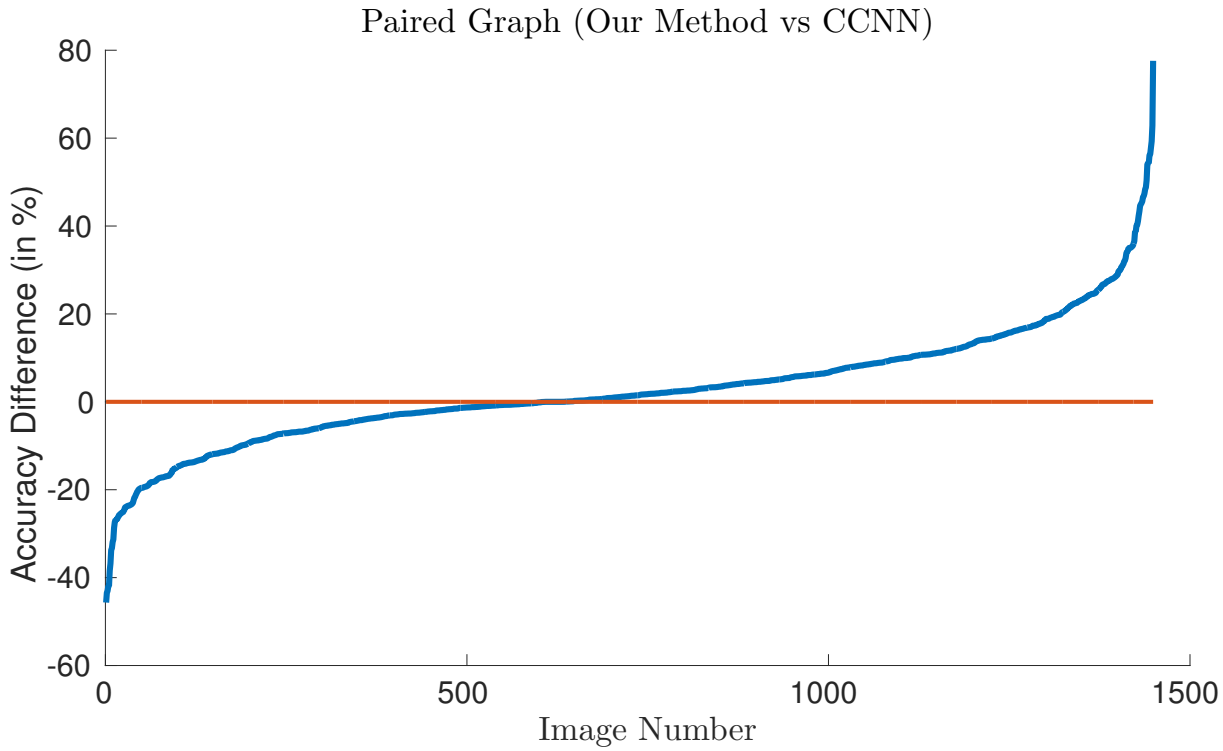


Figure 3.3: Paired Graph showing accuracy difference for each image in the *val set* between our method and [35].

indoor objects. The third row shows the failure case when an object comes with the same background multiple times. In this case the algorithm labels the background as part of the object. The fourth rows shows an example case with the difficulty to segment an intricate object.

Test Set

The *test set* of PASCAL VOC 2012 contains 1456 images. The ground truth pixel-level annotations for each of these images are private and one has to submit the results on the evaluation server [6]. We report results on this set in Table 3.3. All of the reported results except [38] are after CRF post-processing. We again outperform all previously proposed tag-supervised approaches. We also compare our tag-supervised approach with strongly supervised approach which uses the same *Deeplab-LargeFOV* network. We observe that there is still a large gap between the performance of strongly-supervised methods and tag-supervised methods.

3.6 Summary

In this chapter, we presented a method for semantic segmentation of images without using any pixel-level annotations. We use human labeled image-level tags for training a CNN. Tags could

[6] http://host.robots.ox.ac.uk:8080/

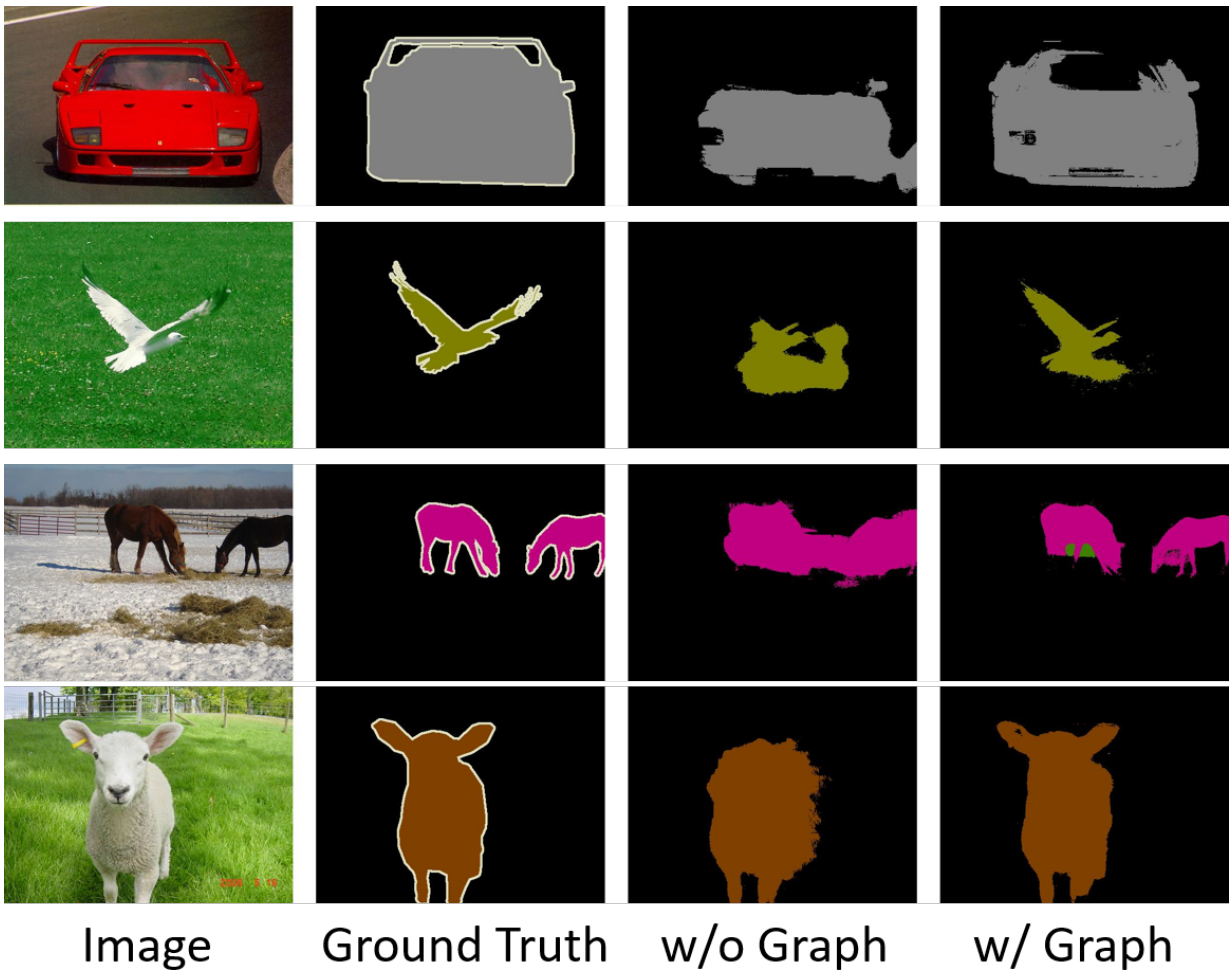


Figure 3.4: Example successful cases for our tag-supervised approach. We could see that with graph regularization we are able to get the shape of objects correctly.

Method	Data for training	IoU
Ours	VOC12	**39.3**
CCNN [35]	VOC12	35.6
EM-Adapt [34]	VOC12	35.2
MIL Base w/ ILP [38]	Imagenet	35.8
Strong Supervision		
Deeplab-LargeFOV-CRF [8]	VOC12	66.4

Table 3.3: Comparisons on *test set* using Aggregate Results.

collected for a fraction of time as compared to pixel labels. Thus, allowing to collect large amount of labeled data in a short span of time.

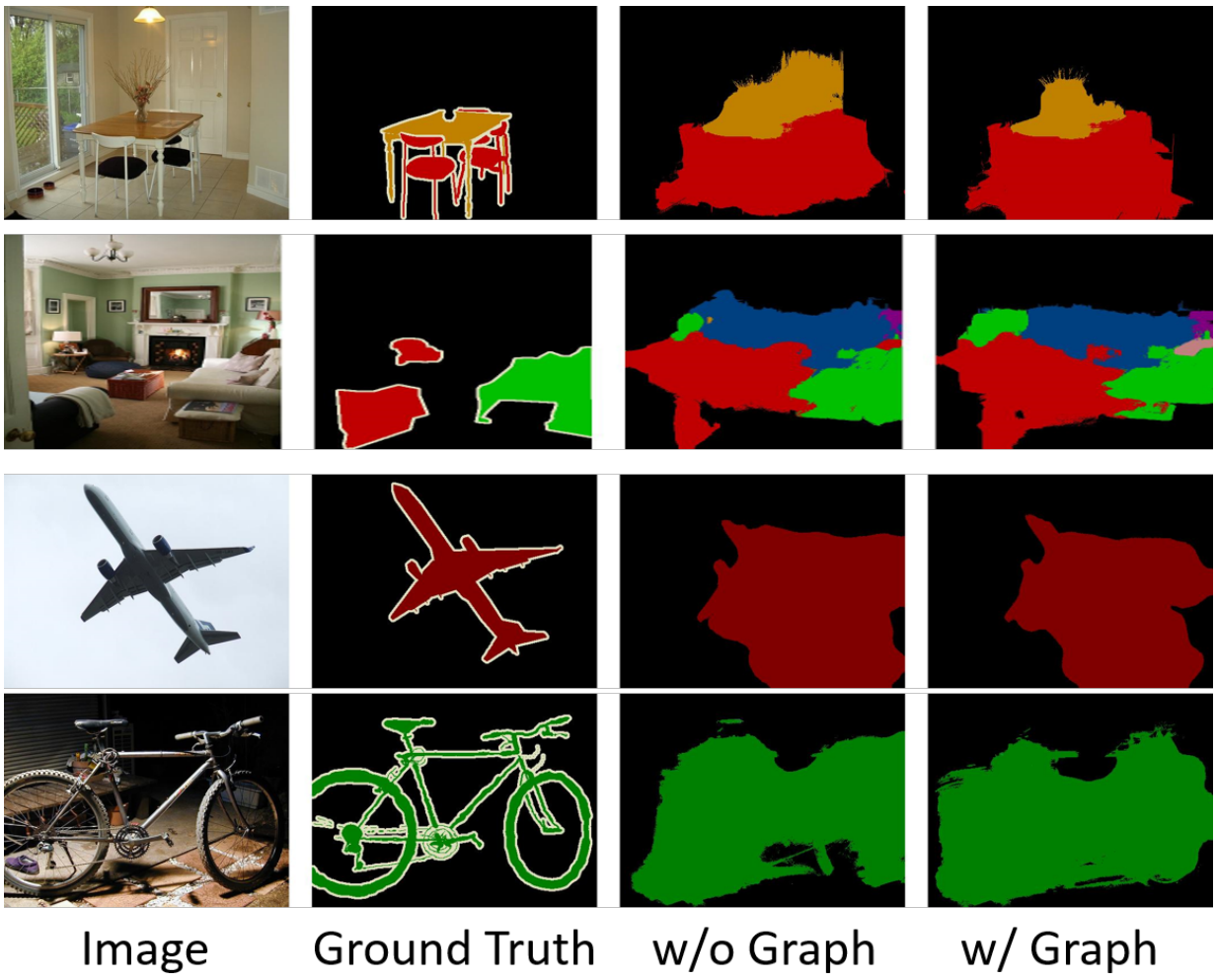


Figure 3.5: Example failure cases for our tag-supervised approach. See text for details.

We train a CNN using a graph-regularized multi-label multiple instance (G-MIML) loss. The MIML part encodes information from images tags and the graph incorporates the image structure. We observe that, including the graph in multiple instance learning framework improves the coverage of the object. We show that our method is able to outperform previously proposed tag-supervised method.

Our method works well on outdoor scenes because they are less cluttered, thus recognizing the pixels belonging to objects easily. However, our method method struggles in cluttered indoor scenes. The indoor scenes are in general harder to segment than outdoor scenes as evident by the results of strongly supervised methods [8, 30].

We also observe that our method is able to segment the classes which appear with diverse backgrounds more effectively than the classes which almost always appear with same background. This is due to the reason that if objects of a particular class appear with different backgrounds than discriminating between the object and background pixels is straightforward. Whereas, when an object almost always appears with the same background then, discriminating

the object and background pixels by using only image tags is impossible. We think that this could be tackled by incorporating objectness [1] measure into the training loss.

Chapter 4

Conclusion

4.1 Summary

In this thesis, we presented methods to reduce the strong human supervision requirement for training pixel-level prediction CNN. The proposed methods reduces human labeling effort. Thus, making the training of CNNs scalable and cost-effective. In particular, we tackled the problem of road detection and semantic segmentation.

For road detection, we presented a map-supervised, monocular image-based drivable road area detection system. It can automatically generate training labels for drivable road recognition using the noisy data from publicly available OpenStreetMap and other localization sensors on the vehicle. We show that a CNN trained using this automatically annotated data has similar performance as a CNN trained using human labeled data.

For semantic segmentation, we presented a novel graph regularized multiple instance multi-label (G-MIML) loss to train a dense pixel-level prediction CNN using image-level tags. We show that use the graph regularization to incorporate the image structure helps improve the performance. We also show that the proposed approach is able to outperform all the previous tag-supervised methods for this task on PASCAL VOC 2012 [12].

4.2 Future Work

In this thesis, we proposed methods to reduce human labeling effort for pixel-level prediction tasks. However, these approaches still have inferior performance than the strongly supervised methods. Therefore, we conclude by suggesting some ideas for further improvement of the methods described in this thesis.

For road detection, we would like to use the temporal information present in videos to improve the labeling step. We could use the motion information present in the videos to remove the false positives on the moving vehicles. This would be very useful in cluttered scenes where large part of road is occluded by moving vehicles. We would also like to extend our system for extreme weather conditions such as rain and snow, which are currently not handled by our system. For this, we suggest to use 3D features from videos and CNN features to distinguish between the road and background pixels.

For semantic segmentation, we would like to extend the graph to an inter-bag graph. This would further regularize the MIL problem effectively. We would also like to include other weak forms of supervision such as scribbles [28] by including a semi-supervised loss term into the overall loss function. This would help in improving the extent estimation of objects. To further improve the performance, we would like to extend our system to include the semi-supervised setting where we also have access to a small number of strongly-labeled images or active learning setting in which we get strong labels for pixels which will provide the maximum information to the CNN.

Bibliography

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *CVPR*, 2010.

[2] J. M. Alvarez and A.M. Lopez. Road detection based on illuminant invariance. *ITS*, 2011.

[3] Jose M Alvarez, Theo Gevers, Yann LeCun, and Antonio M Lopez. Road scene segmentation from a single image. In *ECCV*. 2012.

[4] Jose M. Alvarez, A M. Lopez, T. Gevers, and F. Lumbreras. Combining priors, appearance, and context for road detection. *ITS*, 2014.

[5] Jose M Alvarez, Mathieu Salzmann, and Nick Barnes. Large-scale semantic co-labeling of image sets. In *IEEE WACV*, 2014.

[6] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[7] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. 2004.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[9] Hsu-Yung Cheng, Bor-Shenn Jeng, Pei-Ting Tseng, and Kuo-Chin Fan. Lane detection with moving vehicles in the traffic scenes. *ITS*, 2006.

[10] P. Dollar and C.L. Zitnick. Fast edge detection using structured forests. *TPAMI*, 2015.

[11] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[13] Rob Fergus, Yair Weiss, and Antonio Torralba. Semi-supervised learning in gigantic image collections. In *NIPS*, 2009.

[14] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *ITSC*, 2013.

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[16] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[17] Derek Hoiem, Alexei Efros, Martial Hebert, et al. Geometric context from a single image. In *ICCV*, 2005.

[18] Kazuki Irie and Masahiro Tomono. Road recognition from a single image using prior information. In *IROS*, 2013.

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[20] James D Keeler, David E Rumelhart, and Wee-Kheng Leow. *Integrated Segmentation and Recognition of Hand-Printed Numerals.* 1991.

[21] Hui Kong, J.-Y. Audibert, and J. Ponce. General road detection from a single image. *Image Processing, IEEE Transactions on*, 2010.

[22] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*. 2011.

[23] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[25] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, 2010.

[26] Dan Levi, Noa Garnett, Ethan Fetaya, and Israel Herzlyia. Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In *WACV*, 2015.

[27] David Lieb, Andrew Lookingbill, and Sebastian Thrun. Adaptive road following using self-supervised learning and reverse optical flow. In *RSS*, 2005.

[28] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *CVPR*, 2016.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[31] Volodymyr Mnih and Geoffrey Hinton. Learning to label aerial images from noisy data. In *ICML*, 2012.

[32] Rahul Mohan. Deep deconvolutional networks for scene parsing. In *arXiv:1411.4101*, 2014.

[33] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *ECCV*, 2010.

[34] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *ICCV*, 2015.

[35] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural

networks for weakly supervised segmentation. In *ICCV*, 2015.

[36] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. *ICLR Workshops*, 2015.

[37] Lina Maria Paz, Pedro Piniés, and Paul Newman. A Variational Approach to Online Road and Path Segmentation with Monocular Vision. In *ICRA*, 2015.

[38] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[39] Niloufar Pourian, S Karthikeyan, and BS Manjunath. Weakly supervised graph based semantic segmentation by learning communities of image-parts. In *ICCV*, 2015.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[41] S. Salvador and P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *ICTAI*, 2004.

[42] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, 2014.

[43] Patrick Yuri Shinzato, Denis Fernando Wolf, and Christoph Stiller. Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion. In *IV*, 2014.

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[45] C. Tan, Tsai Hong, T. Chang, and M. Shneier. Color model-based real-time learning for road following. In *ITSC*, 2006.

[46] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *ECCV*, 2012.

[47] Alexander Vezhnevets and Joachim M Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *CVPR*, 2010.

[48] Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *CVPR*, 2015.

[49] Liang Xiao, Bin Dai, Daxue Liu, Tingbo Hu, and Tao Wu. Crf based road detection with multi-sensor fusion. In *IV*, 2015.

[50] Jian Yao, Srikumar Ramalingam, Yuichi Taguchi, Yohei Miki, and Raquel Urtasun. Estimating drivable collision-free space from monocular video. In *WACV*, 2015.

[51] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2015.

[52] Dan Zhang, Yan Liu, Luo Si, Jian Zhang, and Richard D Lawrence. Multiple instance learning on structured data. In *NIPS*, 2011.

[53] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *CVPR*, 2015.

[54] Zhi-Hua Zhou and Min-Ling Zhang. Multi-instance multi-label learning with application to scene classification. In *NIPS*, 2006.