Efficient Annotation and Knowledge Distillation for Semantic Segmentation

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Tejaswi Kasarla
201550846
kasarla.tejaswi@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2019

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Efficient Annotation and Knowledge Distillation for Semantic Segmentation**" by **Tejaswi Kasarla**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. Vineeth N. Balasubramanian
Date	Adviser: Prof. C.V. Jawahar

To *amma* and *nanna*

Acknowledgments

This thesis is the outcome of support, guidance and encouragement I received from several people.

Firstly, I would like to thank my advisers, Prof. C.V. Jawahar and Dr. Vineeth N. Balasubramanian for their constant guidance and support throughout the course of my Master's degree. They have given me substantial room and flexibility to improve myself as a student and a researcher.

I would to extend my thanks to the Research and Technology Center, Bosch India, for collaborating with me on the work I have done in this thesis. In particular, I am very thankful to Dr. Guruprasad Hegde and Koustav Mullick for being wonderful mentors during my time at Bosch.

I am very fortunate to learn from Anand Mishra and Nagendar during the time I worked with them. I would undoubtedly thank Abhishek for being a wonderful friend during all of my MS, and for motivating me to stay on-track in moments of self-doubt. I am grateful to Sahil, Vamsi, Ayushi, Isha for being the great friends that they are and for all the fun times that we had. I am also very thankful all other lab-mates and friends at CVIT for being a crucial part of my professional and personal life.

Most importantly, I would like to thank my family for their unending support throughout the pursuit of my goals. Their love and encouragement is the reason I am able to pursue research with such passion and enthusiasm. Thank you amma, nanna and Mukthi.

Abstract

Scene understanding is a very crucial task in computer vision. With the recent advances in research of autonomous driving, road scene segmentation has become a very important problem to solve. Few exclusive datasets were also proposed to address this research problem. Advances in other methods of deep learning gave rise to deeper and computationally heavy models and were subsequently adapted to solve the problem of semantic segmentation. In general, larger datasets and heavier models are a trademark of segmentation problems. This thesis is a direction to propose methods to reduce annotation effort of datasets and computational efforts of models in semantic segmentation.

In the first part of the thesis, we introduce the general sub-domain of semantic segmentation and explain the challenges of dataset annotation effort. We also explain a sub-field of machine learning called *active learning* which relies on a smart selection of data to learn from, thereby performing better with lesser training. Since obtaining labeled data for semantic segmentation is extremely expensive, active learning can play a huge role in reducing the effort of obtaining that labeled data. We also discuss the computational complexity of the training methods and give an introduction to a method called *knowledge distillation* which relies on transferring the knowledge from complex objective functions of deeper and heavier models to improve the performance of a simpler model. Thus, we will be able to deploy better performing simpler models on computationally restrained systems such as self-driving cars or mobile phones.

The second part is a detailed work on active learning in which we propose a Region-Based Active Learning method to efficiently obtain labelled annotations for semantic segmentation. The advantages of this method include: (a) using lesser labeled data to train a semantic segmentation model (b) proposing efficient models to obtaining the labeled data thereby reducing the annotation effort. (c) transferring model trained on one dataset to another unlabeled dataset with minimal annotation cost which significantly improves the accuracy. Studies on Cityscapes and Mapillary datasets show that this method is effective to reduce the dataset annotation cost.

The third part is the work on knowledge distillation and we propose a method to improve the performance of faster segmentation models, which usually suffer from low accuracies due to model compression. This method is based on distilling the knowledge during the training phase from models with high performance to models with low performance through a teacher-student type of architecture. The knowledge distillation allows the student to mimic the performance of the teacher network, thereby improving its performance during inference. We quantitatively and qualitatively validate the results on three different networks for Cityscapes dataset.

Overall, in this thesis, we propose methods to reduce the bottlenecks of semantic segmentation tasks, specifically related to road scene understanding: the high cost of obtaining labeled data and the high computational requirement for better performance. We show that smart annotation of partial regions in data can reduce the effort of obtaining labels. We also show that extra supervision from a better performing model can improve the accuracies for computationally faster models which are useful for real-time deployment.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep Learning has ubiquitously changed the field of statistical machine learning and artificial intelligence. The advancements in semiconductor technology have allowed to build bigger and better computational systems. This unprecedented growth in computational power allowed us to leverage vastly available data to solve long-standing problems in visual understanding. With access to large amounts of data and the ability to harness computational power faster graphical processing units, deep learning has been at the forefront of revolutionizing the field of artificial intelligence.

The success of deep learning was first realized when the system was able to solve the image recognition problem better than its predecessors with a large margin. Researchers around the world spent their efforts to develop better and deeper neural networks to solve various other problems relevant to computer vision. These networks resulted in state-of-art accuracies in their relevant tasks such as image recognition, object detection, semantic segmentation, and instance segmentation. As the networks became complex and deeper, they required more data and more computational power. With low resource systems such as mobile phones and self-driving cars, deploying such computationally intensive models is infeasible. Also, as the network complexity increases, it requires larger amounts of data to learn and generalize better. All these challenges need to be addressed to develop better systems which are computationally efficient.

In this thesis, we present a set of solutions which address the data and the computational bottleneck of deep learning models. These proposed solutions try to solve the challenges pertaining to a specific sub-domain of computer vision which is the most affected by these bottlenecks – semantic segmentation. Our first work gives a solution to the data bottleneck by addressing a way to reduce the data annotation cost. In the second work, we address the computational bottleneck of semantic segmentation models for real-time deployment on necessary devices such as self-driving cars and mobile phones. We present a solution to improve the performance of small, compact models deployed on such devices.

1.1 Motivation

The need for large, annotated datasets has seen increasing significance with the growing capabilities of deep neural network models in solving real-world vision tasks, and their subsequent absorption in

usable technologies. Autonomous systems, such as self-driving vehicles, have further underscored this need, in order to scale to various geographies and settings around the world. Existing efforts to create large-scale vision datasets have largely been localized, and there is a need to provide better methods to create datasets more efficiently. This work is an effort in this direction.

Among vision tasks, semantic segmentation has recently attracted a lot of attention, where the objective is to classify each pixel into its corresponding semantic class. Fully supervised methods for this problem require annotation of all pixels of all given images/videos. This is a tedious task, and requires a huge amount of time and need human effort [57]. While semi-supervised methods (where part of the data used to train a machine learning model is unlabeled) can be used to offset the annotation load, fully supervised methods have continued to maintain state-of-the-art performance on tasks, especially in automation. Our objective in this work is to reduce the annotation load for semantic segmentation tasks.

Active learning methods have been known in machine learning for a couple of decades now. They operate in a setting where given a learned model, a learning algorithm chooses data points intelligently (using the given model) and subsequently queries an oracle for the labels of the chosen data points. The model is then updated using the newly obtained labeled data, with a view to increasing the model's performance on unseen data. There have been limited related efforts in the past on active learning in semantic segmentation [69, 70]; however, their focus was different and algorithm-driven and did not show tangible improvements in dataset development for real-world tasks. We instead focus on using active learning to contribute to dataset development for tasks based on contemporary datasets such as Cityscapes and Mapillary, with the main objective to reduce annotation cost on unlabeled data.

While data annotation cost can be reduced there is still a very high computational cost required to run better performing state-of-the-art models on autonomous systems such and self-driving cars and is practically infeasible to deploy on mobile phones. This calls for compressed models for segmentation. While compressed models are fast and fit on a constrained computation system, their performance is oftentimes very less compared to the state-of-the-art models. Our objective in this work is to improve the semantic segmentation accuracies for the compressed models.

A teacher-student learning was introduced in early 2014 by Ba and Caruna [3] which leverages deeper network's supervision to learn similar complex functions in shallow networks. Knowledge distillation introduced by Hinton et.al., [28] is a special case of teacher-student learning. In this, the softmax of a deeper network (teacher) is used along with labeled data to train a shallow network (student). This improves the performance of the student network when compared to training it alone with the labeled data. This is because the student network learns to mimic the complex objective function learnt by the teacher network. There has been an adaptation of this concept to image classification and object detection. We focus on using knowledge distillation to shallow and fast segmentation networks in improving their performance.

1.2 Problem Definition

We will be focusing our work in this thesis on reducing the annotation cost and improving model performance for semantic segmentation. Particularly, we discuss two problems in this domain.

Active Learning: The aim of this work is to reduce the high annotation cost for datasets for semantic segmentation. Dense pixel-wise annotations for semantic segmentation are very expensive to obtain, often requiring 100s of hours of human annotation. This is prohibitive when we need to add new data of cities for autonomous driving situations. Instead, by leveraging the existing models and with very little additional annotation on new data, getting almost similar performance as full supervision with whole dataset annotation is quit. However, such data needs to annotated in a smarter and intelligent way and active learning plays a huge role in such a scenario.

Knowledge Distillation: The aim of this work is to improve the performance of fast and real-time networks for semantic segmentation. These real-time and smaller networks are very useful to deploy on computationally constrained systems, like self-driving cars and mobile phones. More often than not, the real-time inference and reduction in model size reduce the performance of such networks significantly. Adapting knowledge distillation to such models is instrumental in improving their performance with no additional cost during inference.

1.3 Contribution

The contribution of this thesis to semantic segmentation is two-fold:

- Reduction of annotation effort for datasets through *active learning* without compromising much on the performance, discussed in chapter 3.
- Improvement of the performance of segmentation models during training through *knowledge distillation* without increasing the time during inference, which was presented in chapter 4.

In the first work, we propose an entropy-based active learning approach for efficient labeling in semantic segmentation. While entropy has been extensively used for active learning in the past [31] for classification [77], we use this exclusively for semantic segmentation in this work. Furthermore, we propose a region-based active learning methodology, where annotation at the superpixel level in images, along with the use of fully connected Conditional Random Fields (CRF) [38] for label propagation, provides significant benefits in reducing annotation cost. We show our results on datasets for autonomous driving, namely, Cityscapes and Mapillary. We show the results of our methods on cityscapes dataset. We then transfer the learned model on cityscapes dataset to mapillary dataset with very little additional labelling.

In the second work, we propose an ensemble based knowledge distillation approach to real-time and fast segmentation models to achieve improvement of their performance. We leverage the knowledge learnt in deeper and complex networks, which are better performing, and distill that knowledge to the fast segmentation networks. We show our results on Cityscapes dataset and prove the utility of this method on two small sized, real-time segmentation networks.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 presents an overview of supervised learning for semantic segmentation. It discusses the evaluation strategy for semantic segmentation. The relevant datasets and methods of annotation used for segmentation are introduced. It also introduces the concepts of active learning and knowledge distillation and discusses the relevant literature pertaining to both.

- Chapter 3 details several active learning strategies in the context of semantic segmentation. It proposes various novel active learning algorithms for segmentation in addition to presenting the existing query selection methods. The results are presented for two major datasets for self-driving. We also compare our method to another popular active learning strategy, which used monte-carlo dropout as an active learning metric. We discuss the benefits and merits of our method compared to the other one. We also briefly discuss our proposed user annotation strategies.

- Chapter 4 describes the research focusing on improving the performance of smaller models through *knowledge distillation*. We discuss existing relevant literature for knowledge distillation which proved successful in problems such as image annotation, object detection. We introduce our proposed ensemble teacher-student training method and discuss its contributions. We present our results on a dataset for autonomous driving and show its adaptability to various real-time segmentation networks.

- We finally end the thesis by summarizing it, presenting conclusions and discussing possible future directions.

Chapter 2

Background

In this chapter, we address the problem of scene understanding and the challenges it presents. Scene understanding is one of the very fundamental tasks of computer vision that deals with knowing what is present in the scene and where it is present. Algorithms for this task are required to understand the semantics and the contextual information in the scene. The work in this thesis discusses semantic segmentation in particular, with a focus on data and computational constraints of the task. In section 2.1 we discuss the problem of semantic segmentation and also briefly discuss the supervised deep learning methods used to solve the task and explain a popular real-time segmentation network, ICNet. Section 2.2 discusses how to evaluate the performance of a semantic segmentation task, and provides a detailed introduction to the IOU metric. Existing datasets for segmentation are described in section 2.3. This section also includes the standard annotation methods used to densely annotate an image for semantic segmentation. We show how expensive the annotation is through such methods, and theorize it will be beneficial to intelligently annotate the image pixels partially. *Active learning*, described in section 2.4 is a sub-domain of machine learning which is a way to reduce the annotation cost for various machine learning tasks. A major constraint to deploying segmentation systems onboard a self-driving car or a mobile phone is the size of the system. Though real-time segmentation systems are introduced lately, they usually suffer from reduced performance. We discuss *knowledge distillation* in section 2.5, a teacher-student model training method to transfer or distill the knowledge from a larger machine learning model to a smaller model. The smaller model distilled with the knowledge of a larger model will have better performance than the stand-alone smaller model trained without distillation. Subsequently, active learning and knowledge distillation were adapted to semantic segmentation in chapters 3 and 4.

2.1 Supervised Learning for Semantic Segmentation

The task of semantic segmentation can be summarized as "classifying each pixel of the given image to its corresponding semantic class". While this sounds very similar to image classification, it is more complex the this problem. The inherent spatial correlation of pixels in the image is what makes the problem hard to solve. Some of the earlier works on this problem predict the class probabilities of a pixel in an image patch using *random forest* [7][64] and *boosting* [39]. These noisy pixel-level probabilities

can then be smoothed out using the semantic correlation with its neighboring pixel. This was explained in the use of *unary* and *pairwise* potential terms in Conditional Random Fields (CRF)S and subsequently in fully-connected CRFs [38] which improved the accuracy of the predictions.

After the success of deep neural networks on the ILSVRC challenge [54] for the image classification task, Shelhamer et. al. [45] proposed the first end-to-end neural network architecture for semantic segmentation. This architecture, called the Fully Convolutional Network (FCN), captures the low-level and high-level information in the image. It had a VGGNet base architecture [65] of image classification network, adapted to predict the segmentation map. This was accomplished by adding a deconvolutional layer through bilinear interpolation after the convolutional layers instead of the fully connected layers. This ensured the preservation of spatial information. They presented their results on the Pascal VOC dataset [20], which showed a high improvement over previously proposed conventional methods.

After the visible success of FCN for segmentation, several other methods were proposed building upon it. SegNet by Kendall et.al., [36] proposes learning of deconvolutional layers along with the weights of convolutional layers. DeepLab [14] combines the semantic segmentation architecture with CRF inference to improve its performance. With also the improvement in architectures for image classification such as ResNet [27] and others, same things have been adapted to semantic segmentation. Other recent networks are PSPNet [79], DeepLab V3 [16], which have much deeper and complex architectures.

Next, we will explain in detail the base neural network architecture and strategies like crf used in our work.

2.1.1 ICNet

ICNet is a semantic segmentation architecture which has decent prediction accuracy and is also fast. It also processes the input images at high-resolution, which is the desired requirement for real-time segmentation architectures.

It processes low-resolution images through full semantic prediction architecture. Through this, we get a coarse prediction map. Then these predictions are gradually fused to mid-resolution and high-resolution version of this image through their proposed cascade feature fusion (CFF) and cascade label guidance architectures. Since both the mid and high-resolution inputs are computationally heavy, they go through only partial semantic prediction and then the low-resolution prediction is interpolated through the CFF unit to fuse with the mid-resolution prediction. A similar CFF unit is applied to high-resolution prediction fusion with mid-resolution prediction. This reduces the computational time thus allowing the network to run at real-time.

2.1.2 Fully-connected Conditional Random Fields

Traditionally, the image segmentation problem was posed as a graphical inference in a Conditional Random Field CRF [40] defined over pixels or image patches. A unary potential term models the uncertainty of individual pixels or image patches and a pairwise potential term models the relationship between the neighboring pixels or patches. To model long-range connection instead of just adjacent

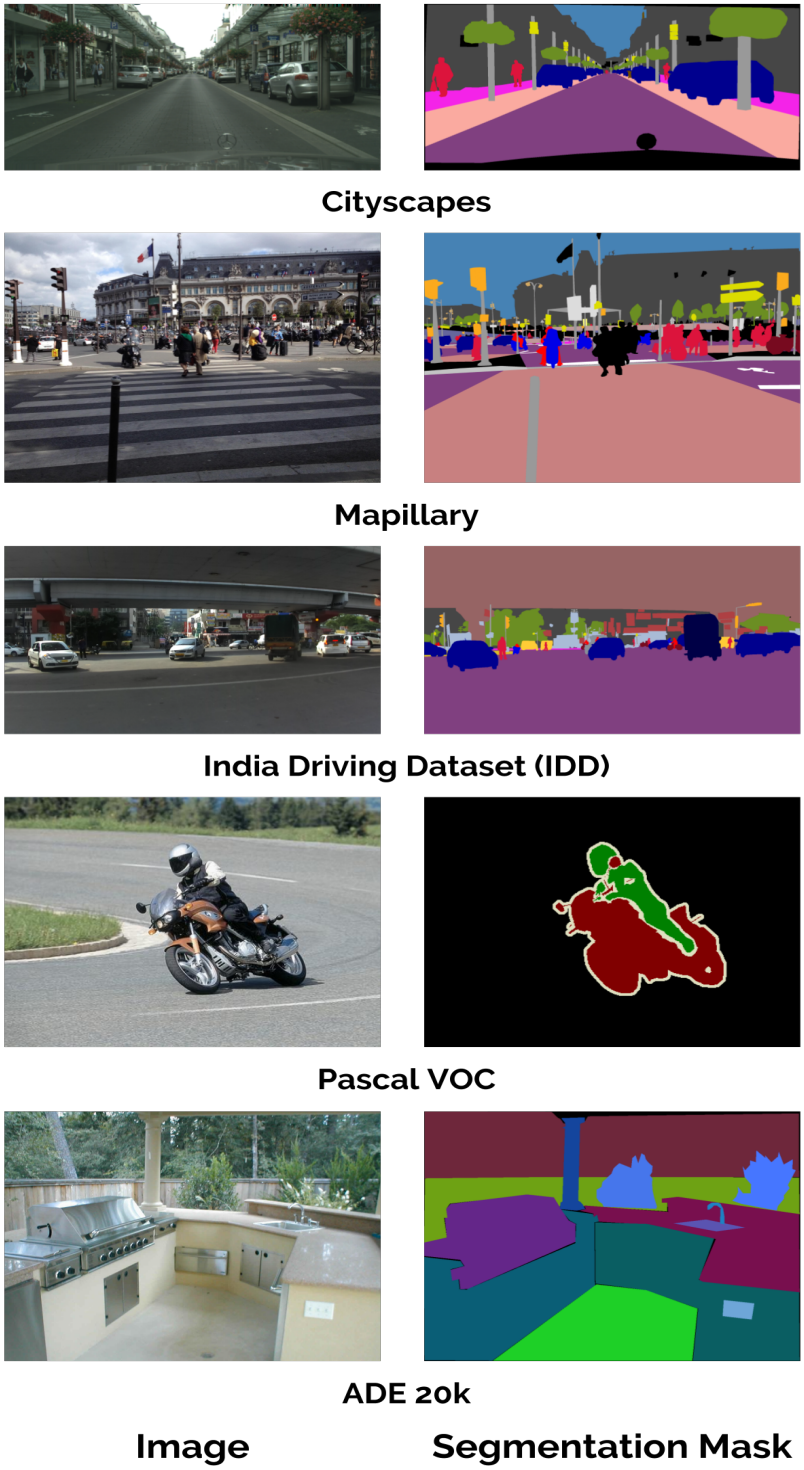

Figure 2.1 Example images with corresponding segmentation masks from a few existing datasets (named below each pair) for semantic segmentation. (on right) are the sample images from a specific dataset (below) and (on left) are the corresponding segmentation masks representing each class with a color.

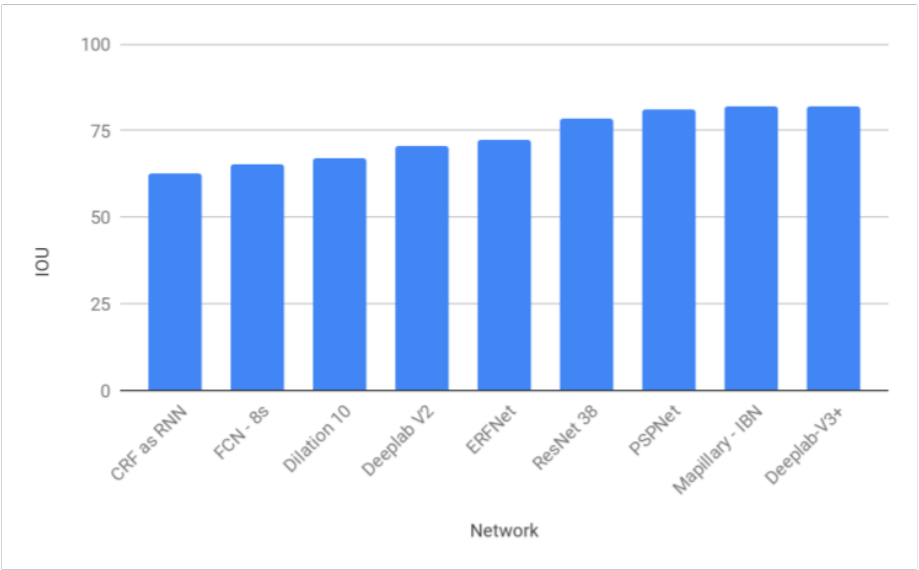

Figure 2.2 Improvement in the IoU of segmentation with proposed methods. The performance of the method increases with introduction of deeper layers and complex multi-scale architectures to obtain finer segmentation predictions.

pairwise potential, Krahenbuhl et.al., [38] proposed a fully connected CRF which establishes pairwise potential on all pairs of pixels in a given image.

Other recent methods improving the efficiency of higher order terms in CRF are proposed by Shanu et.al., [62] [63].

$$E(x) = \sum_{i} \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{2.1}$$

i,j range from 1 to N, where N is the number of pixels in the image. ψ_u is the unary potential for each pixel. The unary potential is the distribution over all classes x_i for a given image that is produced by a classifier. For a deep neural network, the softmax prediction probabilities is a suitable unary distribution. The pairwise potential incorporates a Gaussian kernel for each pixel or image patch pair over an arbitrary feature space. This model improves the segmentation performance when used over a deep learning model prediction.

2.2 Evaluation Measures and Methodology

Once a *model* is trained, we will need to evaluate how good it performs on unseen data. For this, we need some sort of evaluation metric. In this section, I will outline the metrics I have used to evaluate the experiments in my thesis.

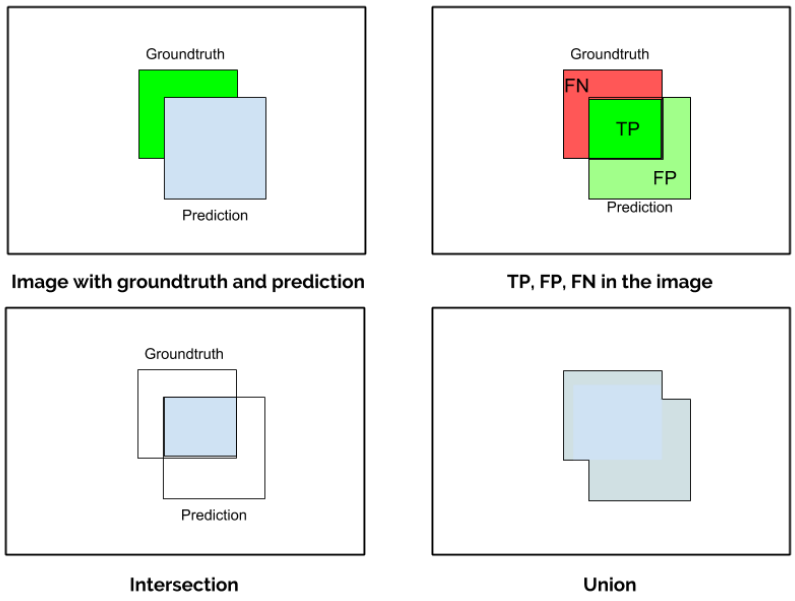

Figure 2.3 Visualization of Intersection over Union IOU metric. In the top row (left) the green segmentation mask is the groundtruth for that object and blue is the predicted segmentation mask; (right) is the regions labeled TP, FP, FN showing the true positive, false positve, false negative regions of the predictions. In the bottom row (left) the colored region is the intersection of the groundtruth and predicted segmentation (right) is the union of the groundtruth and predicted segmentation.

For evaluation, we will have a prediction and a correct segmentation (ground truth). Let us first describe a few terms which will be used in the subsequent evaluation metrics. Given a prediction and a groundtruth, for a given class c_i, TP (true positive) measures the number of pixels which are predicted correctly to c_i; FP (false positive) is the total number of pixels wrongly predicted as c_i when they actually belong to another class; and FN (false negative) is the total number of pixels predicted as some other class when they actually belong to c_i. The evaluation metric, Intersection over Union (IOU), also known as Jaccard Index is defined as,

$$IoU = \frac{TP}{TP + FP + FN} \tag{2.2}$$

In other words, it measures the fraction of the total number of common pixels of a given class c_i between the prediction and groundtruth, and the total number of pixels of the class c_i in prediction and groundtruth. This IOU is calculated separately for all classes and averaged over all classes to get a mean IOU for segmentation.

$$IoU = \frac{prediction \cup groundtruth}{prediction \cap groundtruth} \tag{2.3}$$

The data is partitioned into *training* and *validation* sets. For all the datasets we use in our experiments, the split of *training*,*validation* and *testing* sets are already provided wherein the dataset creators

Dataset	Total	Training	Validation	Testing
Pascal VOC 2012	4378	1464	1449	1456
Cityscapes	5000	2975	500	1525
ADE20k	25210	20210	2000	3000
Mapillary	25000	18000	2000	5000
IDD	10003	6993	581	2029

Table 2.1 Comparison of various datasets and their metrics. Each of these datasets are for semantic segmentation and detail the total number of images in dataset along with their training, validation and testing split for the fine pixel-level annotation.

made sure that the sets do not overlap. We train our models on the training set and finally test it on the validation set, and present all our results in the work on the validation set as access to evaluation servers for the test was very limited.

2.3 Datasets and Annotation

2.3.1 Datasets

There have been many proposed datasets to solve the problem of semantic segmentation. One of the first large scale datasets for segmentation was the Pascal VOC 2011 dataset. Another popular dataset for solving general segmentation problem is the COCO dataset. ADE20k is another recently proposed dataset.

Specifically for road scene understanding, the main topic of the thesis, there have been several popular datasets. The first one was KITTI dataset. Other datasets are CamVid, Cityscapes, and Mapillary. Recently, Varma et.al., [68] released the first large-scale dataset, IDD for autonomous driving in unconstrained environments for Indian Roads. We present the dataset properties for semantic segmentation task in Table 2.1. The table details the total number of images along with providing the split between training, validation and test images. Cityscapes, Mapillary and IDD are datasets for autonomous driving situations which also have dense pixel-wise annotations for instance segmentation. Pascal VOC 2012 is a dataset for general segmentation images. ADE 20k has scenes from indoor and outdoor for general scene parsing. Examples of the images and its corresponding annotations from various datasets are also presented in Figure 2.1. All of these datasets are finely annotated for the semantic segmentation task.

The major drawback of such datasets is the very high cost of its annotation, often requiring 1.5 hours per image. Access to large dataset always increases the performance of a model but it also comes at the cost of increased annotation. The IDD dataset, which is more semantically complex than its predecessors reports the annotation time as 2.25 hours per image. The increase in scene complexity also affects the annotation cost. Section 2.3.2 describes the existing annotation strategies for semantic segmentation.

Figure 2.4 A sample annotation for an image from Mapillary dataset on the web-based annotation tool, LabelMe. (left) is the annotation of an image in which an object is annotated with a polygon brush across its boundary; the circle in green is the control point to click when annotation for an object is completed. (middle) is the dialog box asking for the attributes of the annotated object along with its class label. (right) is the final segmentation mask of the object, car, in the image.

2.3.2 Annotation

Dense pixel-wise annotation of an image is a crucial task for both semantic segmentation and instance segmentation. Dense pixel-wise annotation is usually very costly. To speed up this annotation, several methods make use of interactive segmentation [52][73][11]. Few other methods also use super-pixel annotation [76][23] in which they divide an image to be annotated into its corresponding superpixels and fill the superpixels with labels for the object present in it. There are also other works, which annotate in a semi-supervised manner in which they manually annotate squiggles [6][75] or points [6][33] and inference the annotation of an object from there. All these strategies, definitely to speed up the annotation, but often result in a lower quality.

The popular datasets like Cityscapes and Mapillary use much more quality controlled annotation method derived from LabelMe [55]. This annotation is achieved by drawing polygon bounding boxes around an object to be as fit as possible to its boundaries. The strict Quality Assurance (QA) makes sure that the annotated images are $97 \sim 98\%$ correct in their labels. These annotations often also include void classes, which are ignored during training or testing; the void class is usually such regions in an image where the human annotator is unsure of the class of the object in the categories defined for the dataset. In the following paragraphs, we explain the annotation strategy used for Cityscapes and Mapillary datasets and provide some examples for it.

2.3.2.1 Polygon Annotations

Polygon annotations are the de-facto standard to annotate images for the task of semantic segmentation. The annotation tool for Cityscapes and Mapillary datasets is based on LabelMe. LabelMe is a web-based annotation tool using JavaScript. The user annotates an object by starting a control point and guiding the annotations around the object and end it by clicking the control point. A dialog box pops up asking for the class of the object to be entered. A sample annotation is shown in figure 2.4

The available large scale datasets for autonomous driving scenarios, as mentioned are Cityscapes and Mapillary. They adapt the LabelMe annotation tool and build an interactive annotation tool with additional features. Each image takes 1.5h on average in cityscapes for a 1024×2048 image and similarly 94min average in mapillary for a 1920×1080 image. The objects in both the datasets are annotated from back to front, starting from the farthest object in the image to the nearest one. They also ensure that no object boundary is marked more than one time. This strategy is implicitly beneficial since it will provide depth ordering of the object in the given image.

With the scaling of datasets to prepare robust self-driving systems, the cost of annotation equally increases. This requires 100s of hours of human annotation cost and is laborious and repetitive. The scaling of datasets needs to be cost-effective in terms of annotation. Active Learning [57] has been used in literature for a long time to intelligently annotate data in image recognition and detection which can also be adapted for segmentation. In the next section, we describe active learning and its methods which will be helpful in reducing data annotation cost. In chapter 3 we did extensive experimentation to show how active learning can be adapted for semantic segmentation to efficiently label new data.

2.4 Active Learning

Supervised learning models train on the data and groundtruth available to them. However, these datasets are usually massive, which translates to hundreds of hours of fine annotation. Instead, it is desirable to reduce the annotation effort to train these models while retaining a similar performance. We argue that active learning is a suitable strategy to achieve this. The main idea behind *active learning* is that a given machine learning algorithm will be able to achieve greater performance with fewer training labels, if it is allowed choose that data from which it learns [57]. This strategy had a proven success in image classification [44] [72] [34], object detection [71] , object tracking [5] and image retrieval [66]. There had been few recent works to demonstrate the use of active learning in semantic segmentation on smaller multi-class datasets [69] and in medical image segmentation [24].

While there are nuances present in algorithms for the above works, a general active learning system can be described as in figure 2.5

In the beginning, we train a machine learning model on already present labeled data. We also have access to a large collection of unlabeled data from the same domain. Any proposed active learning method uses some strategies to pick a single or a pool of unlabeled instances which will improve the accuracy or performance of the machine learning model. Such unlabeled instances are queried for labels to an oracle and then the obtained labeled instances are appended to labeled data to retrain the machine learning model.

2.4.1 Querying Strategies

To make sure that we are querying for the right instances for labels that will maximize the performance while minimizing the annotation effort, we need strategies through which a machine learning

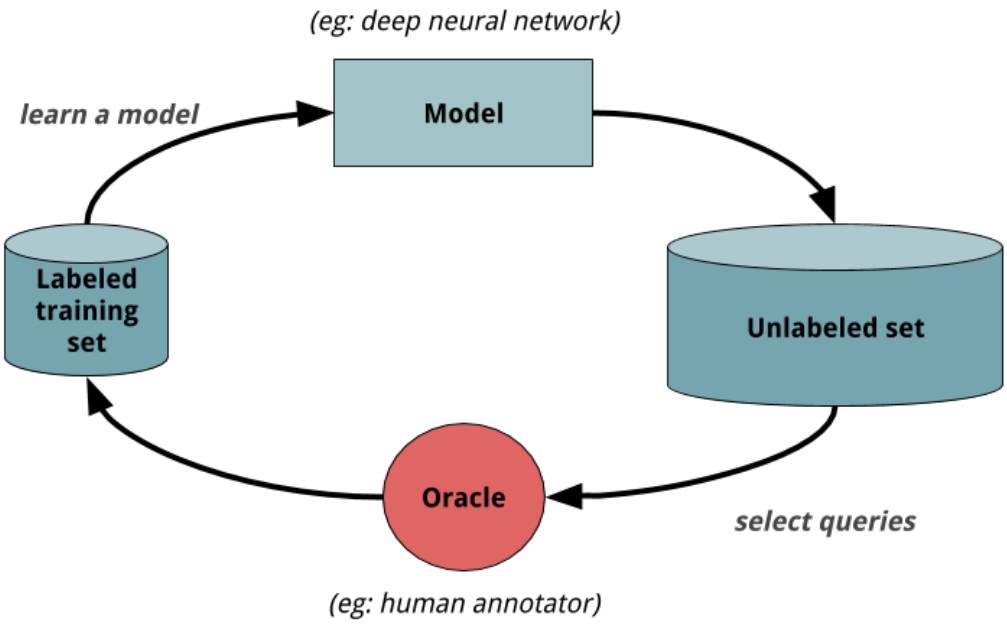

Figure 2.5 A general pipeline describing the active learning methodology.

algorithm can query for the labels from an annotator/oracle. In the following sections, we introduce a few active learning strategies used and discussed in this thesis:

2.4.1.1 Uncertainty Sampling

The concept of uncertainty sampling was first introduced in Lewis et.al [43]. In this strategy, the active learning algorithm queries for those instances for which it is highly uncertain to label. For example, in binary classification tasks described in Lewis et.al. [43] [42] which uses a Bayes Classifier, this strategy queries those instances whose posterior probability is close to 0.5. However, a general strategy for uncertainty sampling is *Shannon Entropy* [61], or commonly known as *entropy*.

$$H = -\sum_{i=1}^{C} p_i \log(p_i) \tag{2.4}$$

For a c-class classification task, if the predicted probabilities of the classes are $p_1, p_2, .., p_c$, equation 2.4 is the *entropy* measure. Higher the entropy, higher is the uncertainty of that instance, or in other words, the classifier is the least confident to classify the instance.

There are other uncertainty measures such as *least confident* and *margin sampling*. However, entropy is a popular uncertainty measure that has been widely used, though, the correct strategy is still application-dependant.

2.4.1.2 Monte Carlo Dropout Sampling

This method has been used in deep-learning models by Kendall et.al., [37] to model the uncertainty of the SegNet architecture [4]. This method models the approximate uncertainty through monte carlo dropout integration.

$$p(y = c|\mathbf{x},\mathbf{X},\mathbf{Y}) \sim \frac{1}{T}\sum_{i=1}^{T} Softmax(\mathbf{f(x)}) \tag{2.5}$$

T is the number of model weights obtained through dropout distribution as theorized in [22]. Here, we average over all model probabilities from dropout at inference time.

2.4.2 Other Active Learning Methods

We have surveyed active learning literature to learn the various strategies used for selecting the appropriate strategy based on the application. We present this section to give an introduction to the reader intrigued about the sub-domain of active learning. This is a brief introduction to various methods and the reader is encouraged to refer *Active Learning Literature Survey* by Burr Settles [57] for a detailed analysis of these methods.

2.4.2.1 Query-By-Committee

This strategy uses a 'committee' of different models $C = \{\theta^1, \theta^2, .., \theta^n\}$ trained on the same labeled dataset. These models then predict the label of given unlabeled instances. The instances on which most of the models disagree is considered to be the most informative query. This is intuitively similar to uncertainty sampling, but the uncertain measure is the disagreement of the committee of the trained models. This strategy was first proposed in Seung et.al., [60]. In order to implement this algorithm, it is necessary to have some measure of disagreement among the models of the committee.

2.4.2.2 Expected Model Change

This strategy selects a query in such a way that it provides the greatest change to a given model once the label is revealed. This was introduced in Settles et.al., [59] for multiple-instance active learning. It was also applied in Settles and Craven [58] for models like CRF. In general, this can be applied in any learning problem which uses gradient-based training. In this, the learner queries for an instance x, which will be added to a labeled set L and result in gradient with the largest change. The expected change score is defined as

$$x^*_{EC} = \underset{x}{\operatorname{argmax}} \sum_i p_\theta(y_i|x) \left\| \nabla l_\theta(L \cup \langle x, y_i \rangle) \right\| \tag{2.6}$$

In this equation, $\nabla l_\theta(L)$ is the gradient of the objective function l and $\nabla l_\theta(L \cup \langle x, y_i \rangle$ is a new gradient we will obtain if we add the pair $\langle x, y \rangle$. Since we do not know label prior to query, we calculate

the expected change, EC as the expectation over the possible labels y_i. This approach has been implemented for semantic segmentation in [69]. However, this model becomes computationally expensive for a larger feature space.

2.4.2.3 Expected Error Reduction

Another similar approach to expected model change is to choose a query that will provide the greatest reduction in the generalization error. In this model, the expected future error is evaluated on the unlabeled set, U if we train the labeled set newly queried pair, $L \cup \langle x, y \rangle$. We query for the instance which will have the least expected future error by minimizing the expected $0/1$ loss or the log-loss. This equivalent to reducing the expected entropy of the unlabeled set or maximizing the expected information gain of a query x. The expected $0/1$-loss is represented as:

$$x^*_{0/1} = \underset{x}{\operatorname{argmin}} \sum_i p(y_i|x) \Big(\sum_{u=1}^{U} 1 - P_{\theta^{+\langle x, y_i \rangle}}(\hat{y}|x^{(u)}) \Big) \tag{2.7}$$

In the equation, $\theta^{+\langle x, y_i \rangle}$ refers to the retrained model with the labeled set L and the new queried instance $\langle x, y_i \rangle$. Similar to expected model change, we do not know the label prior to the query, so we calculate the expected error reduced over all the possible labels y_i. Roy et.al. [53] first proposed this framework for text classification. This method has been improved and used in other semi-supervised active learning methods. Unfortunately, expected error reduction is very expensive in most cases since it requires estimating future error over all unlabeled instances along with incrementally retraining over all possible labels.

2.4.2.4 Variance Reduction

As discussed in expected error reduction, minimizing the generalization error is very expensive. This, however, can be done indirectly by minimizing the output variance. The query strategy can be formulated as follows:

$$x^*_{VR} = \underset{x}{\operatorname{argmin}} \left\langle \tilde{\sigma}^2_{\hat{y}} \right\rangle^{+x} \tag{2.8}$$

Here, $\left\langle \tilde{\sigma}^2_{\hat{y}} \right\rangle^{+x}$ is the estimated mean output variance after we retrain the model on the queried label for instance x. This was introduced in Cohn et.al., [18]. Equation 2.8 is differentiable with respect to x, so it can be used in gradient-based learning methods to reduce the generalization error by indirect reduction of the expected variation.

2.4.3 Related work for Active Learning and Semantic Segmentation

Several deep neural networks have tackled the problem of semantic segmentation. The work Badrinarayan *et al.* on Segnet [4] makes use of encoder-decoder network which eliminates the need to learn

for up-sampling by using pooling indices computed in max pooling. Another work, DeepLab [16] combines the last layer of the DCNN with a fully connected CRF to improve local performance. CRF as RNN[80] integrates the CRF into CNN for end-to-end training. A recent state-of-the-art method, Pyramid Scene Parsing Network [79] divides feature maps into sub-regions to capture global context information before the deconvolutional layers. All this work described work uses the high level of supervision through densely labeled frames to predict the segmentation of an image. As reported by Cityscapes [19], the dense segmentation of each image takes about 2.5 hours which means obtaining labels is a very costly task. In this work, we aim to solve the problem of segmentation with only a few labeled instances and a large number of unlabeled instances, thereby reducing the cost for labeled examples.

Papandreou *et al.* use the information image level tags and partial labels to obtain the segmentation of images. Similarly, work done by Xu *et al.* from University of Toronto [75] incorporate image tags, bounding boxes and partial labels to solve the problem of segmentation. Work by Pathak *et al.* [50] make use of only image level tags to obtain segmentation by constraining the output prediction space. In a similar way, we a trying to solve this problem using only the information from the partial labels.

Active learning methods have been proposed for many years now, and many criteria have been proposed for selecting the most informative data points to query for labels from an oracle. In [35, 42], uncertainty-based measures are used for choosing the queries. In [69], expected change (EC) in the labeling is used for selecting the informative data points, based on which points induce the largest expected change in the current model. In [36], Epistemic and Aleatoric uncertainty measures are studied in Bayesian deep learning models for vision tasks. The epistemic uncertainty measure is used for active learning in [25] and [37]. In [25], a Cost-effective Active Learning approach using dropout at test time as Monte Carlo sampling is proposed to model the pixel-wise uncertainty. It estimates the uncertainty based on the stability of the pixel-wise predictions when a dropout is applied to a deep neural network. An earlier method proposed in [67] achieves semantic segmentation by active learning query for foreground object and Sparse Gaussian Process to obtain segmentation. The other commonly used active learning methods include query-by-committee [60], expected error reduction [26, 53], expected model change [21], variance reduction [29, 56] and Min-max view active learning [30, 32]. The proposed method can be viewed as an uncertainty-based sampling method.

Region-based methods [2, 9] have been popular for image segmentation in the past, and have only recently been integrated into deep neural network models for semantic segmentation. In typical region-based methods, the image is first divided into small coherent regions, until some stopping criterion is satisfied. These small regions form seed regions for the given image and can capture complete objects or its canonical parts. Labels over these seed regions are used to classify its neighboring pixels/regions. We leverage this approach for label propagation in active learning.

2.5 Knowledge Distillation

Deep Neural Networks (DNNs) have achieved state-of-the-art performance in numerous areas of computer vision, like, image classification, object detection, and semantic segmentation. However, these state-of-the-art networks are deeper and complex, and therefore take up significant storage space. This also leads to the requirement of larger computational power and time. This is a huge factor that restricts the deployment of such models on systems with limited resources.

Smaller and faster networks have been developed to overcome the problem of not being able to deploy larger networks for limited resource systems such as self-driving cars and mobile devices. However, these smaller networks suffer in performance as they won't be able to capture as much detail as the deeper networks. One convenient way to improve the performance of the machine learning system designed for a specific task would be to train the same data on multiple networks: smaller or larger networks depending on the requirement, and then averaging their predictions. This invariably increases the computation time at deployment. Especially in self-driving cars, the machine learning model needs to be both smaller and computationally faster and at the same time have a good prediction accuracy. This rules out the possibility of using an ensemble of models during deployment.

This problem was addressed in Hinton et.al. [28] when he introduced the concept of *knowledge distillation*. Knowledge distillation allows transfer of the knowledge of a *teacher network*, which is either an ensemble of high-performance networks or a single high-performance network, to a *student network*, which is usually a shallow network or low-performance network. The concept of knowledge distillation is discussed in detail in the next section.

2.5.1 Distillation

The main premise behind distillation from larger networks to smaller networks is that we use the knowledge gained in larger networks and train the smaller network to mimic that knowledge. This way, the smaller network trained to mimic the larger network will typically perform much better than the smaller network trained in the normal way. Design choices for larger networks can be two ways: either a single best performing models or the average of an ensemble of different models. The rest of the section formulates the way to use the knowledge from larger networks to train the smaller networks.

A neural network for a given task, predicts the class probabilities through a softmax layer. The softmax layer takes the input of logits score of the network predictions and outputs the class wise probabilities. If z_i is the logits score, the softmax layer class probabilities p_i are given in equation 2.9

$$p_i = \frac{e^{z_i/T}}{\sum_i e^{z_i/T}} \tag{2.9}$$

In this equation, T is the temperature the softmax predictions. Usually, T is set to 1 softmax layer. If T is set higher, it gives a softer distribution of probabilities of all the classes.

So distillation is where we transfer the trained model with a softer distribution to another smaller model. If the dataset used for teacher and student networks is the same, we can formulate this problem

as follows. We use two objective functions to train the smaller model. The first is cross-entropy loss function to train with correct labels or groundtruth with softmax computed at temperature 1. The second objective function is for the teacher-student distillation, and is usually a cross-entropy loss function or a KL Divergence loss function at the softened higher temperature depending on the application we want to train.

2.5.2 Related Applications

This concept of knowledge distillation has been adapted to various computer vision and machine learning problems. Romero et.al. [51] introduced Fitnets; they provide additional supervision at intermediate layers along with teacher-student training, called hint-based training to improve the knowledge distillation model performance. Fitnet based approach was adapted to object detection by Chen et.al. [13]. Along with teacher-student training from Fitnets, they had a distillation loss for bounding boxes thus distilling the knowledge from a larger teacher model. Lan et.al., [41] use on-the-fly teacher ensemble to train a student network. This, however, is also computationally very expensive during training because it requires space and computational power for the ensemble of networks.

Knowledge distillation has been applied and adapted to simplify other problems which do not have traditional CNN models. Chen et.al., [17] adapted knowledge distillation to sequence model, to the connectist temporal classification (CTC) for speech models. They introduced knowledge distillation at both frame-level and sequence-level. They showed the improved distillation results on English Switchboard corpus and a large corpus of Chinese speech samples. Mun et.al., [47] learn specialized knowledge distillation models for visual question answering (VQA). They adapt the distillation concept to multiple choice learning problem in where they distill the knowledge from a generalized base model ensemble to specialized multiple choice models. Castro et.al., [12] use distillation loss to retain the information from already trained classes and propose incremental learning of new classes for image classification to overcome the problem of catastrophic forgetting.

Our work aims to adapt the knowledge distillation to semantic segmentation. The main motivation is to increase the performance of smaller, real-time networks which find huge applications in autonomous driving situations. We propose a similar approach as [41], but eliminating the need for on-the-fly ensemble and therefore reducing computational power requirement. Another work recently proposed to improve the fast segmentation models [74] use a consistency loss along with distillation loss. This work has been parallel with ours, but we use ensemble of teacher networks to improve the consistency.

Chapter 3

Active Learning for Semantic Segmentation

This chapter details our proposed active learning strategies for labelling data in semantic segmentation tasks. Our key contributions can be summarized as follows. Convolutional Neural Networks (CNNs) have shown to be very effective for the semantic segmentation task in recent years [4, 45, 79, 78]. To the best of our knowledge, none of the earlier methods for active learning in semantic segmentation were based on CNNs, and this is the first such effort. Recently, fully connected Conditional Random Fields have been used with CNNs [16] for improving the model performance in semantic segmentation task too. We leverage this development in our work to use the dense connectivity in fully connected CRF at the pixel level, along with CNN-based models, to achieve semantic segmentation with limited labeling effort. We apply our proposed method for the road scene understanding problem, which has significant applications in autonomous driving. We evaluate our method on the Cityscapes dataset, and our approach achieves 93.4% percent of the accuracy of the corresponding fully annotated model while querying just 10% of the pixel labels. We also demonstrate the effect of transfer learning over the Mapillary dataset, where the initial model is learned on the Cityscapes data, and this model is used to label the Mapillary dataset with minimal annotation effort.

Unlike classification or recognition where we can obtain single or multiple labeled instances with ease, semantic segmentation is more demanding in the labels and types of labels we can obtain. We cannot, for example, label every pixel of selected unlabeled instances. So it is necessary that our query strategies include ways to collectively annotate the pixels. We present such strategies in section 3.1. Various types of query strategies of sampling pixels for annotation affect the image segmentation output differently as shown in Figure 3.1. It can be observed that the segmentation results improve significantly with the third strategy when compared the first and second strategies. We have also dedicated a section to annotation strategies where we present the suitable methods of annotation for the task.

3.1 Proposed Method

Given a (small) labeled dataset and a deep learning model trained on this data, our objective is to reduce the cost of annotations on available unlabeled data, so as to obtain a new model which provides better validation performance than the initial model. Let $X = P \cup Q$ be the given data, where P is a

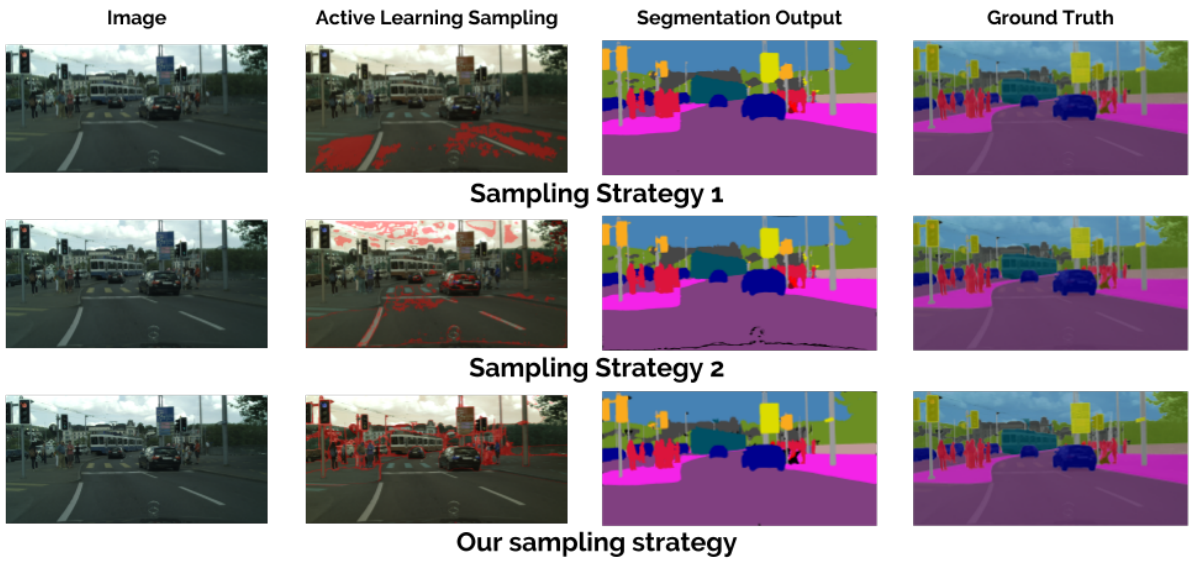

Figure 3.1 An overview of different sampling methods. The manner in which we select the samples (in this case, pixels) affects the final segmentation result. The reference image is from cityscapes dataset. (first and second rows) show datapoints sampled through random strategies and their corresponding segmentation outputs for a given image. (third row) shows the better segmentation output corresponding to our proposed sampling strategy.

set of labeled data and Q is the set of unlabeled data. Let Θ be a deep learning model trained on the labeled data P. Our objective is to reduce the number of annotations on Q using the trained model Θ such that the model's performance (in terms of semantic segmentation) on the un-annotated remainder of Q improves.

We now describe the general approach for active learning in semantic segmentation using uncertainty measures based on entropy.

Our overall active learning methodology starts after training the initial network Θ over the labeled data P. We present active learning strategies to reduce annotations at both image-level and pixel-level. At an image-level, using uncertainty-based measures (which we describe later in this section), we rank images in the unlabeled set Q for annotation. The most uncertain images in Q are then selected as candidate images for annotation. The images are then annotated by an oracle O in groups, i.e. the first group of images is the most uncertain images and these are selected first for annotation. Given a pixel x_i^j, where x_i is the unlabeled image from Q, we can query its label using a given oracle O. $O(x_i^j)$ gives the true label of j^{th} pixel in x_i i.e. x_i^j. We assume group size to be l and the number of groups to be b.

At a pixel-level, for each candidate image in Q, we identify the most uncertain pixels for annotations. For a given image x_i, our objective is to find a given $m \in [0, 1]$ portion of candidate pixels for annotations, i.e. for the image x_i, where $|x_i|$ is the number of pixels, we only annotate $m|x_i|$ number of pixels. These pixels are identified using uncertainty measures computed at a pixel-level (described

later in this section). The oracle O is then queried for the true labels of the identified pixels. Finally, the given network represented by Θ is retrained on the selected data B using the annotations obtained using the oracle O.

The overview of our proposed active learning method is summarized in Algorithm 1. In the final step, the model is retrained using the selected data and the labels obtained from O, to obtain the updated model Θ. We also update the current unlabeled set Q as $Q - B$. The new retrained model and the updated unlabeled set are further provided as input to the next iteration of the algorithm, and the process is continued for the given number of groups b. The uncertainty at image and pixel level is computed using entropy measures, described in Section 3.2.

Algorithm 1 Active learning for Semantic Segmentation

Input: (i) Given data $X = P \cup Q$, where P is labeled, and Q is unlabeled;
(ii) Oracle O for providing labels on unlabeled data;
(iii) Deep learning model Θ trained on P;
(iv) Proportion of annotations m on Q;
(v) Number of groups b and group size l
Output: Updated model Θ
$i = 1$
while $i < b$ **do**
(1) Select a group $B \subset Q$, $|B| = l$ such that $uncertainty(B) > uncertainty(B^{'}) \forall B^{'} \subset Q, |B^{'}| = l$
(2) Query oracle O for relevant m portion of pixels of images in B
(3) Retrain the model Θ on B using the new labels obtained from Step (2) and update Θ
(4) $i = i + 1$; $Q = Q - B$
end while
Return updated model Θ

3.2 Computation of Uncertainty

We describe four different strategies for computing uncertainty, to be used for our active learning methodology described in Algorithm 1: *Pixel-level Entropy*, *Image-level Entropy*, *Edge Pixel-based Entropy*, and finally, the *Region-based Entropy* (which constitutes our proposed region-based active learning strategy and is the most effective of the proposed strategies).

3.2.1 Image-level Entropy:

The entropy at an image level is obtained by summing the uncertainties over all the pixels in the image x_i, as below:

$$H_i = \sum_{j=1}^{|x_i|} H_i^j \tag{3.1}$$

Entropy for an image x_i gives the uncertainty present in the prediction for the model Θ over the entire image x_i. In Step 1 of Algorithm 1, we compute the most uncertain images for the current model Θ (by ranking images in Q based on their respective entropies) and, using the oracle O, we only select a group of l images for annotation. We hence annotate a total of $l \times b$ number of images only in this strategy (which is much lesser than the size of Q in our experiments).

3.2.2 Pixel-level Entropy

Given an unlabeled image x_i, we compute its probability score map $p(c_k/x_i)$, where $k \in \{i, \ldots, C\}$ and C is the number of classes. This gives the probability scores $p(c_k/x_i^j)$ for $j \in \{i, \ldots, |x_i|\}$, where $|x_i|$ is the number of pixels in x_i. $p(c_k/x_i^j)$ gives the probability for the pixel x_i^j belonging to the class c_k. This probability distribution is obtained using the available deep learned model Θ, with no additional annotation effort on the unlabeled set. The entropy (uncertainty estimate), H_i^j at each pixel, is then computed using Eqn 3.2 below.

$$H_i^j = \sum_{k=1}^{C} p(c_k|x_i^j, \Theta) \log(p(c_k|x_i^j, \Theta)) \tag{3.2}$$

This entropy is computed for each image x_i separately. Our active learning methodology first ranks all images in the unlabeled set Q according to their entropies (as in Section 3.2.1), and the m portion of each image is then selected based on pixel-level entropy for labeling.

3.2.3 Edge Pixel-based Entropy

In general, the misclassification rate for pixels at object boundaries/edges is more when compared to the other pixels in the image. This suggests that edge pixels inherently have high uncertainty.

However, not all edge pixels have high uncertainty like small edges inside an object. Also, few boundary pixels have a high chance of being misclassified, but their uncertainty is not as high as other uncertainty pixels. To consider edge pixels for annotation, we modify the pixel-level entropy strategy to give a higher weightage to edge pixels. We use a Canny edge detector to identify edge pixels, and the weighted entropy computed for edge pixels in a given image x_i is obtained as follows:

$$H_i = \sum_{k=1}^{|x_i|} \sum_{l=1}^{C} w_e p(c_l|x_i^k, \theta) \log(p(c_l|x_i^k, \theta)) \tag{3.3}$$

where $w_e > 1$ is the weight given to the edge pixels. For other pixels it is set to 1.

3.2.4 Region-based Entropy

In semantic segmentation, the neighboring pixels are highly likely to have a close relationship and share similar information. Therefore, they are likely to belong to the same semantic class. However,

in all the aforementioned strategies, the entropy of each pixel is calculated independently without considering this relationship. In order to take advantage of the spatial correlation in images, we propose a region-based strategy that is applied at the level of superpixels (SP) in an image. We use (SLIC) [1] to computing the superpixels in a given image, and define the entropy at the superpixel level as the sum of its pixel entropies. To further leverage this strategy, we apply fully connected Conditional Random Field (CRF) model over the segmentation output (probability map) obtained using a deep learned model. This gives the probability score maps for all the pixels. In other words, instead of computing the uncertainty measure over the probability obtained from the current network, we compute our uncertainty measure over the probability map obtained from CRF in this case. We find that this region-based strategy with the propagation obtained using a CRF is immensely useful in obtaining promising results with little annotation. Overview of the Region-based Entropy method is given in Figure 3.2. Here, we take a single image for illustrating each step. The final segmentation image is used for retraining the previously trained network.

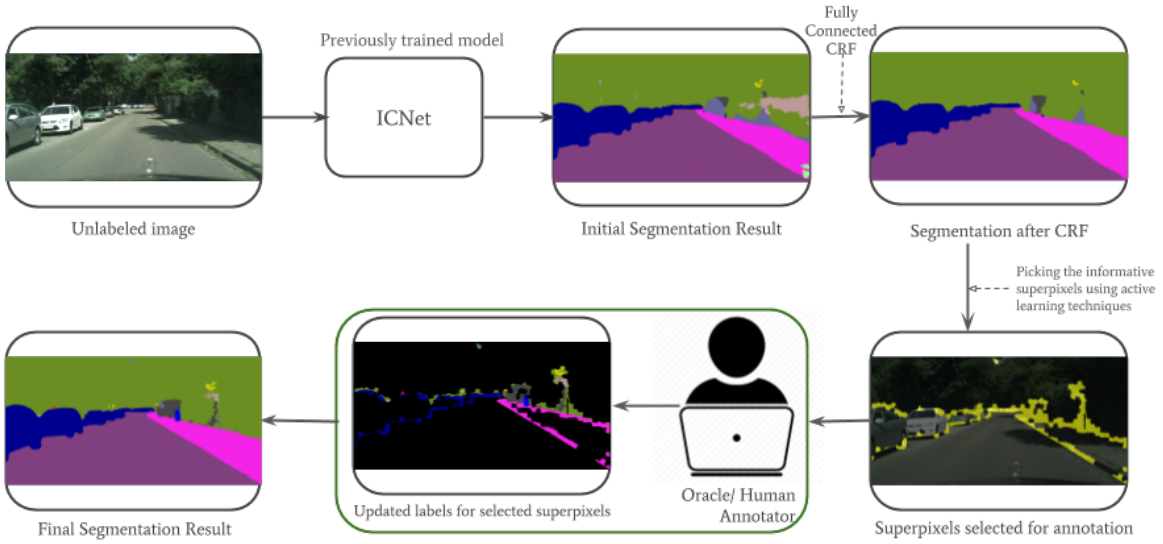

Figure 3.2 Overview of the Region-based Entropy method. Here, the final segmentation image is used for retraining the previously trained model.

3.2.5 Class Specific Selection of Pixels/SuperPixels

In general road scene images, few frequent classes like road, trees, and buildings dominate other classes like traffic signal and signboards. So the number of pixels considered for labeling in these classes dominates other classes. It is desirable to take an equal number of pixels from each class to avoid the dominance of frequent classes. In this method, we first take the deep network trained on the initial labeled data. Now, we represent all the pixels in the labeled images using the feature vector obtained from the network. We construct a feature space using these feature vectors, where each category is

	Baseline	Random 10% GT	Entropy	Entropy + Edge pixels	SP	SP + CRF	Class-specific SP+CRF
# Training images	100% GT	10% GT					
1175	55.6	55.6	55.6	55.6	55.6	55.6	55.6
1475	57.9	55.9 (96.5%)	56.1 (96.8%)	56.4 (97.4%)	56.5 (97.5%)	56.9 (98.2%)	**57.0 (98.4%)**
1775	59.7	56.2 (94.1%)	56.5 (94.6%)	57.0 (95.4%)	57.1 (95.6%)	57.8 (96.8%)	**57.9 (96.9%)**
2075	61.5	56.3 (91.5%)	56.9 (92.5%)	57.9 (94.1%)	58.0 (94.3%)	58.5 (95.1%)	**58.7 (95.4%)**
2375	62.7	56.5 (90.1%)	57.4 (91.5%)	58.7 (93.6%)	58.8 (93.7%)	59.4 (94.7%)	**59.7 (95.2%)**
2675	63.8	56.4 (88.4%)	57.8 (90.5%)	59.4 (93.1%)	59.3 (92.9%)	60.2 (94.3%)	**60.4 (95.2%)**
2975	65.3	56.5 (86.5%)	58.1 (88.9%)	59.8 (91.5%)	60.0 (91.8%)	61.0 (93.4%)	**61.3 (93.8%)**

Table 3.1 Comparison of our various active learning techniques over incremental training data for cityscapes data. Performance is given in meanIoU. Here, SP means SuperPixels and GT means Groundtruth. Results over 1175 images are obtained using the initial trained network over full ground truth. The values in the bracket indicate the ratio to the baseline. It means the percentage of baseline accuracy the method is able to achieve.

denoted with a feature vector calculated using the feature vectors of the pixels in the same category. We represent each category with the mean of feature vectors of all the pixels in that category. Now, for the given unlabeled image, we calculate the similarity (cosine similarity) for each of its pixels with all the class means as follows,

$$sim^{k}_{x_i^j} = \frac{F_{x_i^j} m_k}{||F_{x_i^j}||\,||m_k||}$$

where $F_{x_i^j}$ is the feature vector for the pixel x_i^{jth} obtained from the network and m_k is the feature vector representing k^{th} category. $sim^{k}_{x_i^j}$ gives the similarity between the x_i^{jth} pixel and the feature vector representing the k^{th} category. Next, each pixel is assigned to its most similar category. In this method, instead of computing the entropy independent of its class, we compute the entropy of the pixels in each category separately and pick the high entropy pixels from each class independently.

3.3 Experimental Results

In this section, we evaluate our entropy-based active learning methods for semantic segmentation.

3.3.1 Datasets and Experimental settings

We evaluate our proposed method on two large widely used datasets for semantic segmentation: Cityscapes [19] and Mapillary [48]. Both Cityscapes and Mapillary datasets have emerged as the most popular choices for road scene understanding and autonomous driving. Cityscapes contains 2975 finely annotated training images along with 500 validation images. To evaluate the performance of our proposed method on Cityscapes, we divide the training data into 2 sets. The first set contains 1175 images and the second set contains 1800 images. A deep learning network is trained over the set containing 1175 images using original ground truth, and this model is used as the initial network. Using this model, we try to reduce the number of annotations required on the remaining 1800 images and the performance

is evaluated over a hold-out validation dataset. We use ICNet [78] as our deep learning network. ICNet is a popular network for real-time semantic segmentation on high-resolution images. Since all the images in Cityscapes dataset are high-resolution images (1024×2048), we use ICNet as our deep learning network.

The Mapillary Vistas Dataset [48] is a large scale street-level image dataset. It contains 25000 high-resolution images annotated into 66 object categories. For this dataset, the network trained over cityscapes training data (2975 images) using original ground truth is considered as the initial model. For consistency in the results, we use 19 common classes in both cityscapes and mapillary datasets. The performance over these datasets is measured in terms of the mean of class-wise intersection over union (mIoU).

For all the experiments, we take the number of groups as 6. The group size for cityscapes is 300 and for mapillary is 3000. For both cityscapes and mapillary data, we report the results on the validation data. We only select 10% of pixels for annotation. For computing superpixels (SP), in SLIC, we take the size of the superpixels as 1400. For training ICNet [78], we use SGD solver. For Cityscapes, we resized the training images to 512×1024 and while testing is done on the original images without resizing. On the other hand, for Mapillary dataset, we take the image size as 1920×1080. For both the datasets, we take the base learning rate as 0.01 and the poly learning rate policy is adopted with a power of 0.9. For cityscapes, we train the network for 30K iterations and for mapillay, we train it for 90k iterations. For both the datasets, we fix the momentum as 0.9, weight decay as 0.0001 and we take the batch size as 8. We use Caffe framework for the implementation of ICNet. For fully connected CRF, we take the similar settings as given in Deeplab [16]. We take the default values for $w_2(= 3)$ and $\sigma_y(= 3)$. The values for the parameters w_1, σ_α and σ_β are computed using cross-validation. For cross-validation, we use a small subset of 100 images.

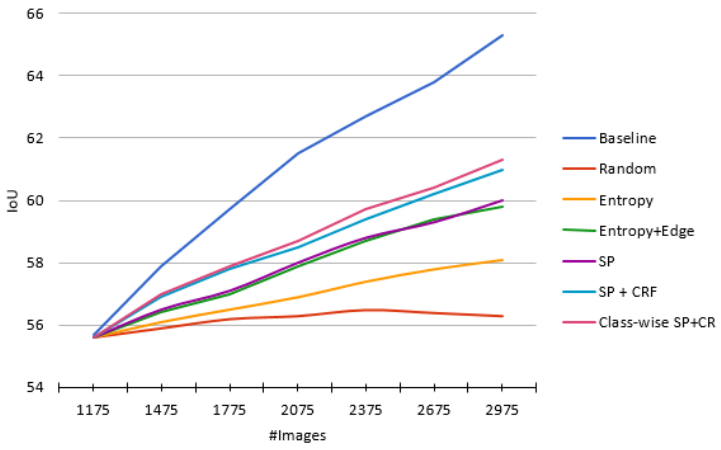

Figure 3.3 Performance of the proposed active learning methods over incremental selection of groups on cityscapes data.

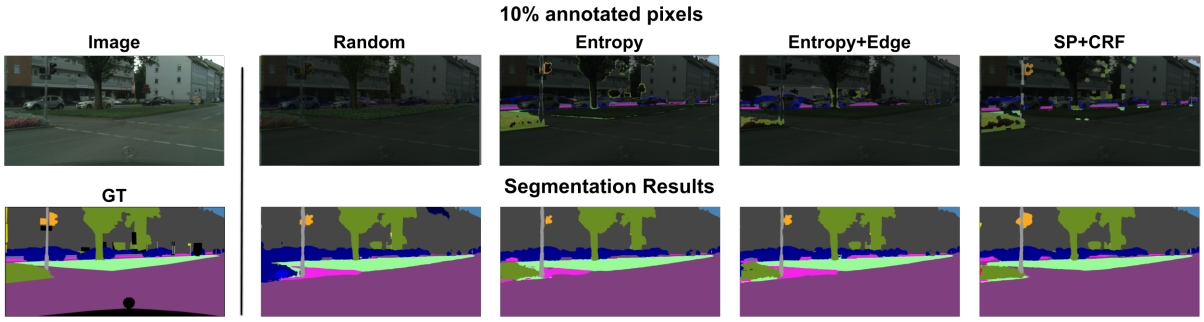

Figure 3.4 Semantic segmentation results on Cityscapes data. Here, we show the 10% of annotated pixels for different methods and their segmentation results. The first column is the original image and its ground truth (GT). In row 1, from column 2-5, shows the selected 10% of pixels in different methods for annotation. In row 2, from column 2-5, shows their corresponding segmentation results.

3.3.2 Results on Cityscapes dataset

3.3.2.1 Image Level Annotations

First, we demonstrate how the incremental selection of images/groups is effective. In this experiment, we label all the pixels in the images using original ground truth (all the pixels in the images are annotated). We only try to reduce the number of images for annotation. We start with the model trained over 1175 images using the full ground truth of pixels. The results for group selection are given in Table 3.1. The results are given under "Baseline". Here, for each group, the performance is evaluated on the validation set. We also show the performance of the network over these incremental groups in Figure 3.3. From the figure (Baseline), we can observe a steep increase in IoU for the initial groups of images compared to the final groups. This is mainly due to the high uncertainty images present in the initial groups compared to the final groups. This means the entropy-based selection allows us to identify the most informative images.

3.3.2.2 Pixels/Region Level Annotations

In all the "Pixels/Region Level Annotations" experiments, we only annotate 10% of the pixels in each image obtained from the corresponding methods and study their performance for the given segmentation task. For all other pixels, we take labels from the trained model.

Random Selection of Pixels The results for the random selection of pixel selection are given in Table 3.1 under Random 10% GT. In this experiment, we randomly select 10% of pixels for annotation. From the results, we can observe that this is not performing well compared to the baseline. Here, the results under the baseline are obtained using full ground truth (annotating all the pixels). The drop in the performance is high compared to the baseline. To get better performance, we need to select the

most uncertain pixels for annotation. However, in the random selection, we are annotating both right prediction pixels as well as wrong prediction pixels for the current network.

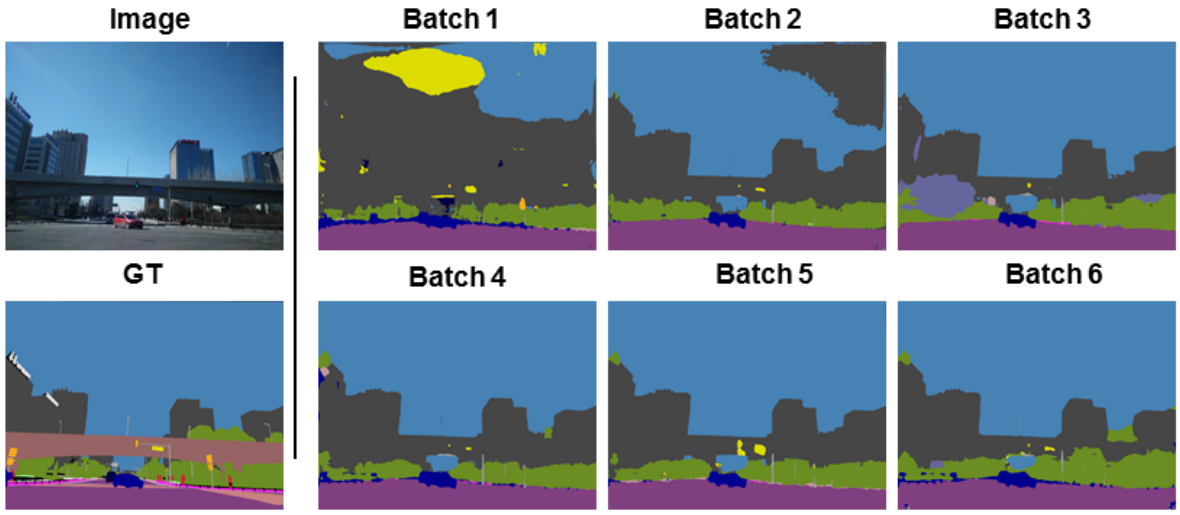

Figure 3.5 Semantic segmentation results on Mapillary data. Here, we show the segmentation results over incremental selection of groups. The first column is the input image and its ground truth (GT). In row 1, from column 2-4, shows the segmentation results over groups 1-3. In row 2, from columns 2-4, shows the segmentation results over groups 4-6.

Pixel-level Entropy based Selection In this experiment, instead of selecting random 10% of pixels for annotation in each image, we use the proposed pixel-level entropy for selecting the pixels for annotation. The results are given under Entropy in Table 3.1. From the results, we can observe that with only 10% of pixel annotations, the network is giving comparable results compared to the baseline. This suggests that our pixel-level entropy based active learning methods are able to pick the more informative pixels (high uncertain pixels). From the results, we can also observe that the pixel-level entropy-based selection is outperforming the random selection of 10% of pixels for annotation. Using pixel-level entropy based selection, we are able to achieve 88.9% of baseline performance.

Edge-Pixels Based Selection In this experiment, while computing the entropy for each pixel, we give higher weightage to the edge pixels. The results are given under Entropy + Edge pixels in the Table 3.1. From the results we can observe that it further improved the performance of pixel-level entropy based selection. In our experiments, mostly the selected edge pixels are coming from the class boundaries. Few segmentation results for edge-based selection are given in Figure 3.4. From the results, we can observe that compared to the entropy-based segmentation results, the class boundaries in Entropy + Edge method are improved. We can also observe that in Entropy+Edge selection, the most uncertain

class boundaries are selected for annotation. Using Edge-Pixels based selection, we are able to achieve 89.4% of baseline performance.

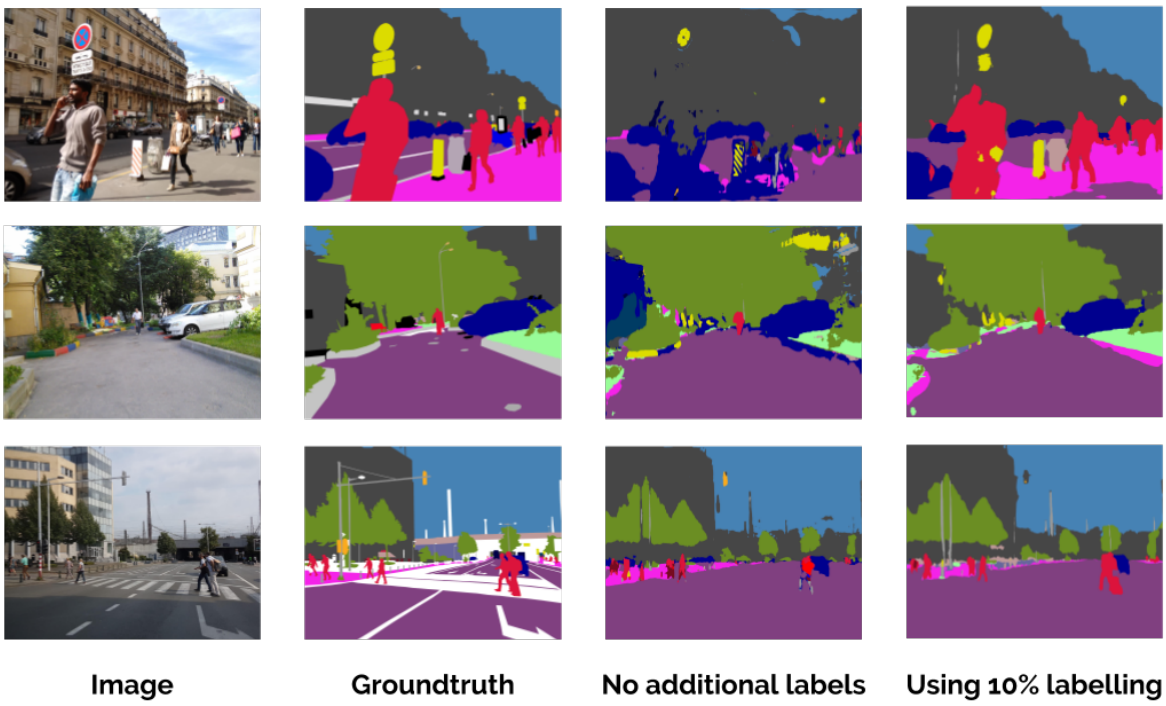

Figure 3.6 Segmentation results using transfer learning on mapillary data. In No Additional labeling, the segmentation result is obtained from the network with out any annotations. In using 10% labeling, the segmentation result is obtained using our active learning method using 10% of pixel annotations.

Region based Selection In region-based selection, we conducted the experiments at the superpixel (SP) level. Here, we select the most uncertain superpixels for annotation. The results are given under SP (Superpixels) in Table 3.1. From the results, it can be observed that the uncertainty computed at the superpixel level is performing well compared to the pixel level. In superpixels, it forces the neighboring pixels which share similar information to get the same class label. Using superpixels, we are able to achieve 91.8% of baseline accuracy. In the next experiment, we use CRF for improving the superpixel-based selection. The results for CRF are given in Table 3.1. Here, we apply CRF at superpixel level. The results are given under SP+CRF. From the results, we can clearly observe that it improved the performance of SP based selection. Using our SP+CRF based selection, we are able to achieve 93.4% of baseline performance.

Class specific based Selection In this experiment, we select the 10% of pixels for annotation from each category. To evaluate its performance, we apply it over SP + CRF based method. The results

Baseline	SP+CRF			
100% GT	10% GT	20% GT	30% GT	40% GT
65.3	61.0 (93.4%)	61.8 (94.6%)	62.4 (95.5%)	**63.6 (97.4%)**

Table 3.2 Performance of SP+CRF on cityscapes data over different percentage of pixel level annotations. GT = ground truth.

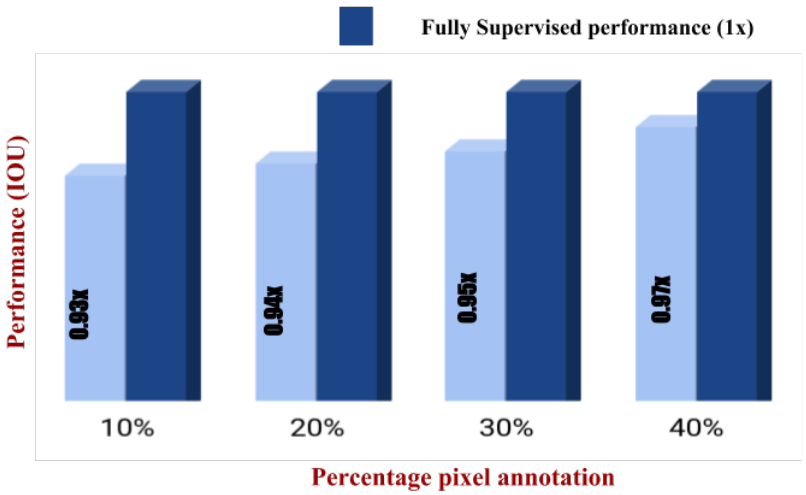

Figure 3.7 Performance of SP+CRF on cityscapes data with respect to different percentage of pixel annotations.

are given in Table 3.1 under Class-specific SP + CRF. From the results, we can observe that it further improved the performance of SP + CRF based method and it outperformed all other methods. Using this method, we are able to achieve 93.8% of baseline performance.

We also compared the performance of all the methods in Figure 3.3. From the Figure, we can observe that Class-specific SP+CRF based selection outperforms all other methods. We have also compared the performance of our active learning method over different percentages of pixel level annotations in Table 3.2. A graphical representation in Figure 3.7 of the performance of the proposed SP+CRF method is helpful in understanding the advantage of this method. Here, we take SP+CRF method for demonstrating the results. From the Table, we can observe that the segmentation performance is improving with the addition of more annotations. Using 40% of annotations, we are able to achieve 97.4% of baseline performance. This also gives us the trade-off between the performance and annotation cost. Here, we can choose, is it better to have 93.4% of peak performance with 10% of annotation cost or 97.4% of peak performance with 40% of annotation cost. We also show the selected 10% of pixels for annotations in different methods in Figure 3.4. From the Figure, we can observe that the regions where both Entropy and Entropy+ Edge methods are given wrong labels are selected for annotation in SP+CRF. We can also observe that the segmentation results are improved in SP+CRF.

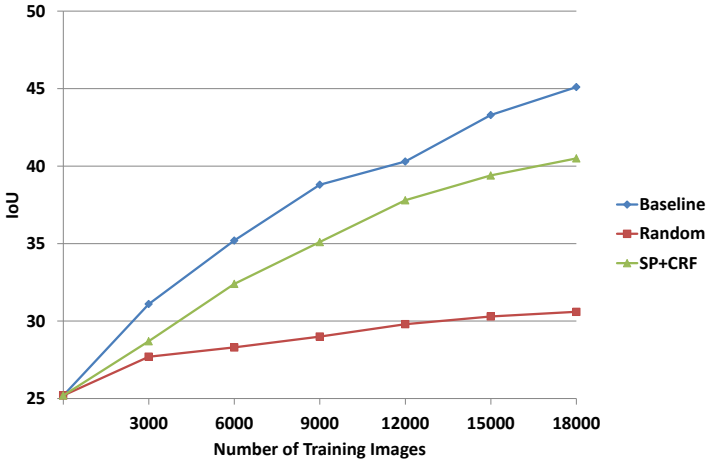

Figure 3.8 Performance of the proposed active learning methods over incremental selection of groups on mapillary data.

# Training images	Baseline	Random	SP + CRF
	100% GT	10% GT	
Cityscapes - 2975	25.2	25.2	25.2
3000	31.1	27.7 (89.0%)	**28.7 (92.2%)**
6000	35.2	28.3 (80.3%)	**32.4 (92.0%)**
9000	38.8	29.0 (74.7%)	**35.1 (90.4%)**
12000	40.3	29.8 (73.9%)	**37.8 (93.7%)**
15000	43.3	30.3 (69.9%)	**39.4 (90.9%)**
18000	45.1	30.6 (67.8%)	**40.5 (89.8%)**

Table 3.3 Results on Mapillary data. The results are given over incremental groups. The percentage values in the bracket indicates the relation to the baseline.

3.3.3 Results on Mapillary Dataset

The results on transfer learning are demonstrated over the mapillary dataset. The results are given in Table 3.3 under Baseline. Here, the trained model on cityscapes data (2975 images) is used for propagating the labels to mapillary data. We also show the performance of the network over these incremental groups in Figure 3.8. From the results, we can observe that the maximum improvement in the accuracy is obtained in the initial groups. We also presented the segmentation results over these groups in Figure 3.5. We can observe that the segmentation results are improving with the addition of more groups.

The results over Mapillary data for SP+CRF are given in Table 3.3. On complete training data with 18000 images, we obtain 89.8% of baseline performance by querying only 10% of pixels for labeling. It outperformed the random selection of 10% of pixels for annotation. In Figure 3.6, we show the

segmentation results obtained using transfer learning. Here, we show the segmentation result obtained from the Cityscapes model without any additional labeling (No additional labels) and the result obtained using 10% of pixel annotations using the proposed method (Using 10% labeling).

Internal experiments showed us that using different cameras with different imaging sensor characteristics (e.g. dynamic range, signal to noise ratio etc) for recording same dataset also leads to failure of the current model even for the same classes. In this context, we refer to similar dataset captured with different imaging sensors as different target domains. In this work, we use Cityscapes as source domain and Mapilliary as target domain. Cityscapes dataset is obtained from a single camera source and Mapillary consists of similar dataset captured using various imaging sensors such as mobile cameras, automotive cameras etc. The predictions on Mapillary from Cityscapes trained model also proves that a different camera plays a significant role in the performance of a model and how active learning is advantageous to improve the performance of that model without heavy annotation load. So, in this work, we define the domain as similar datasets captured using imaging sensors with varied characteristics.

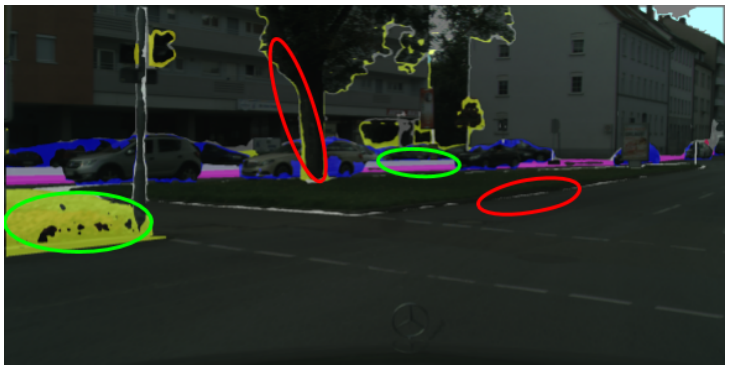

Figure 3.9 Few examples of region selection for human annotation. The regions marked green are labeled by watershed algorithm similar to [10] and the red regions are annotated by [46]

To further evaluate the proposed entropy-based uncertainty measure, we compare it with the uncertainty measure given in [25]. This method uses Monte Carlo dropout to estimate the final predictions and uncertainties. The results are given in Table 3.4. In terms of accuracy, the proposed entropy based uncertainty measure is comparable with [25]. However, the proposed entropy-based method is 8 times faster compared to [25]. In terms of class-wise IoU, the entropy-based method is performing well for the person and vehicle classes (truck, bus, train, motorcycle, bicycle). On the other hand, the uncertainty measure defined in [25] is performing well for the classes like traffic light, traffic sign, terrain, sky, and rider. Class-wise IoUs are given in the supplementary material.

3.4 Annotation Strategies

For our experiments, our oracle has access to all groundtruth we withheld for the unlabeled data. However, for newly collected data without annotations, we need a human annotator to label the pixels

	Baseline	Entropy	[25]
	100% GT	10% GT	
Accuracy (mIoU)	56.0	49.9 (89.1%)	**50.0 (89.2%)**
Computational Time	-	**0.09** Sec	0.7 Sec

Table 3.4 Comparison of the proposed entropy based uncertainty measure with the uncertainty measure given in [25]. The computational time is given in seconds.

	Entropy	Entropy + Edge	SP	SP + CRF	Class Specific SP+CRF
Computational Time (Sec)	0.09	0.09	1.41	2.48	3.62

Table 3.5 Comparison of all the methods in terms of computational time. The computational time is given in seconds.

and superpixels selected by the active learning methods. We propose the following strategies depending on the type of annotation required.

For annotation of regions by an oracle, we use a similar annotation tool to the one in [10]. The larger regions are annotated through the watershed algorithm. For annotating smaller/narrower regions, we selected the regions through a magnetic lasso tool proposed in [46]. Few examples for annotation are given in Figure 3.9. The regions marked in green are suitable for annotation by [10] and the regions marked in red are annotated by [46].

3.5 Additional Results

To show the pixel level uncertainties computed using the proposed entropy-based method, we plot a heatmap for the entropy in Figure 3.10. From the Figure, we can observe that the entropy values are higher at the class boundaries and at the edges.

We compare all the proposed active learning methods in terms of computational time in Table 3.5. From the results, we can observe that "Entropy" method is computationally faster compared to other methods. In the other methods, computation of additional information like edges, superpixels and CRF contribute to the extra computational time. Class specific SP+CRF is computationally slower compared to the other methods. This is mainly due to the computational time involved in the CRF and the computations needed for constructing the feature space.

In all our active learning methods, we select only 10% of pixels for annotation. However, we do not need to relabel all these 10% of pixels, i.e, the labels for few of these pixels obtained from the previous model may be correct (assuming ground truth is present for verification). In Table 3.6, we show the percentage of pixels needed relabeling in different methods. From the results, we can observe that the percentage of relabeling pixels are higher in Class specific SP + CRF compared to the other methods. This means, Class-specific SP + CRF method is picking the more wrong predicted pixels for relabeling. This is the main reason for its superior performance compared to the other methods. We also show the selected relabeling pixels for each method in Figure 3.11.

	Entropy	Entropy + Edge	SP	SP + CRF	Class specific SP + CRF
Accuracy	59%	66%	71%	77%	**79%**

Table 3.6 Comparison of percentage of pixels needed relabeling in the selected 10% of pixels for annotation in different methods.

classes	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
Entropy	0.94	0.69	**0.83**	**0.27**	**0.32**	0.39	0.27	0.43	0.84	0.30	0.66	**0.65**	0.17	0.82	**0.29**	**0.67**	**0.25**	**0.09**	0.49
[9]	**0.95**	**0.70**	0.82	0.25	0.21	0.39	**0.41**	**0.49**	**0.89**	**0.60**	**0.75**	0.46	**0.29**	**0.84**	0.20	0.50	0.07	0.07	**0.53**

Table 3.7 Class-wise IoUs for the proposed entropy-based uncertainty measure and the uncertainty measure given in [25].

We also analyzed the role of group size in the proposed active learning method in Figure 3.12. Here, we compare the performance of the group size of 180 with the group size 300. From the results, we can observe that higher group size is slightly performing well compared to the lower group size. However, lower group size is slightly performing well compared to the higher group size in the initial groups.

In the comparative study, we compare the proposed entropy based uncertainty measure with uncertainty measure defined in [9]. The class-wise IoU for this comparison are given in Table 3.7. We also show the selected 10% of pixels for annotation in both the methods in Figure 3.13.

3.6 Summary

This chapter presents the first contribution of this thesis, reducing the data annotation constraint of semantic segmentation tasks. We proposed several active learning strategies for which picks a small percentage of the most informative pixels or superpixels of an image for annotation. With such strategies we can achieve near fully-supervised performance but still significantly reduce the amount of data annotated. The qualitative and quantitative results were presented on datasets for road scene understanding, cityscapes and mapillary.

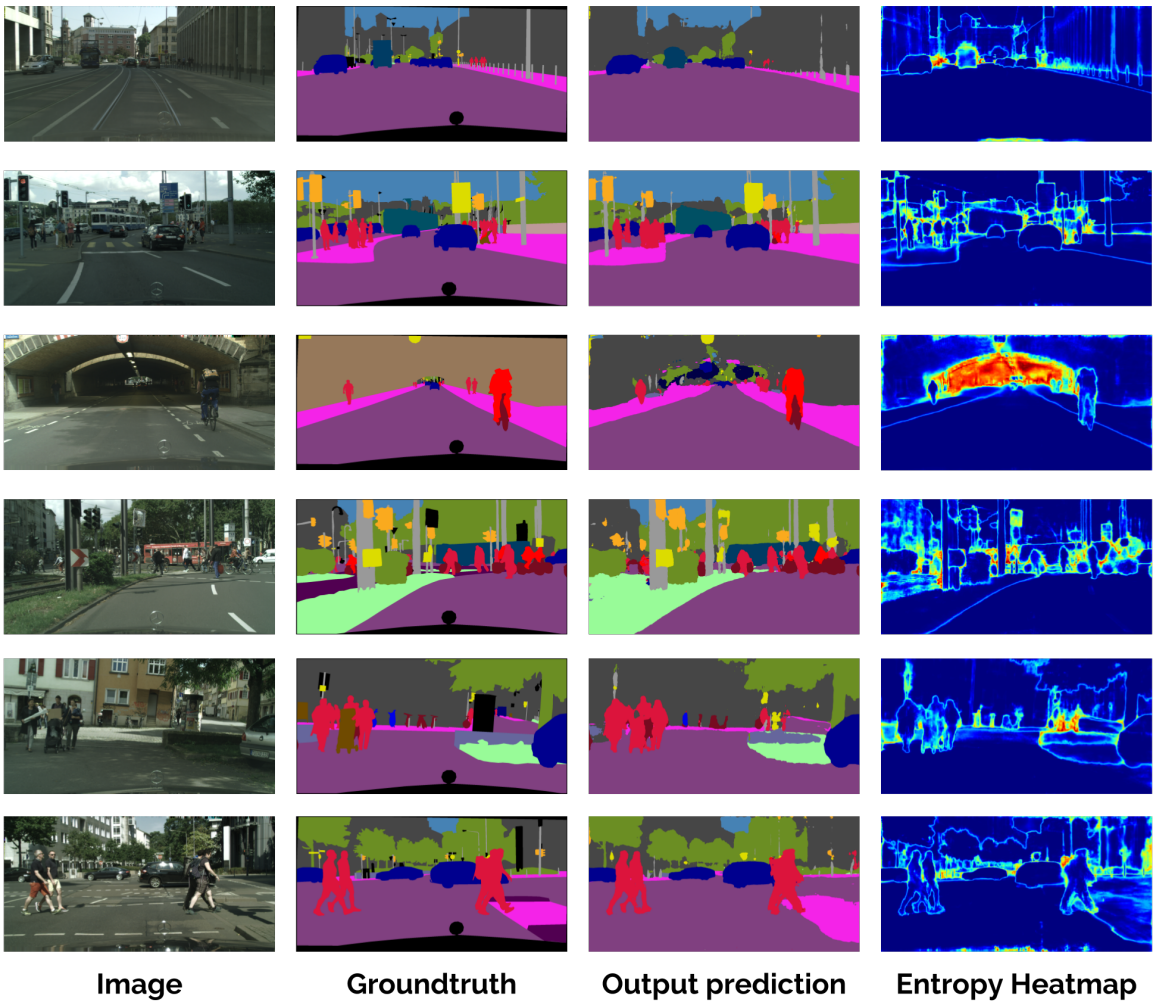

Figure 3.10 Entropy Heatmap. Output prediction is the segmentation output from the network. In the heatmap, we can observe that entropy values are higher at the class boundaries and at the edges.

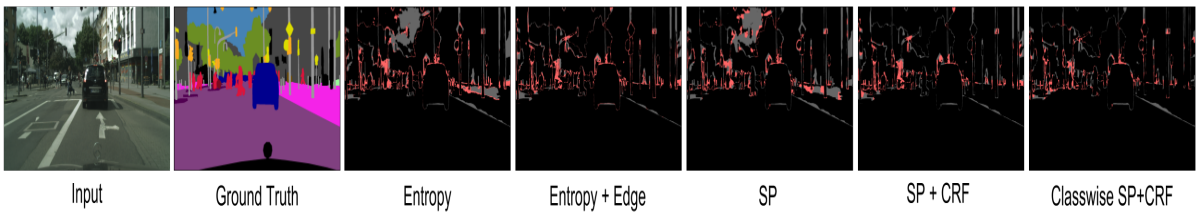

Figure 3.11 Selected relabeling pixels for all the methods.

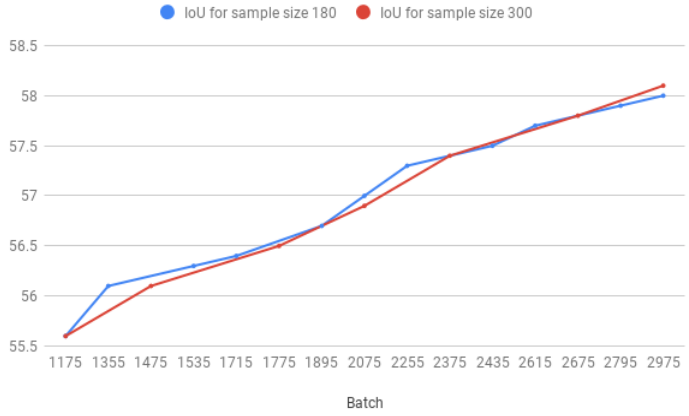

Figure 3.12 Comparison of group sizes for active learning

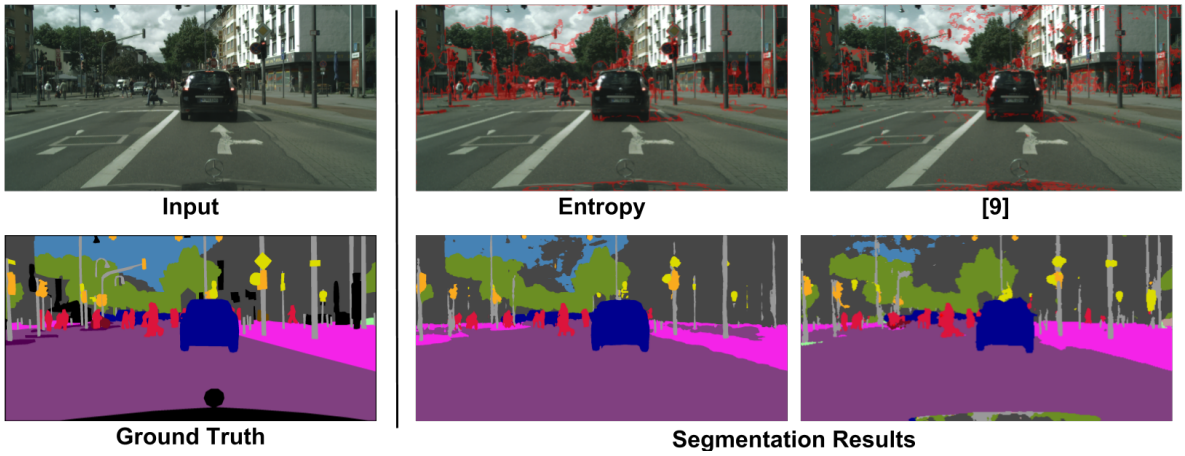

Figure 3.13 Selected 10% of pixels for annotation obtained from Entropy and [9].

Chapter 4

Knowledge Distillation for Semantic Segmentation

This chapter is motivated by the practical requirement of real-time semantic segmentation models for autonomous driving scenarios. Model size and parameters heavily influence the performance of a segmentation model, with the best performing models usually being complex and deeper. The training time and space complexity for such models is also high and require a lot more computational power than average. Often-times these best performing models are also very slow during inference. Usually, any proposed compression of the segmentation network architectures for real-time inference suffers from a significant loss in performance. In this work, we explore the possibility of obtaining better performance for real-time models that learn from the already existing best performing models.

Self-driving cars are perfect examples of complex machine learning systems working together to solve a task. These self-driving cars being stand-alone systems isolated from constant power and computation grids are usually constrained by the onboard computation power, which means it prefers that the model it runs take as little space as possible. At the same time, these models also need to run real-time. For example, ENet [49] has very fewer parameters (0.37M) and runs real-time, but it suffers in performance on any given dataset compared to state-of-the-art models like PSPNet [79]. The main challenge here is to improve the performance on networks like ENet, while not changing its parameter size or time complexity during inference.

4.1 Introduction

Deep neural networks are proven to perform well on computer vision tasks such as image classification, object detection, and semantic segmentation. In particular, for the task of semantic segmentation, deeper network with complex architectures result in a robust segmentation. A recent work, DeepLabv3 [15] proposes using atrous convolutions along with spatial pyramid pooling layers, achieving almost state-of-the-art performance on various segmentation datasets. However, this architecture presents a challenge to deploy as a real-time segmentation system due to its higher size and inference time. There were several attempts in the literature to build smaller networks which will reduce the number of parameters and computational time during inference. These smaller networks usually suffer from a significant decrease in the performance of the model. An ideal model would be one which is

real-time and is better performing which leads to the question of whether we can somehow learn from the state-of-the-art models. One such sub-domain that tackles this problem is *model compression* which is discussed in the subsequent paragraph.

One of the earlier known methods of model compression was in Bucilua et.al., [8] where the proposed method was to train a neural network which will mimic a large ensemble of networks. In this, they obtained labels or unlabeled data through the output from ensembles and then trained their neural network using that data. It was then adopted to deep learning models in Ba & Caurana [3]. Subsequently, Hinton et.al., [28] introduced *knowledge distillation* framework, which is a student-teacher training approach to train a smaller neural network to learn from better performing network. During training, the student network predicts both the true classification labels and also tries to mimic and predict the softened outputs of teacher network. One way to get the softened outputs is the introduce a temperature parameter in the softmax prediction for classification. Higher the temperature, more softened are the outputs and lesser is the disparity between prediction probabilities. Note that introducing the temperature parameter does not change the maximum or minimum value of prediction probabilities but merely scales their distribution to be nearer. Later, Romero et.al., [51] introduced FitNets, which additionally takes *hints* from the teacher network in intermediate layers along with the teacher-student prediction loss. Chen et.al. [13], used this idea of hint learning and proposed a compressed framework to learn object detection through knowledge distillation. We show the adaptation of *knowledge distillation* to compressed semantic segmentation models and our contribution to this domain is as follows:

- We adapt the concept of knowledge distillation to semantic segmentation, with a focus to improve the performance of fast and real-time segmentation models to be deployed for autonomous driving situations.
- We propose to train the student network with the teacher output of ensemble networks and labeled data, thereby eliminating the need for the presence of a teacher network during training. This both reduces the overall network size and training time when compared to the training of the conventional teacher-student method.

The proposed method shows significant improvement of the model performance on the real-time segmentation models. The rest of the chapter is structured as follows. Section 4.2 presents in detail our approach for the proposed method. In, section 4.3 we explain the experimental settings and present our results to show the increased performance of student networks with our proposed method. A brief discussion in section 4.4 summarizes the work that has been presented in this chapter.

4.2 Approach

Our approach has two kinds of networks: the first is an ensemble of segmentation teacher networks. These teacher networks are chosen such that they provide atleast some contrasting information for each class and such that the performance of their ensemble outperforms each individual networks. The second

network is the student network which is a real-time segmentation network and optionally also has very few parameters.

4.2.1 The Power of Ensembles

We choose 3 similar better performing teacher networks and ensemble them to get the final teacher outputs. As we already available trained models for these networks, we obtain their softmax outputs, $p^1(x), p^2(x), p^3(x)$ where x is the input image. We also have access to the class wise performance of each model for a given dataset, c_j^i where j a class in the dataset and i represents the teacher network. To calculate the ensemble probabilities of all teacher networks, we average them according to the weight of class-wise performance.

$$p_j^T = \frac{\sum_{j=1}^{k} c_j^i p^i}{\sum_{j=1}^{k} c_j^i} \tag{4.1}$$

In equation 4.1 k is the total number of classes and p_j^T is the softmax probability of the ensemble for the j^{th} class. We save the weighted softmax average and safely discard the teacher networks and their corresponding trained models.

As proposed in [28], we soften the softmax probabilities for each teacher network. The temperatures for each network are empirically decided to assure consistency of the softmax score across all the teacher networks.

$$p^i(z_k) = \frac{e^{\frac{z_k}{T_i}}}{\sum e^{\frac{z_k}{T_i}}} \tag{4.2}$$

where $p(z_k)$ is the probability score of class k and T_i is temperature of teacher network i.

4.2.2 Training the student network

The student network is trained on the same input data as the teacher networks, white it is optimized to mimic the output predictions of the teacher ensemble. A good choice of the student network for our task is a network that has real-time inference and optionally, has a very low number of parameters.

The student network is, therefore, jointly trained with labeled input data and its corresponding teacher softmax predictions. The loss of the student network for training with the labeled data is the commonly used *cross entropy* loss

$$L_{ce} = -\sum_i y_i \log(p(x_i)) \tag{4.3}$$

Where y_i' is the groundtruth and $p(x_i)$ is the softmax prediction.

To mimic the teacher performance for the same labeled dataset, we found the KL Diverence Loss to be the most effective. The student network is also softened when comparing it to the teacher ensemble during training. This temperature parameter is decided based on the experimentation to maintain

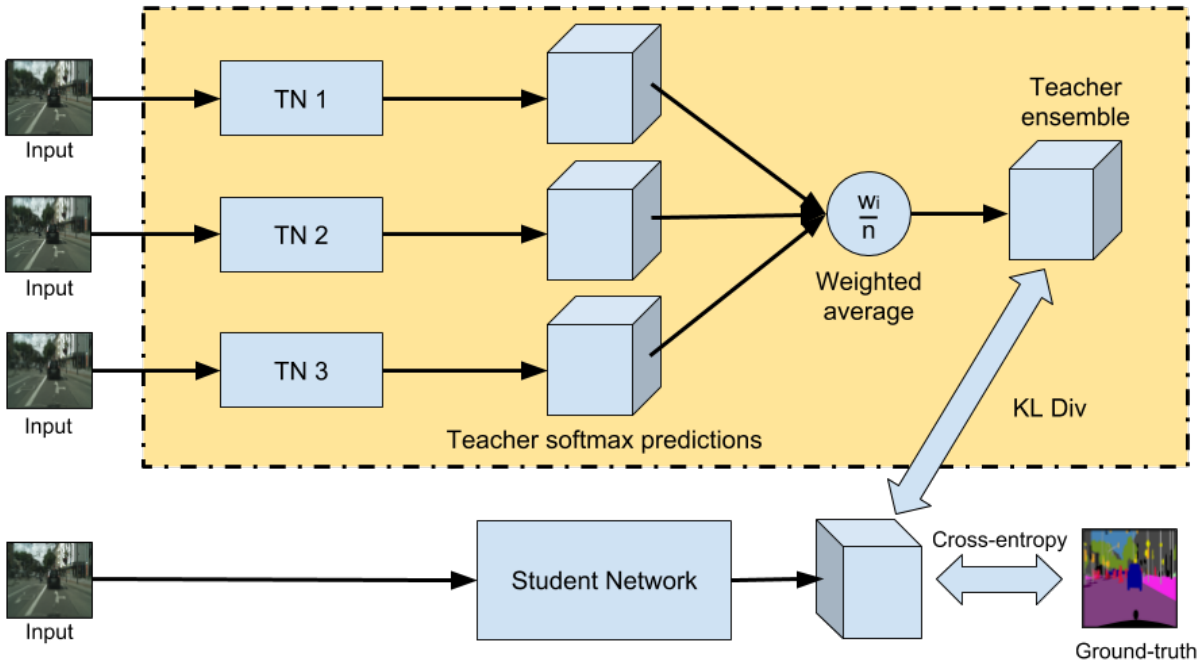

Figure 4.1 Our proposed *knowledge distillation* method for semantic segmentation.

consistency between stand-alone student softmax probabilities and the teacher ensemble softmax probabilities.

$$L_{kld} = \sum_i p_s \log \frac{p_s}{p_t} \tag{4.4}$$

Here p_s is the student softmax probabilities, $p_s = softmax(\frac{z_s}{T})$ with z_s being the final score output of student network. p_t is the softened softmax probability of the ensemble of network as described in equations 4.1 and 4.2.

We jointly train the student network on labeled data and teacher outputs using the proposed knowledge distillation loss.

$$L_{kd} = L_{ce} + \lambda L_{kld} \tag{4.5}$$

Here, λ is a hyper-parameter which decides the weightage of the KL Divergence loss. This has been empirically set to be similar to the cross-entropy loss. This proposed method is shown in figure 4.1.

After training, the teacher ensemble can be discarded, and the inference is real-time on the stand-alone network. This teacher-student trained model has improved performance due to the joint training with the labeled data and the teacher supervision. We show the improved results in the next section and also discuss the experimental settings.

Classes		Road	Sidewalk	Building	Wall	Fence	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	Score Average
mIoU	**ERFNet**	0.9857	0.9093	0.941	0.8515	0.8276	0.6533	0.6674	0.7644	0.9304	0.8383	0.9515	0.8203	0.684	0.952	0.9069	0.9213	0.9221	0.7523	0.7476	**84.35%**
	DeepLab	0.9828	0.8862	0.9333	0.8068	0.8042	0.507	0.6312	0.7329	0.9223	0.8057	0.9388	0.8036	0.6966	0.941	0.8833	0.9162	0.9247	0.7655	0.7553	**82.30%**
	ICNet	0.9824	0.8768	0.9229	0.7306	0.7033	0.4946	0.5266	0.656	0.9172	0.7785	0.947	0.764	0.583	0.9362	0.8434	0.8781	0.8985	0.6516	0.6773	**77.7%**
	Ensemble	0.9889	0.926	0.9509	0.8907	0.8715	0.6782	0.715	0.8108	0.94	0.874	0.9556	0.8531	0.7565	0.959	0.9367	0.9425	0.9465	0.8205	0.7955	**87.43%**

Table 4.1 Class-wise performance of individual teacher networks and the teacher ensemble on Cityscapes dataset

Class	mIoU
ICNet	0.677
ERFNet	0.7234
DeepLab	0.693
Ensemble	**0.7544**

Table 4.2 Validation performance of individual teacher network and the teacher ensemble on Cityscapes dataset

4.3 Experiments and Results

4.3.1 Dataset

As this work is focused knowledge distillation for real-time segmentation, we performed all experiments on the Cityscapes [19] dataset for road scene segmentation in autonomous driving situations. It has 5000 finely annotated images, divided into 2975 for training, 500 for validation and 1525 for test. We use the training set for our teacher-student ensemble and evaluate the model on the validation set. The dataset has presented 31 classes out of which 19 are trainable. All the images are at 1024x2048 resolution.

4.3.2 Teacher Ensemble

As explained in section 4.2.1, we choose three similar performing networks. These networks are chosen such that they provide contrasting information in atleast 3-4 classes. Our networks are ICNet, ERFNet and DeepLab. Table 4.1 shows the class-wise performance of all the networks on the Cityscapes training set. The class-wise IoU of each network is used as the class-wise weights used in equation 4.1. Table 4.2 shows the performance of the individual networks on the validation set of Cityscapes. It is to be noted that class-wise and overall IOU of the teacher ensemble outperform the individual networks.

The teacher ensemble present in Table 4.1 obtained is after softened softmax weighted average of the individual networks. Softened softmax of each individual network does not change the performance (IOU) of the network; it only redistributes the softmax values to be closer in distribution. The temperatures of ERFNet, ICNet and DeepLab are set to be $3, 4, 5$ respectively based on the empirical evaluation.

4.3.3 Training the student network

We have trained and evaluated our proposed method on 3 competing networks on Cityscapes dataset. For all the experiments, the best value for the hyper-parameter λ, the weightage of KL Divergence loss as mentioned in equation 4.5 is empirically determined to be 100.

ENet: We first train ENet stand-alone without the teacher loss to get the baseline performance. We use the experiment parameters mentioned in the ENet paper. We train with a batch size 2, the input resolution of the image is 512×1024. It is trained for 300 epochs on the full training set of 2975 images.

To train the teacher-student network, we empirically set the student temperature to $T_s = 5$. We train the network with the same training parameters as the stand-alone network training details mentioned above. The improvement in the performance is mentioned in Table 4.3

ICNet: ICNet is trained standalone without teacher loss whose performance will give the baseline. The paper is trained with a batch size of 16 which was not possible use due to computational constraints, so we set the batch size to 2. Rest of the experiment parameters are the same as given in the paper.

The temperature set for teacher-student training is the same we set in the teacher ensemble, $T_s = 4$. The student is trained with the parameters mentioned in the above paragraph. We found a significant improvement in the performance of the network and is reported in Table 4.3

ERFNet: The main reason to do this experiment was to check if the teacher ensemble also improves the best performing individual network of the ensemble. The training parameters for the experiment are the same as reported in the ERFNet paper. (mention parameters).

We then jointly train this network with Cityscapes data and the teacher ensemble softmax probabilities, with student temperature set to $T_s = 3$ for the teacher loss. We found no significant improvement in the results over the standalone network and reported the same in Table 4.3.

Table 4.5 shows the comparison of the inference time of the student network during validation when trained stand-alone and when trained along with teacher supervision. It is necessary to note that, the inference time of the network does not increase, as we do not use the teacher supervision once the model is trained.

We found that ICNet, ENet performance improves significantly. This is due to the competing results of these student networks train on KL Divergence loss alone. These results are in Table 4.4. We also observed that the performance of ERFNet remains the same despite the additional supervision from the teacher ensemble. Also, if ERFNet is trained only with teacher supervision as in table 4.4, the network does not perform as better as stand-alone cross entropy loss. We theorize that this is due to saturation of the Cityscapes dataset performance for ERFNet. One way to overcome the performance curse of this network is to use a larger dataset for the experiments.

Student Network	IoU	teacher-student IoU
ENet	52.3	**55.8**
ICNet	54.7	**57.8**
ERFNet	72.34	72.2

Table 4.3 Results of proposed knowledge distillation method on Cityscapes dataset for different student networks. (middle) is the IoU obtained when the network is trained with the regular cross-entropy loss. (right) is the improvement obtained after the teacher-student training method.

Student Network	IoU	KL Divergence IoU
ENet	52.3	52.0
ICNet	54.7	54.4
ERFNet	72.34	69.3

Table 4.4 Comparison of performance of training a student network with (middle) the regular cross entropy loss and (right) with only teacher supervision.

4.4 Summary

This chapter presents the second contribution of this thesis, improving the model performance of real-time networks for semantic segmentation tasks. We presented the work on *knowledge distillation* for semantic segmentation as a means of model compression for real-time deployment. This method definitely improves the performance on smaller and low performing network architectures thereby being readily deployable without an increase in space or computational cost. For moderate to high performing networks on the given cityscapes dataset, the proposed solution is to use larger a dataset to improve the model performance.

Network	Inference Time (ms)	Inference Time (ms) (after teacher-student training)
ENet	145	145
ICNet	170	170
ERFNet	310	310

Table 4.5 Inference time of all the student networks. (middle) time in ms after stand-alone cross entropy training, (right) time in ms after teacher-student training. These are the inference times on a single Nvidia GeForce GTX 1080Ti

Chapter 5

Conclusions

This chapter summarizes the thesis and along with it, presents the conclusions that were derived. A brief discussion on the future direction of this work is also presented.

5.1 Summary

This thesis starts with this discussion of the theory of semantic segmentation and how and why it is highly relevant in the current time of heavily invested research in autonomous vehicles. Chapter 2 was dedicated to explaining the concepts of *semantic segmentation*, *active learning* and *knowledge distillation*. The topic of segmentation was discussed, starting with the older traditional methods formulated to address the problem and we continued to discuss the latest deep learning methods which hold the current state-of-the-art in solving this problem. The currently available datasets for solving semantic segmentation were presented as a brief section. We also detailed how annotations for such datasets are obtained by detailing the standard polygon annotation tool. We explained how time-consuming the fine annotations for these datasets are and theorized that we needed to reduce the annotation load without compromising on the performance. The evaluation metric, intersection over union, IoU, was presented to help understand how the results of a segmentation method are evaluated. The concept of active learning was explained to show that it was an important strategy to reduce the annotation load while achieving higher performance. The relevant active learning methods used subsequently in the thesis were also introduced. A study of existing literature for active learning and semantic segmentation were presented. We also introduced the concept of knowledge distillation, a method to distill the knowledge from a larger and better performing network to a smaller and poorly performing network. We explained how the knowledge can be distilled using the better performing network as a supervision for the poorly performing network. Previous applications of this method showed that the performance of the smaller network improves with the supervision from the larger network when compared to training the smaller network alone. We also briefly discussed the applications of this concept in various other machine learning problems.

Chapter 3 discusses the proposed active learning methods to reduce the annotation load for semantic segmentation, focused on the task of road scene segmentation. The proposed methods were evaluated

on relevant road scene understanding datasets, Cityscapes, and Mapillary. The proposed active learning methods outperform the random sampling of data points in terms of performance. We have successfully adapted the model trained on one dataset, Cityscapes to the other dataset, Mapillary with only very little labelling while the performance of the method is almost close to that of the fully supervised IoU.

Chapter 4 is a work in the direction to improve the performance of fast segmentation models for autonomous driving situations. Fast segmentation models are optimized to perform inference at real-time, which often-times is at the cost of the accuracy of the mode. A study of *knowledge distillation* shows that the performance of larger models, which are very slow, can be distilled to the faster models, which will improve the segmentation performance. The problem was formally theorized for semantic segmentation and the results were presented on the Cityscapes dataset.

5.2 Conclusion

In Chapter 3, active learning was adapted for semantic segmentation. As the annotation cost for datasets for segmentation is very high, active learning is a suitable solution to reduce the annotation cost for segmentation. We proposed pixel-level, edge + pixel-level and superpixel level strategies for selecting the data-points for annotation in an image. All of these methods use entropy as a metric to pick the most uncertain pixels or superpixels to be sent to an oracle or a human annotator. The results were presented for autonomous driving datasets such as cityscapes and mapillary and showed its application on ICNet network. For cityscapes, we considered a subset of unlabeled images. An incremental batch-wise training approach was used for all the proposed strategies. In each proposed approach, we pick only 10% of pixels or superpixels and send them to an oracle for annotation. The rest of the pixel predictions are considered correct and are used as groundtruth for subsequent batch training. With the pixel-level approach, we achieved 86.5% of the performance of the fully-supervised method. Since edge-pixels usually provide more information, in the edge + pixel approach, we saw a slight improvement in results to 88.9% of fully supervised IOU. When we considered superpixels instead of pixels, this retained the annotation cost but also increased performance to 91.5% of fully supervised performance. A CRF post-processing after superpixel annotation improved the performance to 93.4% of a fully supervised counterpart. We also introduced a class-wise superpixel and CRF and it showed a 0.4% improvement over the previous method. We used a model trained on cityscapes dataset, and transferred the model to the mapillary dataset. With only 10% annotation of the whole mapillary dataset through the proposed superpixel and CRF approach, we observed that the performance of such model is almost at 90% of the fully-supervised model. All these experiments successfully showed that active learning is a useful strategy to reduce annotation cost and load for new data in semantic segmentation.

Chapter 4 presents the adaptation of knowledge distillation for semantic segmentation. The training dataset for all segmentation networks is the cityscapes dataset for road scene understanding. We proposed an ensemble of teacher networks, ICNet, ERFNet and deeplab. The ensemble is weighted according to its performance on the individual training data. The softmax output of the ensemble is saved

and the networks were not used in training of student network. We had empirically set the temperature values of softmax in the teacher networks for the ensemble so that they were in the same range. To train a student network, we used the teacher ensemble output as supervision with a KL-Divergence loss. We also trained the student network with the original groundtruth and used a cross-entropy loss. This joint training has been tested on three fast and real-time segmentation networks ICNet and ENet and also on ERFNet. We observed that the performance of ICNet increased by almost 4% compared to training the network only with groundtruth. On ENet, the performance improved by 3.5%. However, on ERFNet we saw no improvement in performance even after the student training with teacher supervision. This is due to the fact that this network is saturated for cityscapes dataset and using a larger dataset will improve the performance. The method proved to be effective in significant improvement in the performance of smaller and low performing networks.

The promising results of reduced data annotation load from chapter 3 and improved model performance through additional supervision in chapter 4 shows that a fully-supervised image-groundtruth pair type of training is not just a single available solution for better segmentation. It is necessary to look for smarter ways to obtain segmentation for road scene understanding. We hope that this thesis would help the readers understanding the importance of semantic segmentation and help them to push towards thinking beyond fully-supervised methods, thus driving research in that direction.

5.3 Future Directions

During the work for this thesis and discussions with other researchers, we have identified the future directions on improving these methods and other directions to solve problems related to semantic segmentation.

Better annotation methods:

Improving the way to annotate the uncertain regions. Currently proposed superpixel annotation is computationally expensive. A further study in annotation methods, in general, can give an idea about improving the annotation strategies. One such strategy briefly explored in during the thesis work was using strokes as annotation for the given regions.

Video supervision for active learning:

Since the temporal information from the video is a valuable source of information of how the different classes of the image interact, video supervision to predict intermediate frame annotation is a good direction for research.

Better dataset for knowledge distillation:

One of the drawbacks of the knowledge distillation methods is that it works well only on already low performing architectures. To adapt it to moderately performing segmentation architectures, a good idea is to use a bigger dataset.

Related Publications

1. **Region-Based Active Learning for Efficient Labelling in Semantic Segmentation**
   IEEE Winter Conference on Applications of Computer Vision (WACV), 2019
   Tejaswi Kasarla*, G Nagendar*, Guruprasad M. Hegde+, Vineeth N. Balasubramanian++, C.V. Jawahar*
   *CVIT, KCIS, International Institute of Information Technology, Hyderabad
   +Bosch Research and Technology Centre, India
   ++Indian Institute of Technology, Hyderabad

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.120.

[2] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages=3378–3385, year=2012, organization=IEEE.*

[3] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 2654–2662, 2014.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):2481–2495, 2017.

[5] P. Bakliwal, G. M. Hegde, and C. V. Jawahar. Collaborative contributions for better annotations. In *Proceedings of the VISIGRAPP*, 2017.

[6] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *Proceedings of the European Conference on Computer Vision*, pages 549–565. Springer, 2016.

[7] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 44–57. Springer, 2008.

[8] C. Bucilu, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541. ACM, 2006.

[9] H. Caesar, J. Uijlings, and V. Ferrari. Region-based semantic segmentation with end-to-end training. In *Proceedings of the European Conference on Computer Vision*, pages 381–397. Springer, 2016.

[10] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1209–1218, 2018.

[11] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5230–5238, 2017.

[12] F. M. Castro, M. J. Marin-Jimenez, N. Guil, C. Schmid, and K. Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[13] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 742–751, 2017.

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *Proceedings of the International Conference of Learning Representations (ICLR)*, 2015. URL `http://arxiv.org/abs/1412.7062`.

[15] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[17] Z. Chen, Y. Qian, and K. Yu. Sequence discriminative training for deep learning based acoustic keyword spotting. *Speech Communication*, 102:100–111, 2018.

[18] D. A. Cohn. Neural network exploration using optimal experiment design. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 679–686, 1994.

[19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[21] A. Freytag, E. Rodner, and J. Denzler. Selecting influential examples: Active learning with expected model output changes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 562–577. Springer, 2014.

[22] Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[23] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *Proceedings of the Asian Conference on Computer Vision*, pages 760–774. Springer, 2012.

[24] M. Gorriz, A. Carlier, E. Faure, and X. Giro-i Nieto. Cost-effective active learning for melanoma segmentation. *arXiv preprint arXiv:1711.09168*, 2017.

[25] M. Gorriz Blanch, A. Carlier, E. Faure, and X. Giro I Nieto. Cost-Effective Active Learning for Melanoma Segmentation (poster). In *Proceedings of the Advances in Neural Information Processing Systems (NIPS) Machine Learning for Health (ML4H) Workshop*, 2017.

[26] Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 593–600, 2008.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[28] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[29] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 417–424. ACM, 2006.

[30] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semisupervised svm batch mode active learning with applications to image retrieval. *ACM Transactions on Information Systems (TOIS)*, 27(3):16, 2009.

[31] A. Holub, P. Perona, and M. C. Burl. Entropy-based active learning for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–8, June 2008. doi: 10.1109/CVPRW.2008.4563068.

[32] S.-J. Huang, R. Jin, and Z.-H. Zhou. Active learning by querying informative and representative examples. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 892–900, 2010.

[33] S. D. Jain and K. Grauman. Click carving: Segmenting objects in video with point clicks. *arXiv preprint arXiv:1607.01115*, 2016.

[34] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification.

[35] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[36] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2017.

[37] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[38] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2011.

[39] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 424–437. Springer, 2010.

[40] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of International Conference on Machine Learning (ICML)*, 2001.

[41] X. Lan, X. Zhu, and S. Gong. Knowledge distillation by on-the-fly native ensemble. *arXiv preprint arXiv:1806.04606*, 2018.

[42] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings*, pages 148–156. Elsevier, 1994.

[43] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[44] X. Li, L. Wang, and E. Sung. Multilabel svm active learning for image classification. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume 4, pages 2207–2210. IEEE, 2004.

[45] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[46] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proceedings of the ACM SIGGRAPH*, 1995.

[47] J. Mun, K. Lee, J. Shin, and B. Han. Learning to specialize with knowledge distillation for visual question answering. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 8092–8102, 2018.

[48] G. Neuhold, T. Ollmann, S. R. Bul, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, Oct 2017.

[49] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[50] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1796–1804, 2015.

[51] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[52] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[53] N. Roy and A. Mccallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 441–448. Morgan Kaufmann, 2001.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[55] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[56] A. I. Schein and L. H. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007.

[57] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6 (1):1–114, 2012.

[58] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1070–1079. Association for Computational Linguistics, 2008.

[59] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 1289–1296, 2008.

[60] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[61] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3): 379–423, 1948.

[62] I. Shanu, C. Arora, and P. Singla. Min norm point algorithm for higher order mrf-map inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5365–5374, 2016.

[63] I. Shanu, C. Arora, and S. Maheshwari. Inference in higher order mrf-map problems with small and large cliques. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7883–7891, 2018.

[64] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[65] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[66] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM International Conference on Multimedia*, pages 107–118. ACM, 2001.

[67] R. Triebel, J. Stühmer, M. Souiai, and D. Cremers. Active online learning for interactive segmentation using sparse gaussian processes. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, pages 641–652. Springer, 2014.

[68] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019.

[69] A. Vezhnevets, J. M. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3162–3169, 2012.

[70] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2262–2269. IEEE, 2009.

[71] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1-2):97–114, 2014.

[72] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2017.

[73] T. Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *Computer Vision and Image Understanding*, 120:14–30, 2014.

[74] J. Xie, B. Shuai, J.-F. Hu, J. Lin, and W.-S. Zheng. Improving fast segmentation with teacher-student learning. *arXiv preprint arXiv:1810.08476*, 2018.

[75] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015.

[76] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3577. IEEE, 2012.

[77] Y. Yang and M. Loog. Active learning using uncertainty information. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 2646–2651. IEEE, 2016.

[78] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. Icnet for real-time semantic segmentation on high-resolution images. *arXiv preprint arXiv:1704.08545*, 2017.

[79] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[80] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537, 2015.