Abstract
- superpixel-based active learning
- click-based measurement of annotation costs
- class-balanced acquisition function

Introduction
Active learning (AL) offers one approach to address this annotation burden by selecting only the most informative samples for labeling.
Previous work suggests that region-based selection outperforms image-based selection due to the increase in data variability [23]; we thus focus on region-based AL in this work.
These clicks can be reduced (even avoided) if AL is conducted with boundary preserving regions, such that the annotator only needs to focus on assigning classes for each region.
This consideration of annotation costs motivates the use of (irregularly-shaped) superpixels in region-based AL [18, 30] instead of regularly-shaped squares or rectangles.
It remains unclear if a superpixel-based approach can indeed reduce annotation cost compared to the traditional “rectangle+polygon” based approach, because pixel-based annotation costs were used in the evaluation.
We address this question in this work, by revisiting the use of superpixels for region-based AL, performing analyses on the effect of region shape and size on region-based AL with more realistic, click-based measurements of annotation costs.

Our contributions can be summarized as follows:
• We revisit the superpixel-based approach for AL in semantic segmentation with realistic click based annotation cost taken into consideration, and demonstrate its effectiveness over the traditional “rectangle+polygon”-based approach.
• We investigate how region size affects the superpixel based scheme and the traditional rectangle-based scheme respectively, and show that the former outperforms for a wide range of region sizes.
• We propose a class-balanced acquisition function to further boost the performance of the superpixel-based approach by favoring the selection of informative samples from under-represented object categories.

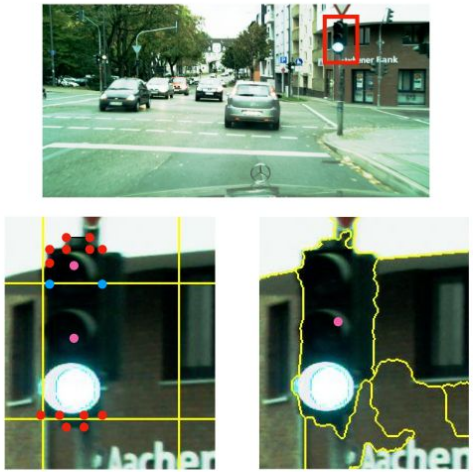

Figure 1: Annotating a traffic light by "rectangle+polygon" based approach (bottom left) *vs*. the superpixel-based approach (bottom right). The former requires quite a few polygon clicks (red dots), intersection clicks (blue dots) and class clicks (pink points), while the latter only requires a class click. If the annotation cost is measured in pixels, the two schemes perform closely, yet when measured in clicks, the latter is much more efficient.

2. Related Work

2.1. AL for Deep CNNs

Based on the the criteria used to select samples, AL approaches for deep CNNs can be grouped into three categories: uncertainty-based, diversity-based and the hybrid methods.
In this work we build upon uncertainty-based methods that have been shown to work well for region-based AL.

2.2. AL for CNN-based Semantic Segmentation

AL for semantic segmentation can be classified into image-level approaches [35, 31, 8] and region-level approaches [23, 4, 6] based on the granularity of sample selection.
The work most related to ours is [18] where superpixel-based selection is used for autonomous vehicle datasets like Cityscapes. However, [18] did not consider realistic annotation costs (clicks), and their results suggest that the advantage of superpixel-based selection without post-processing is marginal. It thus remains unclear if the superpixel-based approach can indeed reduce annotation costs compared to the widely-used “rectangle+polygon”based approach.
Our work addresses this gap.

2.3. Annotation Cost Measurement in AL

MetaBox+ [6] argued that box clicks were not necessarily required with a suitable labeling interface, while class clicks used to select a class label for each polygon should be considered. We consider the same types of clicks as in [6].

2.4. AL for Class Imbalance

These methods address the class imbalance at the training stage, while Ertekin et al. [10] showed that AL was capable of solving this problem implicitly by selecting informative samples to annotate at the data collection stage. Explicit handling of class imbalance in AL is often achieved by estimating the pseudo label of an unlabeled sample which decided whether this sample should be preferred in selection.
Kasarla1 et al. [18] represented each category with a feature vector, assigned each pixel to its most similar category and performed selection independently for each class. Similar to [18], we also tackle the class imbalance problem in semantic segmentation. However, instead of discretely assigning annotation budgets to each class, we take a soft weighting strategy based on the pseudo labels of superpixels. This avoids nearest neighbor search in highdimensional feature space and extra engineering for good features.

2.5. Superpixel Generation
Traditional superpixel generation algorithms can be broadly classified into graph-based and clustering-based approaches.
In this work, we stick to the traditional methods, i.e., SEEDS, to avoid additional labels for training.

3. Methodology
- 3.2 superpixel generation
- 3.3 class-balanced sampling
- 3.4 annotation cost measurement

3.1. Overall Framework
Given a set of unlabeled images, our method first divides each image into superpixels.
Next, we perform class balanced sampling to select a batch of informative samples, which are then annotated by an oracle.
Here, we use the ground truth semantic segmentation label to simulate such annotation process. Instead of the traditional polygon-based labeling, we use a dominant labeling scheme where each superpixel is assigned only a single class label.
The model is then retrained using all the data labeled so far and the process is repeated until the annotation budget is exhausted.

3.2. Superpixel Generation
In this work, we employ off-the-shelf SEEDS algorithm [34] due to its good performance in ensuring class coherency within each superpixel while maintaining object boundaries and ready-to-use interface.
In short, SEEDS is a clustering-based superpixel generation algorithm that begins with a uniform partition of image and iteratively refines the results by exchanging neighboring blocks in a coarse-to fine manner.

dominant labeling 할거면 superpixel의 object boundary 보존이 중요

Region Generation We consider two types of regions in the experiment:

(A) Rectangles (Rec): The image is uniformly divided into non-overlapping rectangles of size $m \times m$. We fix $m = 32$ in Section 4.3 and investigate the effect of different region sizes in Section 5.

(B) Superpixels (Sp): We employ SEEDS algorithm implemented in OpenCV to divide the image into non-overlapping superpixels. Before applying SEEEDS, we first apply histogram equalization to the image to improve its contrast, followed by converting to HSV color space. We use the following hyperparameters for the SEEDS algorithm: `prior` = 3, `num_levels` = 5, `num_histogram_bins` = 10, and `double_step` is enabled to slightly improve the quality of the superpixels. The number of superpixels is specified in such a way that it is the same as the number of rectangles when dividing the image using the Rec scheme.

3.3. Class-Balanced Sampling

$$s^* = \arg\max_{s \in \mathcal{U}_t} a(s, M_t). \quad (1)$$

$$u(x, M_t) = \frac{p(y = c^{sb}|x, M_t)}{p(y = c^b|x, M_t)}, \quad (2)$$

: uncertainty of pixel : pixelwise BvSB

$$u(s, M_t) = \frac{\sum_{x \in s} u(x, M_t)}{|\{x : x \in s\}|}. \quad (3)$$

: uncertainty of region s : average uncertainty of pixels within this region

$$Do(s) = \arg\max_{c \in \mathcal{C}} |\{x : l(x) == c \text{ and } x \in s\}|,$$
$$l(x) = \arg\max_{c \in \mathcal{C}} p(y = c|x, M_t), \quad (4)$$

: dominant label of region s

: dominant label of pixel

$$p(cls) = \frac{|\{s : Do(s) == cls\}|}{\sum_{c \in \mathcal{C}} |\{s : Do(s) == c\}|}. \quad (5)$$

: cls에 해당하는 region 개수 전체에서 존재 비율

$$a(s, M_t) = u(s, M_t) e^{-p(Do(s))}. \quad (6)$$

: class-balanced acquisition function : used to query the next sample

4.2. Benchmarking Methods

We benchmark with the following selection strategies:

(A) Random: This scheme randomly selects a region.

(B) Uncertainty: This scheme selects a region using the uncertainty-based acquisition function defined in Eq. (3).

(C) ClassBal: This scheme uses the class-balanced acquisition function defined in Eq. (6), *i.e.*, uncertainty weighted by the inverse of class posterior for region selection.

Given an annotation budget of K clicks, the algorithm to select a batch of samples is summarized in Algorithm 1.
Note that cost is used to simulate the actual annotation process where the annotator will stop labeling when the annotation budget is exhausted.

Algorithm 1: Batch-Mode Active Selection

Input : unlabeled set of regions $\mathcal{U}_t$, labeled set of regions $\mathcal{L}_{t-1}$ selected in previous batches, model M_t trained on $\mathcal{L}_{t-1}$, annotation budget of K clicks for batch t

Output: Output selected set of regions $\mathcal{B}_t$

$\mathcal{B}_t = \emptyset$;
$total\_cost = 0$;
while $total\_cost < K$ **do**
  $s^* = \arg\max_{s \in \mathcal{U}_t} a(s, M_t)$;
  $\mathcal{B}_t = \mathcal{B}_t \cup s^*$;
  $\mathcal{U}_t = \mathcal{U}_t \setminus s^*$;
  $total\_cost = total\_cost + cost(s^*)$;
end

3.4. Annotation Cost Measurement

In this work, we follow MetaBox+ [6] to consider three types of clicks that an annotator uses to label an image, the computation of which is detailed as below (refer to Fig.1 for an illustrative example for each click type).
Polygon clicks : clicks used for delineating the object boundaries : the number of polygon vertices needed for this region c_p
Class clicks : clicks that define the class for each annotated polygon : the number of connected components c_c
Intersection clicks : clicks incurred in the “rectangle+polygon”-based approach and caused by the intersection of the region boundaries and natural object boundaries : the total number of intersection points on its boundaries c_i

We consider two annotation schemes used to annotate a segmentation dataset and the involved clicks are discussed as below.
Precise labeling (Pr) $c_p + c_c + c_i$
Dominant labeling (Do) c_c

4.3. Experimental Results

In this section, we conduct extensive experiments with various combination of 1) selection strategies, 2) region types, and 3) annotation types to study the effect of each component.

1) Random, Uncertainty, ClassBal
2) Rectangles, Superpixels
3) Precise labeling, Dominant labeling

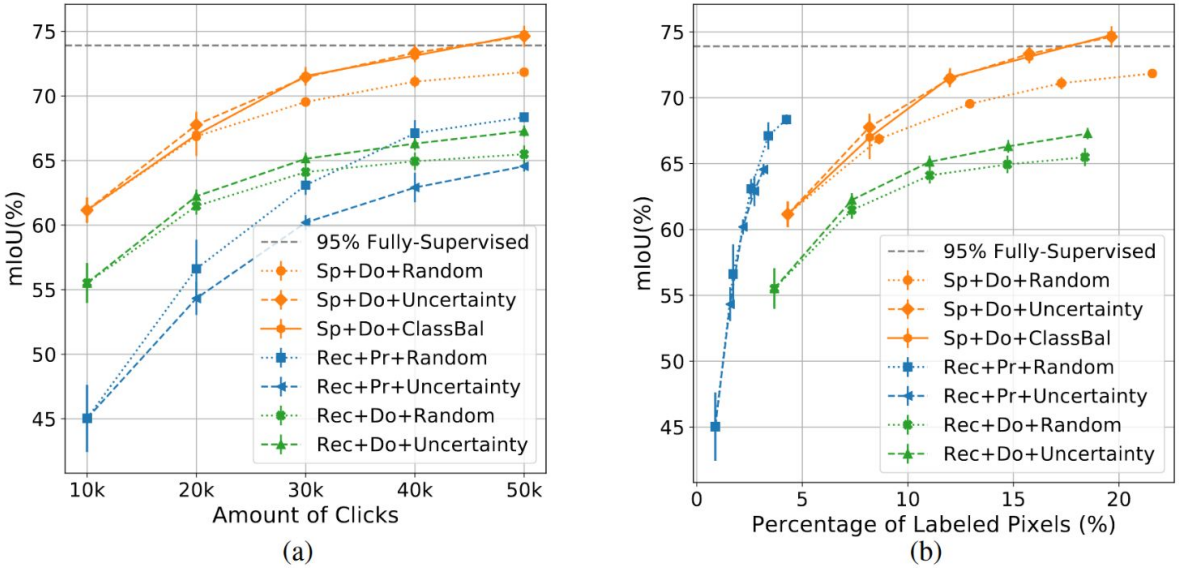

Figure 3: Active learning results on PASCAL VOC 2012. We report the mean and standard deviation of 3 runs. (a) Benchmarking at fixed amount of annotation budget measured in clicks. (b) Plot the same results with annotation cost measured in the percentage of labeled pixels.