

Size Constraints in Segmentation and Size Predictions

by

Xingye Fan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2024

© Xingye Fan 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Segment size, which one may equivalently refer to as volume, area, or cardinality, is uniquely defined by segmentation, e.g. by aggregating pixel-level predictions. On the other hand, size provides only a very weak constraint on segmentation. However, as this thesis observes, explicit size constraints are powerful cues for training segmentation models in weakly supervised settings, e.g. when only image-level class tags are provided or a fraction of pixels is labeled. We also observe that some standard unsupervised losses may have implicit size bias resulting in notable segmentation artifacts.

This thesis addresses three closely related problems regarding size constraints in segmentation and size predictions. First, we propose explicit size targets for training segmentation models without ground truth masks. We show approximate size targets predicted by human annotators result in segmentation quality on par with full pixel-precise supervision. The second contribution of this thesis is to show implicit size bias in standard unsupervised segmentation losses common in scribble supervision, e.g. mutual information or the Potts model. We show that this bias leads to the performance collapse as the amount of scribbles decreases. In contrast, our size-target supervision works well without any scribbles. Lastly, inspired by the enhanced segmentation outcomes achieved through size-target supervision, we explore the potential of deep models in predicting sizes directly.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Yuri Boykov, for his invaluable guidance, unwavering support, and encouragement throughout the duration of my master's research project. His expertise, patience, and insightful feedback have been instrumental in shaping the direction and quality of this project.

I extend my heartfelt appreciation to the members of my thesis committee, Olga Veksler and Hongyang Zhang.

I am grateful to my colleague, Zhongwen (Rex) Zhang, for his encouragement, support, and stimulating discussions throughout this journey.

Special thanks to my family for their unwavering love, encouragement, and understanding during this challenging endeavor.

Dedication

This is dedicated to the one I love.

Table of Contents

Author’s Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
2 Related work	8
2.1 Semantic segmentation	8
2.1.1 Supervision with full ground truth masks	8
2.1.2 Weak supervision: partial masks and image-level tags	14
2.2 Size constraint losses for segmentation	17
2.2.1 Size distribution	18
2.2.2 Unsupervised size constraints	18
2.2.3 Supervised size constraints	19

2.3	Unsupervised losses for segmentation	21
2.3.1	CRF regularization	21
2.3.2	Clustering techniques for segmentation	24
3	Size Targets for Segmentation	28
3.1	Size-target loss and its properties	29
3.2	Human-annotated size targets	32
3.3	Synthetic size targets with additive noise models	34
3.3.1	Basic linear AE model	35
3.3.2	Advanced AE models	36
3.4	Experimental results	40
3.4.1	Experimental settings	40
3.4.2	Approximate and uninformative size targets	42
3.4.3	Enhancing scribble-based segmentation with size targets	44
3.4.4	Comparison with the state-of-the-art methods	44
3.4.5	Medical data: size-target vs. size-barrier	46
3.4.6	Segmentation with human-annotated size	49
4	Size Bias in MI Loss and Weakly-supervised Segmentation	51
4.1	Mutual information loss	52
4.1.1	Formulation and related work	52
4.1.2	Uniform size bias in the context of low-level segmentation	56
4.2	MI for deep segmentation	58
4.2.1	Conditional posterior model and MI loss	59
4.2.2	Unbiased MI loss	61
4.3	Experimental results	62
4.3.1	Experimental settings	62
4.3.2	Scribbles vs. size targets	63

5	Size Prediction: Between Regression and Classification	68
5.1	Architectures	69
5.1.1	Naive approaches: regression and “soft” classification	69
5.1.2	From averaging features to averaging predictions	70
5.2	Losses for supervision with “soft” distribution targets	71
5.3	Experimental results	71
5.3.1	Experimental settings	71
5.3.2	Size prediction quality	72
6	Conclusion and Future Work	74
	References	76

List of Figures

1.1	Examples of semantic segmentation	2
1.2	Example of different forms of supervision	3
2.1	Fully convolutional neural networks	10
2.2	An illustration of SegNet using encoder-decoder architecture	11
2.3	Examples of supervision with varying scribble lengths	15
2.4	Illustration of class activation mapping (CAM)	16
2.5	An illustration of the SEC training framework	17
2.6	Illustration of quadratic size constraint loss	21
2.7	Graphical model for low-level binary segmentation	22
2.8	Illustration of sparse and dense CRF	23
3.1	Labeling speed and accuracy for different forms of supervision on PASCAL	29
3.2	Semantic segmentation with DeepLabV3+(R101) models	30
3.3	Forward vs. reverse KL divergence	31
3.4	Size-target supervision annotation tool	33
3.5	Quality of human annotations	34
3.6	Comparison of the conditional mean absolute deviation ($MAD(\epsilon \hat{v})$) between the linear AE model and human annotations	37
3.7	Alpha value selection for the Renyi model	39
3.8	Comparison of the $MAD(\epsilon \hat{v})$ and $mRE(\hat{v})$ between the AE models and human annotation	40

3.9	Segmentation accuracy for our approximate size-target supervision on PASCAL’s training and validation data	42
3.10	Segmentation accuracy on PASCAL validation set for scribble supervision of varying lengths	43
3.11	Segmentation examples using size-target supervision	46
3.12	Size-targets vs. size-barriers on ACDC dataset	47
3.13	Qualitative segmentation results on ACDC dataset	48
3.14	Segmentation accuracy with size-target supervision on three PASCAL classes (cat, dog, bird)	50
4.1	Relationship between entropy, conditional entropy, and mutual information	52
4.2	Volumetric bias of MI loss in low-level segmentation	56
4.3	Performance comparison: scribbles vs. size targets	66
4.4	Qualitative examples of segmentation models trained with different supervision and regularization losses	67

List of Tables

3.1	Human-annotated size targets: speed and mRE	33
3.2	Semantic segmentation results (mIOU%) on PASCAL and COCO	45
3.3	Binary semantic segmentation accuracy of three classes (cat, dog, bird) . .	49
5.1	Comparison of size prediction models	73

Chapter 1

Introduction

Semantic image segmentation, a fundamental problem in computer vision, involves dividing an image into parts and classifying each segment with predefined categories. This task goes beyond detecting objects by also understanding the context and elements within an image, making it essential for machines to visually perceive the world as humans do. Semantic segmentation has a wide range of applications. In virtual reality, it enables immersive environments by accurately mapping virtual objects to real-world scenes. Autonomous driving benefits from precise environment perception for safe navigation. In medical imaging, it aids in the detailed analysis and diagnosis of diseases, showing the profound impact of this technology.

Initially, traditional machine learning methods like random forests and support vector machines were employed, leveraging handcrafted features. However, the advent of deep learning [70] has revolutionized this domain. Convolutional Neural Networks (CNNs) [68] optimizes feature extraction and classification in a unified framework. Fully Convolutional Networks (FCNs) [81] further advanced this by enabling pixel-wise predictions without the need for image resizing. Architectures like U-Net [90] and SegNet [11] have been particularly successful, offering precise segmentation for specific applications like medical imaging through their unique network designs. Despite progress, challenges persist. Fully supervised semantic segmentation requires detailed labeling of every pixel. High annotation cost limits the availability of high-quality training data. To overcome this challenge, current research focuses on developing weakly supervised semantic segmentation technologies that leverage much cheaper supervision to achieve comparable results attained by full pixel-precise supervision. Figure 1.1 shows some examples of weakly-supervised semantic segmentation results.



Figure 1.1: Examples of semantic segmentation. The segmentation results are produced by the ResNet101-based DeeplabV3+ model trained with size targets. The top row shows three input images and the bottom row shows their corresponding semantic segmentation predictions. Class labels are represented by different colors in the prediction masks. For instance, the bottle, person, sheep, and bike in the images are indicated in purple, pink, brown, and green. The images are from the Pascal dataset [42].

Many forms of weak supervision (shown in Figure 1.2) and their corresponding training paradigms are extensively studied to reduce the annotation cost while maintaining segmentation accuracy. *Bounding boxes* [74], which are rectangular outlines around objects of interest, provide rough spatial information without detailed boundaries. Models are trained to refine these coarse location cues, often using GrabCut [20] or similar algorithms to sharpen boundaries. *Scribble* supervision [77] draws scribbles over objects indicating class presence. The *Point* supervision [12] is a special case of scribble supervision, which applies one click or single point to indicate the category of each object. *Image-level super-*

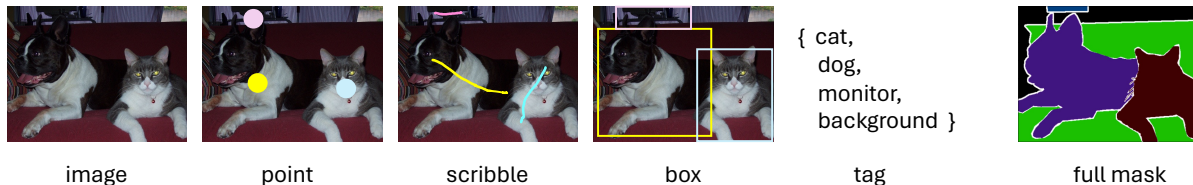


Figure 1.2: Example of different forms of supervision. The left is the image and the rest shows its various forms of supervision including points, scribbles, bounding boxes, tags, and full ground truth masks. The image and ground truth mask are from the Pascal dataset [42].

vision [62] annotates the classes or tags presented in the image. Many works [84, 117, 111] also use hybrid forms of weak supervision for semantic segmentation.

Image-level or tag supervision is promising due to its low labeling cost. According to [13], the time required for annotating an image varies significantly with the type of labeling: tagging a class takes merely 1 second, whereas marking a point, drawing a bounding box, and creating a labeling mask for each instance require 2.4 seconds, 10.2 seconds, and 78 seconds, respectively.

Despite the easy accessibility of tag supervision, there are many challenges in tag-supervised segmentation. Image-level labels indicate the presence of classes in an image but not their locations and sizes. This ambiguity makes it difficult for models to accurately assign class labels to each pixel. Without pixel-level annotations, distinguishing between foreground objects and the background becomes challenging, especially in cluttered scenes. Models may incorrectly include background regions as part of the object (over-segmentation) or miss parts of the object, assuming them to be background (under-segmentation).

To mitigate the lack of location information, techniques like class activation maps (CAM) [127] are used to identify the discriminative regions of objects and generate pixel-level pseudo labels. These pseudo-labels can provide location cues to help the model identify objects and mitigate the ambiguity of tag supervision. However, this approach is not ideal. On one hand, a small portion of false pseudo-labels may mislead the model to overfit the errors and be detrimental to the performance. On the other hand, the multi-stage process of generating pseudo-labels and subsequently training segmentation models with these pseudo-labels results in a less efficient training procedure compared to the more streamlined one-stage (end-to-end) training approach.

Another key challenge associated with all forms of weak supervision is the absence

of size information for each image. Weakly supervised methods typically rely on less granular annotations, such as image-level tags, bounding boxes, or scribbles, which do not provide size information. This weakness makes it difficult for models to accurately learn and predict the sizes of objects, a crucial aspect for many applications where precise segmentation is required. Without detailed size information, models are prone to making errors in segmenting objects, often resulting in over- or under-segmentation. To mitigate this issue, some research efforts [86, 57] have focused on incorporating size constraints into the training process. By integrating additional guidance on the expected sizes of objects, these approaches aim to refine the model’s ability to gauge object proportions more accurately during segmentation.

In addition, unsupervised loss functions [105, 103] are extensively used in weakly supervised semantic segmentation tasks to enhance and refine the model’s predictions. These unsupervised techniques play a crucial role in improving the accuracy of segmentation by leveraging the inherent structure and appearance of the image data, without the need for supervision. This approach bridges the gap between the model’s predictions and the actual image content, facilitating a more accurate and visually coherent segmentation outcome.

Motivation and contributions

In this thesis, we address three problems to tackle the challenge of image-level semantic segmentation. Following a background review, this section summarizes the problems we have addressed and discusses our motivations.

In Chapter 3, we propose a new form of supervision, size targets, for semantic segmentation. It can be seen as an extension of tag supervision. Tag supervision only indicates the classes in an image, while size-target supervision also labels the approximate segment size. With modest extra information, we can achieve segmentation performance on par with full supervision. It is hard to train high-performance segmentation models with tag supervision. Typical tag-based methods heavily rely on class activation maps (CAM) techniques [31, 30] to generate pixel-wise pseudo labels. Subsequently, the pseudo labels are used for segmentation model training. Although CAM can effectively improve segmentation accuracy, it complicates the training procedure. To further enhance the performance with tags, other architectural modifications are usually employed. Using size targets not only largely enhances the segmentation accuracy but also greatly simplifies the training. We propose the forward KL divergence as the loss function, enabling the segmentation models to be trained end-to-end. Moreover, we only use standard segmentation models commonly used in full supervision to achieve such high performance.

Exact size targets may not be practical without pixel-precise segmentation masks. Therefore, we test the robustness of the size-target approach and find that approximate size targets are enough for high-performance segmentation. Such approximate size targets are easy to obtain. We propose a labeling interface to assist the annotators. We evaluate the size-target approach with human-annotated sizes and synthetic approximate sizes. Synthetic sizes are generated by corrupting the exact size targets using Gaussian noise. Different noise models are studied to best approximate human errors.

In Chapter 4, we propose a new unsupervised loss for semantic segmentation based on *mutual information* (MI). Mutual information between two random variables measures the reduced uncertainty of one random variable given the other. Maximizing mutual information is a reasonable principle for clustering. It opens doors for generative and discriminative clustering methods. We are motivated by the generative clustering techniques commonly used in low-level segmentation [23, 93, 104]. Generative methods perform the clustering by estimating the data density. In our context, the data is the raw colors in an image. As generative methods are underexplored for semantic segmentation, we want to fill this gap.

Our MI loss can be seen as an entropy clustering criterion, which minimizes the entropy of colors within each segment. The color density is estimated with a density model. The MI loss is optimized with the Expectation-Maximization (EM) based method by alternating two steps. The first step conducts the density model fitting while the segmentation predictions are fixed. In the second step, the density model is fixed, and the network parameters are optimized with gradient descent to produce more accurate segmentation predictions.

According to the prior work [56], the MI loss prefers equal segment size. Such equal-size bias can be revealed by decomposing the MI loss into three terms: entropy of average segmentation predictions, conditional entropy of segmentation given color, and the KL divergence between the density model and true data density. The equal-size bias is caused by the entropy of average segmentation predictions. Another prior work fixed the bias and proposed an unbiased MI [16]. We evaluate both biased and unbiased MI loss with tag, scribble, and size-target supervision.

In chapter 5, we investigate the potential of neural networks for size prediction. The effectiveness of size-target supervision motivates us to automate the size-labeling process using neural networks.

We start with the native algorithms: using standard classification or regression network architectures for size predictions. As size distribution is continuous on the probability simplex, it is a problem that incorporates both classification and regression. We find that using a classification or regression model cannot effectively solve the size prediction problem. To improve the size prediction accuracy, we propose an architectural modification. Specifi-

cally, we change from averaging deep features to averaging predictions. This modification avoids information loss through deep feature averaging. We also evaluate a variety of loss functions for size prediction.

Our key contributions are summarized as follows:

- We propose a new form of image-level supervision, size targets, which represent the volumetric distribution of the normalized segment cardinality. Size targets can be seen as an extension of the standard tag supervision. Image tags are in one-to-one correspondence with the positive size targets. Size-target supervision is cheap and is easy to estimate for human annotators compared to pixel-precise ground truth masks. On top of size targets, we propose the zero-avoiding variant of KL divergence as a general training loss for any segmentation architecture. Our general end-to-end approach is much simpler than recent CAM-based methods which typically require architectural modifications or multi-stage training schemes. The size-target approach achieves segmentation quality on par with full supervision on the Pascal VOC dataset. Moreover, our training loss shows significant robustness to size errors. A certain amount of error may even improve the generalization quality of segmentation networks.
- We analyze implicit size biases in standard unsupervised loss functions for low-level segmentation: mutual information (MI) and the Potts model. In particular, we discuss MI loss that has the bias to equal-size segments or uniform volumetric distribution, an unbiased formulation of MI loss with no size bias, and the Potts model that has “shrinking” bias, which can be seen as the bias to one-hot volumetric distribution. While the Potts model has been previously used for weakly-supervised semantic segmentation with partial pixel masks (scribbles) [105], one of our contributions is to test MI in a similar “deep” context. In part, this is motivated by different forms of implicit size bias in these losses.
- We empirically evaluate unsupervised segmentation losses (two forms of MI and the Potts) for weakly-supervised segmentation on PASCAL using different levels of supervision. We gradually decrease the number of scribbles on pixels and compare them to image-level supervision using class tags only. We observe that the performance significantly decreases as the amount of scribbles decreases. It completely collapses when only image-level tags are used: the background class takes over as an easy solution, even for the MI biased to equal-size segments. This motivates stronger cardinality constraints [86]. We evaluated *log-barrier* cardinality constraints to prevent the collapse for the trivial solution. Figure 4.3 presents our comprehensive evaluation of unsupervised losses (with implicit and explicit size biases) for weakly-supervised

segmentation across different supervision levels, including approximate image-level size targets. The latter works at near full supervision level using a combination of our size target loss and unsupervised MI loss.

- The effectiveness of size targets motivates us to automate size target annotation. We investigate the potential of neural networks for size prediction. Size prediction is a hybrid task incorporating regression and classification. We propose an architectural modification on top of the standard image classification networks, shifting from averaging features to averaging predictions. The architecture modification significantly improves the size prediction accuracy.

We will detail each contribution in the following chapters. In Chapter 3, we propose the size-target supervision. In Chapter 4, we employ the MI loss for semantic segmentation. We also reassess the size-target approach and conduct a comprehensive evaluation combining size targets and MI. Chapter 5 explains our approach to size prediction.

Chapter 2

Related work

The structure of Chapter 2 is outlined as follows: In Section 2.1, we will introduce fully- and weakly-supervised semantic segmentation. We will detail the architectural designs, loss functions, optimization strategies, evaluation metrics, and different forms of supervision. In Section 2.2, we focus on segmentation methods that incorporate size constraints. Our proposed size-target approach falls under this category, thus it's crucial to elaborate on size constraints to clarify how our method differs. In Section 2.3, we will discuss two widely used unsupervised losses for semantic segmentation: conditional random field (CRF) and clustering techniques.

2.1 Semantic segmentation

2.1.1 Supervision with full ground truth masks

Algorithm

Fully-supervised semantic segmentation algorithms are trained on datasets containing pixel-precise ground truth masks, see the right image in Figure 1.2. Every pixel on the ground truth mask is annotated. For visualization purposes, different classes are indicated in distinct colors. Let I be an image of dimension $h \times w \times c$, where h and w are the height and width of the image, and c is the channel number. The $Y \in \mathbb{R}^{h \times w \times K}$ is the full ground truth mask, where K is the number of predefined classes. The $\mathcal{X}_{train}$ represents a training dataset containing N data pairs $\{(I_1, Y_1), \dots, (I_N, Y_N)\}$. Segmentation models are deep

neural networks ϕ_θ with parameters θ . The segmentation model ϕ_θ maps input image I_i to the segmentation prediction: $S_i = \phi_\theta(I_i)$. Model parameters θ are learned on the training dataset $\mathcal{X}_{train}$ with gradient-based optimization methods, such as *stochastic gradient descent* (SGD), *RMS-Prop* [106], or *Adam* [60]. The trained model can subsequently be used to predict unseen data on a testing dataset $\mathcal{X}_{test}$.

Neural network architecture

Various architectures were studied to improve the accuracy and efficiency of semantic segmentation [81, 90, 11, 26, 27, 28, 115, 101]. We will primarily focus on the most prevalent CNN-based architectures. Despite the recent rise in the popularity of vision transformers [115, 101], architectural innovations are beyond the scope of this thesis.

Due to the extraordinary results achieved by convolutional neural networks (CNN) on image classification [66], CNNs were later adapted for segmentation tasks. Classification networks typically consist of convolutional layers combined with non-linear activation functions (e.g. ReLU (2.2)) and pooling layers to reduce spatial resolution. Following the convolutional layers, the intermediate features are flattened or averaged to convert the 2D feature maps into 1D feature vectors. These condensed features are then passed through linear (fully-connected) layers to generate the deep feature representation, denoted by f . A linear classifier with weights $W = \{w^1, \dots, w^K\}$ is employed for K-way classification. The logit for each class is obtained as follows:

$$l^k = \langle w^k, f \rangle, \quad (2.1)$$

where $l^k \in \mathbb{R}$ is the logit of the k th class. Logits are mapped onto the probability simplex through an activation function $\mathbb{R}^K \rightarrow \Delta^K$, transforming the real-valued vector $\mathbb{R}^K$ into a probability distribution over K classes. For binary classification tasks, the Sigmoid function (2.3) is typically employed to convert logits into probabilities between 0 and 1. In the context of multi-class classification, the SoftMax function (2.4) is utilized to generate probabilities for multiple classes. The generated categorical distribution $\{\sigma^1, \dots, \sigma^K\}$ is interpreted as posterior $\Pr(k|I)$.

$$f(x) = \max(x, 0) \quad \text{ReLU} \quad (2.2)$$

$$s(x) = \frac{1}{1 + e^{-x}} \quad \text{Sigmoid} \quad (2.3)$$

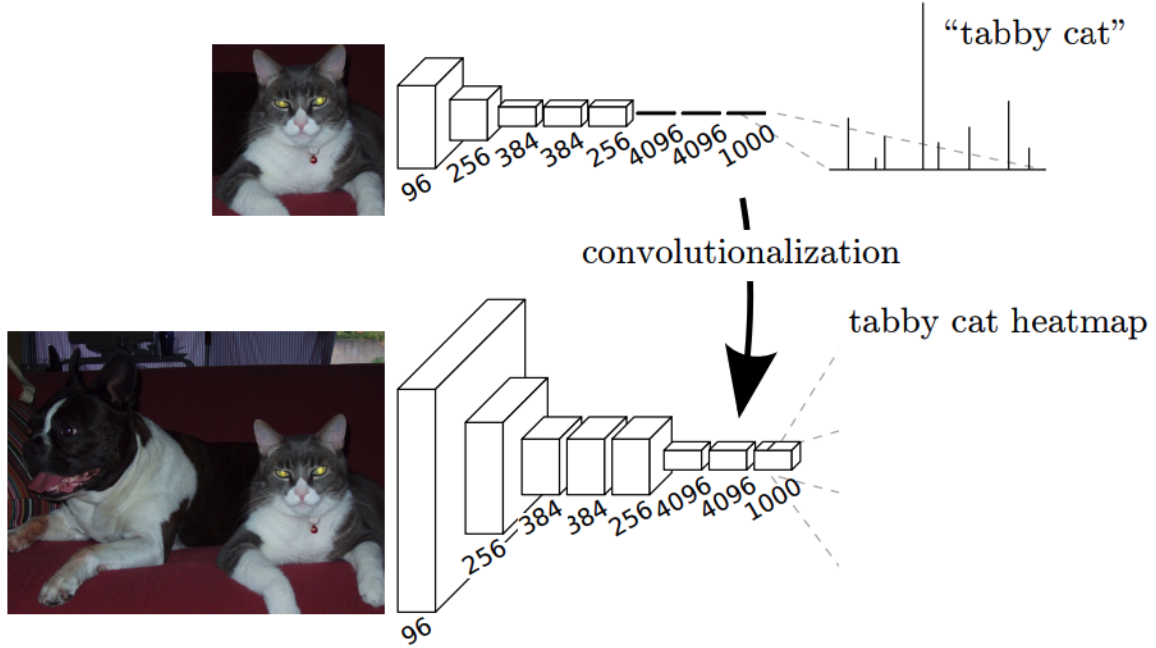


Figure 2.1: Fully convolutional neural networks for semantic segmentation. The upper diagram is a typical image classification neural network consisting of convolutional layers (represented by boxes) and fully connected layers (represented by lines). The upper right corner shows the predicted distribution of the given image. The convolutionalization demonstrates the transformation from a classification network to a fully convolutional network, enabling the network to output a heatmap. Image from [81]

$$\sigma_i(x) = \frac{e_i^x}{\sum_j^K e_j^x} \quad \text{SoftMax} \quad (2.4)$$

Image classification methods are extended to segmentation for pixel-level classification. As convolutional kernel can be applied to inputs of any size, Darrell et al. [81] dropped the fully-connected layers and proposed fully convolutional networks (FCN) for semantic segmentation, see Figure 2.1. Such architecture was pioneered by [1] for digit recognition. The transition to convolutional layers enables the network to retain spatial resolution. The output resolution is at a reduced scale compared to the original image. Directly up-sampling the outputs to the size of the original image yields coarse predictions. To generate fine-grained spatial details delineating the boundaries of objects within images, encoder-decoder architectures [11, 90] are widely used, see Figure 2.2.

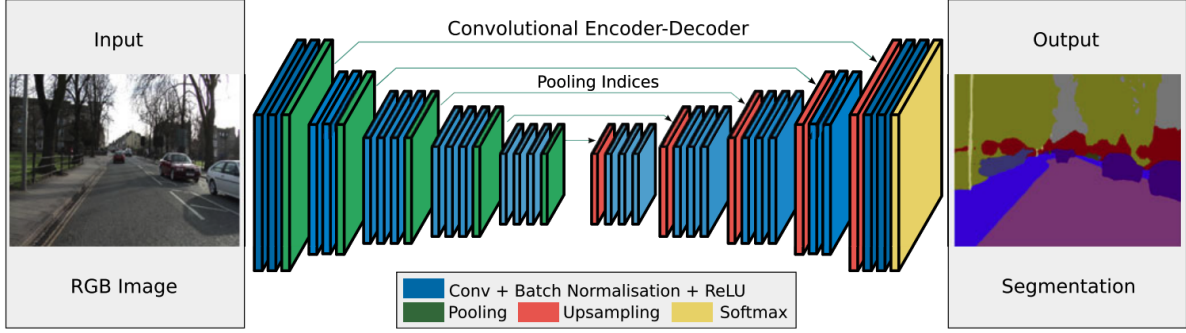


Figure 2.2: An illustration of SegNet using encoder-decoder architecture. Colored boxes represent different layers in the segmentation architecture. The size of the boxes represents the resolution of encoded feature maps. Image from [11]

The network encoder progressively reduces the spatial dimensions of the input image through a series of convolutions and pooling operations. This process encodes the input image into a compact, high-dimensional feature representation. The encoder captures the context in the image but reduces the spatial resolution. The decoder of the network gradually recovers the spatial dimensions and details from the encoded representation. This is achieved through a series of up-sampling and convolution operations. The decoder aims to map the high-level, abstract features back to the original spatial dimensions, recovering the spatial information necessary for precise segmentation. Finally, the decoder output passes through a linear classifier that adjusts the number of output channels to match the number of classes. This layer typically uses a SoftMax or Sigmoid activation function to generate the class probabilities for each pixel. In some cases, additional post-processing steps like conditional random fields (CRF) are applied to refine the segmentation maps and improve the boundaries precision [26].

Similar to (2.1) The logit of a pixel p for each class is computed with

$$l_p^k = \langle w^k, f_p \rangle, \quad (2.5)$$

where f_p represents deep features of pixel p , and w^k denotes the linear classifier weights of the k th class. Logits are converted to categorical distributions $\{S_p^1, \dots, S_p^K\}$ by SoftMax function (2.4).

Losses for fully-supervised semantic segmentation

To train segmentation models effectively, various loss functions have been designed to measure the discrepancy between the predicted segmentation maps and the pixel-precise ground truth annotations. Below are some of the commonly used loss functions in fully-supervised semantic segmentation:

- Cross-entropy loss: *Cross-entropy* measures the difference between two probability distributions. Optimizing the cross-entropy loss makes the predicted segmentation mask S close to the actual mask Y . Since Y_p is a one-hot distribution (i.e. each pixel is only associated with one class), the cross-entropy can be formulated as negative log-likelihood, where y_p is the ground truth class for pixel p .

$$\mathcal{L}_{CE} = - \sum_{p \in \Omega} \sum_{k=1}^K Y_p^k \log S_p^k = - \sum_{p \in \Omega} \log S_p^{y_p} \quad (2.6)$$

- Binary cross-entropy loss: *Binary cross-entropy* loss is a special case of cross-entropy loss. In binary classification, each output prediction is supposed to be a probability value ranging from 0 to 1, indicating the likelihood of the input belonging to the positive class.

$$\mathcal{L}_{BCE} = - \sum_{p \in \Omega} (Y_p \log S_p + (1 - Y_p) \log(1 - S_p)) \quad (2.7)$$

- Weighted cross-entropy loss: *Weighted cross-entropy* loss is a variant of the standard cross-entropy loss (2.6) that assigns different weights to classes in the dataset. It is useful in dealing with imbalanced datasets where some classes are underrepresented compared to others. By assigning higher weights to the underrepresented classes, the loss function penalizes misclassifications of these classes more heavily than those of the over-represented classes. This approach encourages the model to focus more on learning the under-represented classes.

$$\mathcal{L}_{wCE} = \sum_{p \in \Omega} \sum_{k=1}^K w^k Y_p^k \log S_p^k = \sum_{p \in \Omega} -w^{y_p} \log S_p^{y_p}, \quad (2.8)$$

Evaluation criteria

Intersection over union (IoU) is commonly used to measure the accuracy of segmentation results. Intersection over union, also known as the *Jaccard coefficient*, measures the discrepancy between the predicted segmentation masks and the ground truth masks. It is

calculated as the ratio between the intersection area and the union area of two masks:

$$J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}, \quad (2.9)$$

where $X, Y \in \Omega$ are two subsets of pixels. Intersection over union is easy to interpret, as it aligns well with human perception of segmentation quality. It measures the spatial overlap between predicted and ground truth masks, yielding a number between 0 and 1, often represented as a percentage. For multi-class semantic segmentation tasks, the *Mean Intersection over Union* (mIoU) is a common evaluation metric. It takes the IoU score of each class and averages them over all classes, providing a comprehensive measure of segmentation performance.

$$mIoU = \frac{1}{K} \sum_{k=1}^K IoU_k \quad (2.10)$$

Dice coefficient (DSC) is commonly used in medical segmentation tasks. Dice coefficient focuses more on the spatial overlap between the predicted and ground truth segmentation masks. DSC is defined as follows

$$DSC = \frac{|X \cap Y|}{|X| + |Y|}, \quad (2.11)$$

where $X, Y \in \Omega$ are two subsets of pixels. Dice coefficient produces values between 0 and 1, where 1 indicates perfect overlap between the predicted and ground truth segmentation masks. This simplicity makes it easy to interpret and compare segmentation results across different algorithms and datasets.

2.1.2 Weak supervision: partial masks and image-level tags

Weakly supervised techniques rely on weaker forms of annotations such as partial ground truth masks (scribbles/seeds), image-level tags, or bounding boxes. This section mainly focuses on introducing partial ground truth masks and image-level tags, which are more relevant to this thesis.

Partial masks.

Partial mask supervision only annotates a small portion of pixels with simple, scribble-like lines, see Figure 1.2. These scribbles are drawn over the objects of interest within the images, providing a rough guide of the object’s location and shape. Scribbles can be quickly drawn by annotators, significantly lowering the cost of data annotation. Despite the simplicity, deep neural networks can be trained to interpret these sparse annotations and infer the full object segments. Training with scribble supervision typically uses partial cross-entropy loss (pCE) [43, 113]:

$$\mathcal{L}_{pce} = \sum_{p \in \Omega_L} -\log S_p^{y_p}, \quad (2.12)$$

where Ω_L is the set of sparsely labeled pixels or seeds. Partial cross-entropy loss is only calculated among the annotated pixels, encouraging the model to match these known labels while inferring the labels of unlabeled pixels through context.

To enhance the performance of segmentation using partial masks, one approach is to expand the sparse partial masks to full pseudo label masks with graphical models [77]. However, the errors in the full pseudo label masks may mislead the training and overfit mistakes. Another approach is to directly employ a regularization loss for segmentation model training [103, 105]. Regularization losses will be detailed in Section 2.3.

To assess the resilience of algorithms with the decrease in the amount of scribbles, it’s common to evaluate the performance across varying lengths of scribbles. This approach helps to understand how well the algorithms can adapt to differing extents of supervision. As shown in Figure 2.3, the 100% length corresponds to full-scribble supervision. The scribble lengths in other images are shortened. In the extreme case (0%) only one pixel is labeled for each object (including the background).



Figure 2.3: Examples of supervision with varying scribble lengths from 100% full-length scribbles to 0% one click. The colors of strokes represent their corresponding classes. Scribbles are shortened from the two endpoints. Image from [105]

Image-level tags.

Image-level tags are more cost-effective than scribbles. Only class tags are labeled. These tags indicate the present classes in an image, see Figure 1.2.

It is challenging for semantic segmentation models to make pixel-level predictions with only image-level labels. One popular approach for tag-supervised semantic segmentation is to generate pixel-level pseudo labels and use them as partial ground truth masks. Pseudo labels are typically generated by the class activation maps (CAM) technique.

CAM [127] is a technique used in image classification to visually present which parts of the image are most influential to the model’s predictions. Essentially, CAM provides an insight into the model’s decision-making process. CAM is generated from the feature maps f of the last convolutional layer. The idea is to weigh these feature maps with the linear classifier (discriminator) weights w^k . CAM is formulated as

$$M_p^k = \langle w^k, f_p \rangle, \quad (2.13)$$

where M is the class activation maps before normalization and the Relu (2.2) operation, w^k is the weights of the k th class, and f_p is the features of pixel p . Equation (2.13) applies the learned weights to the pixel-level feature maps rather than the class features in (2.5). CAM is a heat map highlighting the important areas for that class. CAM helps to understand why the network predicts certain classes, increasing the transparency of the model’s decision-making process. In weakly supervised semantic segmentation, CAM can be used for object localization tasks without requiring detailed pixel-level annotations. For instance, Kolesnikov et al. [62] proposed the seed, expand, and constrain (SEC) training framework using pseudo labels generated by CAM, see Figure 2.5 for details.

Training schemes can be categorized into two types: single-stage and multi-stage. Single-stage methods train segmentation models in an end-to-end manner, where models perform the task from the input data to the output predictions in one single forward pass. The training does not require separate phases for different aspects of the task. Single-stage models [10, 94] are known for their training efficiency, making them easier to train and deploy. Here is some related work for single-stage tag-supervised semantic segmentation. Araslanov et al. [10] proposed three losses to improve the properties desired in weakly supervised segmentation including completeness, local consistency, and semantic fidelity. Ru et al. [94] employed the vision transformer backbone and trained a side affinity prediction branch to improve CAM via random walk.

Multi-stage methods [112, 62, 54, 24, 41, 128, 76, 75] involve more complex procedures where the training occurs in separate, distinct steps. Each stage might focus on different aspects of the learning task such as pre-training on related tasks, gradually introducing complexity, or refining certain parts of the model based on intermediate outputs. Multi-stage approaches are often more complex and may offer higher accuracy. Semantic segmentation with tag supervision typically adopts a multi-stage method. The first stage

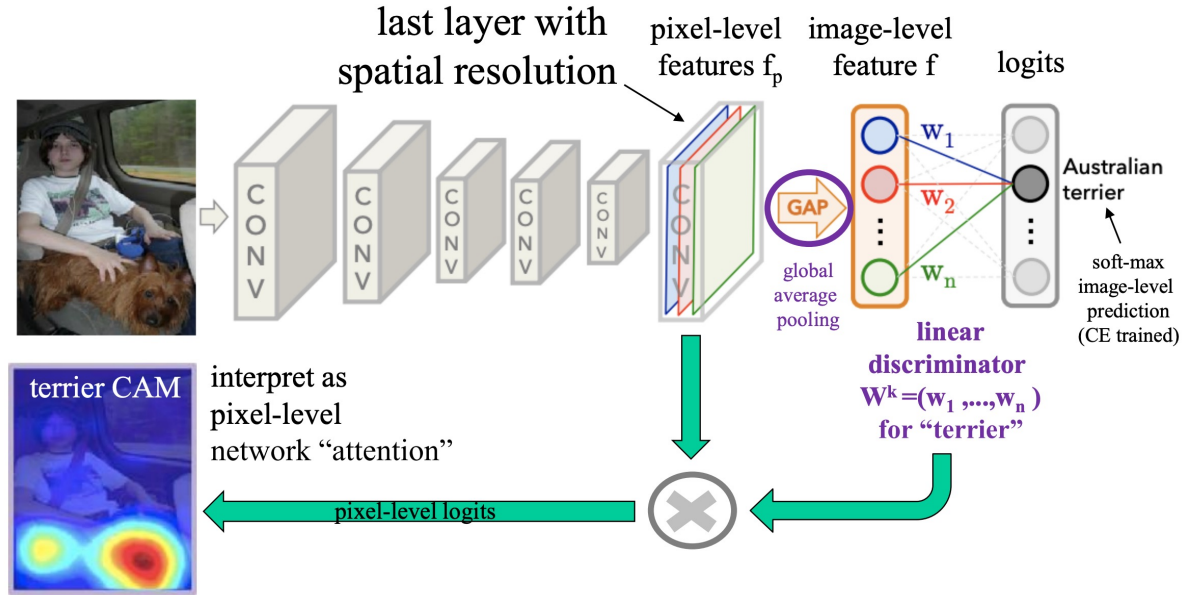


Figure 2.4: Illustration of class activation mapping (CAM). The class activation map shown in the left bottom corner is computed by multiplying the pixel-level features and the linear discriminator. Image from [127]

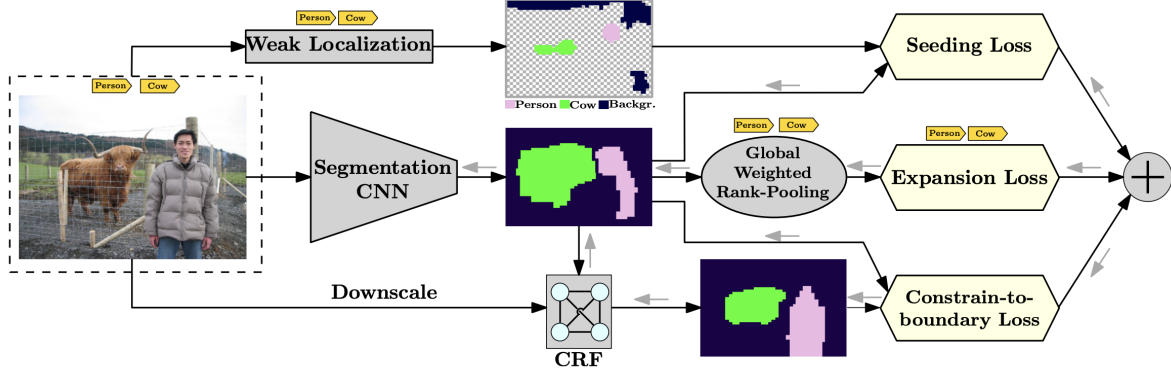


Figure 2.5: An illustration of the SEC training framework. The framework consists of three pipelines. The upper one generates pseudo labels with CAM, providing weak location cues. Pseudo labels are extracted from the high-confidence area of CAM with thresholds. The middle pipeline expands the predictions of tag classes and suppresses the predictions of non-tag classes. The object boundaries are refined in the bottom pipeline with CRF regularization. CRF will be elaborated on in Section 2.3.1. Image from [62]

is to extract pseudo labels, which are subsequently used for training segmentation models. The SEC training framework [62] is multi-stage. Weak location cues are extracted from a classification network using CAM, which is separated from the segmentation training pipeline shown in Figure 2.5.

2.2 Size constraint losses for segmentation

Size constraint losses penalize segmentation models when the predicted segment sizes deviate from the size targets. These constraints are especially useful in medical image segmentation, where the size of a tumor, organ, and anatomical structure can be an important feature for accurate diagnosis. Additionally, the sizes of anatomical structures do not vary much making the training benefit significantly from size supervision.

Section 2.2.1 discusses the representation of segment sizes. In Section 2.2.2, we present several unsupervised forms of size constraints that are typically used when no size supervision is available. Supervised size constraints are detailed in Section 2.2.3, where we primarily focus on two relevant papers [86, 57].

2.2.1 Size distribution

The segmentation prediction at an arbitrary pixel p , $S_p = (S_p^1, \dots, S_p^K)$, is a categorical distribution across the predefined K classes (including background). The average prediction of a given image is defined as

$$\bar{S} = \frac{1}{|\Omega|} \sum_{p \in \Omega} S_p \quad (2.14)$$

where Ω is the image domain. The average prediction $\bar{S} = (\bar{S}^1, \dots, \bar{S}^K)$ is also a categorical distribution over K classes. It can be seen as an image-level prediction of the relative or normalized sizes of the image objects. The *average prediction* $\bar{S}$ is often referred to as the size or volumetric distribution. It only differs from the actual segment size (in terms of pixel count) by a factor of the image size $|\Omega|$.

2.2.2 Unsupervised size constraints

Unsupervised size constraints refer to methods that impose restrictions on the size of the segments or clusters without relying on labeled data. These constraints are particularly useful when we do not know the ground truth sizes.

In the context of clustering with posterior models [21, 65], it is common to use *fairness* loss to encourage equal-size clusters. In this case, fairness is one of the discriminative properties enforced by the total loss in order to improve the model predictions on unlabeled training data. The fairness was motivated by information-theoretic arguments in [21] deriving it as a negative entropy of the data-set-level *average prediction* $\hat{S} := \frac{1}{|D|} \sum_{i \in D} S_i$ for dataset D

$$\begin{aligned} L_{fair}(\hat{S}) &= -H(\hat{S}) \equiv \sum_k \hat{S}^k \ln \hat{S}^k \\ &\stackrel{c}{=} \sum_k \hat{S}^k \ln \frac{\hat{S}^k}{1/K} \equiv KL(\hat{S} \| u) \end{aligned} \quad (2.15)$$

where $u = (\frac{1}{K}, \dots, \frac{1}{K})$ is a uniform categorical distribution, and symbol $\stackrel{c}{=}$ indicates that the equality is up to some additive constant independent of $\hat{S}$. Perona et al. [65] pointed out the equivalent KL-divergence formulation of the fairness in (2.15) and generalized it to a balanced partitioning constraint

$$L_{bal}(\hat{S}) = KL(\hat{S} \| v) \quad (2.16)$$

with respect to any given prior distribution v that could be different from uniform.

In the context of weakly-supervised semantic segmentation methods, segments for tag classes should have positive sizes, and segments for non-tag classes should have zero sizes. Size-based constraints are typically defined for the image-level *average prediction* $\bar{S}$ computed from pixel-level predictions S_p , e.g. see (2.14). Many generalized forms of pixel-prediction averaging can be found in the literature, where they are often referred to as *prediction pooling*. Some decay parameter often provides a wide spectrum of options from basic averaging to max-pooling. For example, Kolesnikov et al. [62] proposed to average the predictions with global weighted rank pooling where weights are higher for confident areas. Araslanov et al. [10] proposed normalized global weighted pooling using weights computed from the CAM heat map. While the specific form of pooling matters, for simplicity, we discuss the corresponding balancing loss functions assuming basic average prediction $\bar{S}$, as in (2.14). The SEC method on tag-supervised segmentation [62] uses *log-barriers* to “expand” tag objects in each training image and to “suppress” the non-tag objects. Assuming image tags T , their *suppression loss* is defined as

$$L_{\text{suppress}}(\bar{S}) \propto - \sum_{k \notin T} \ln(1 - \bar{S}^k) \quad (2.17)$$

encouraging each non-tag class to have zero average prediction $\bar{S}^k$, which implies zero predictions S_p^k at each pixel. Their *expansion loss*

$$L_{\text{expand}}(\bar{S}) \propto - \sum_{k \in T} \ln \bar{S}^k. \quad (2.18)$$

encourages positive average predictions $\bar{S}^k$ and non-zero volume for tag class segments. The expansion loss (2.18) may have a bias to equal-size segments, which is particularly evident in the case of average predictions. Indeed, as follows from (2.18)

$$L_{\text{expand}}(\bar{S}) \propto KL(u_T \| \bar{S}) \quad (2.19)$$

The intention of the log barrier loss (2.18) is to push image-level size prediction $\bar{S}$ from the boundaries of the probability simplex Δ^K corresponding to the zero-level for the tag classes T . Note that the mentioned decay parameter used for generalized average predictions should reduce such bias.

2.2.3 Supervised size constraints

Supervised size constraints incorporate explicit size information into the training to improve segmentation accuracy. As there is only a limited amount of related work in this domain, we focus on discussing two relevant papers.

Pathak et al. [86] proposed applying linear constraints on predictions to enhance semantic segmentation with tag supervision. The constraints can be formulated as

$$AS \leq b, \quad (2.20)$$

where S is the segmentation predictions generated by the neural network. Linear constraints are represented by $A \in \mathbb{R}^{m \times nk}$ and $b \in \mathbb{R}^m$, where k is the number of classes, n is the number of pixels, and m is the number of linear constraints. Such constraints are applied to each image in the dataset. These linear constraints can be categorized into two classes: non-tag constraints and tag constraints. It is natural to use suppression loss for non-tag classes. Unlike (4.13), they used a linear form of suppression loss

$$\sum_{p=1}^n \sum_{k \notin T} S_p^k \leq 0, \quad (2.21)$$

where T is the set of tag classes, p is the pixel index, and k represents the class. For tag classes, a lower and upper bound of constraints are applied to the predictions

$$a \leq \sum_{p=1}^n \sum_{k \in T} S_p^k \leq b, \quad (2.22)$$

where a and b are the lower and upper bounds of the size.

To efficiently optimize the constrained problem, they introduced a latent variable P and encouraged P and S to be the same with the KL divergence. Latent variable P can be considered as the synthesized full ground truth mask. The network is easy to train with cross-entropy loss (2.6) when P is given. They employed the block coordinate descent algorithm to optimize the network parameters. In the first step, network parameters are fixed, and variable P is solved with the constrained Lagrangian dual optimization. Specifically, the latent variable P is optimized to minimize the KL divergence under the linear constraints. Next, the latent variable P is fixed and network parameters are optimized with gradient descent. However, the iterative Lagrangian dual optimization is expensive. To mitigate that, Kervadec et al. [57] proposed a quadratic size-constraint loss.

The size constraint loss proposed by [57] is

$$\mathcal{C}(V_S) = \begin{cases} (V_S - a)^2 & V_S \leq a \\ (V_S - b)^2 & V_S \geq b \\ 0 & \text{Otherwise,} \end{cases} \quad (2.23)$$

where V_S is the segment volume (equivalent to $\bar{S}$), a and b are the lower and upper bounds, see Figure 2.6 for an illustration of this loss. The loss function (2.23) achieved better performance than [86] on medical images.

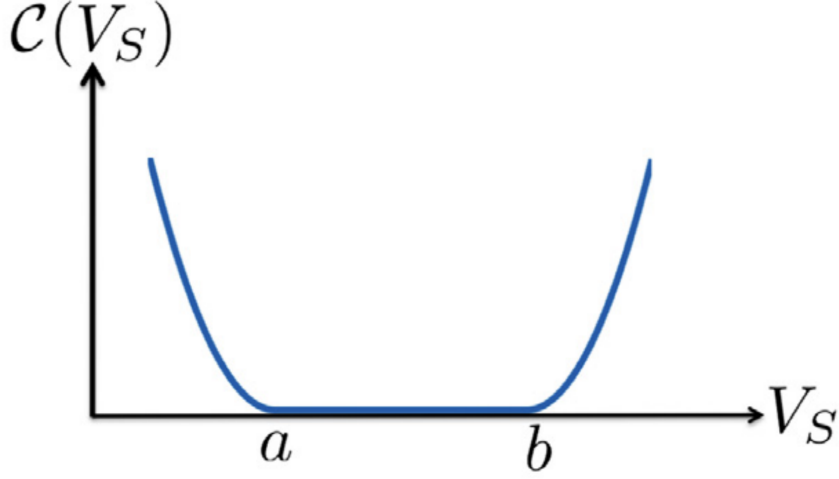


Figure 2.6: Illustration of quadratic size constraint loss. Segment volume is denoted by V_S . Parameters a and b represent the lower and upper bounds of the size. Loss function is indicated by the blue line. Image from [57]

2.3 Unsupervised losses for segmentation

In this section, we mainly discuss two types of unsupervised losses: CRF and clustering loss, which are closely related to the methods we proposed in Chapter 3 and 4.

2.3.1 CRF regularization

Conditional random fields (CRF) are common basic examples of pairwise graphical models. The pairwise potentials reflect the cost of label assignments between pairs of pixels. Mathematically, the energy of a label configuration S given an input image I can be expressed as:

$$\sum_{p,q} \psi_p(S_p, S_q | I), \quad (2.24)$$

where $S = \{S_1, \dots, S_{|\Omega|}\}$ represents the label assignment for all pixels in image domain Ω . Each pixel belongs to one class $S_i \in \{1, \dots, K\}$, where K is the number of categories.

Potts model [87] is a famous form of the pairwise term that can be expressed as follows:

$$\sum_{p,q \in N} w_{pq} [S_p \neq S_q], \quad (2.25)$$

where w_{pq} is non-negative affinities between pairs of pixels $p, q \in \Omega$, and $[\cdot]$ is the Iverson bracket, and N is the neighborhood system. In low-level binary segmentation with seeds, the pairwise energy (2.25) can be optimized via graph cut [17], see Figure 2.7. In practice, affinity weights w_{pq} are set close to 1 if two neighboring pixels p, q have similar intensities, and weight w_{pq} is set close to zero either when two pixels are far from each other on the pixel grid or if they have largely different intensities [17, 63, 105]. The affinity matrix W could be arbitrarily dense or sparse. In a sparse neighborhood system, each pixel is only connected with its four intermediate neighbors, see the left two images in Figure 2.8. Conversely, dense CRF connects each pixel with all other pixels in the image as shown in the right part of Figure 2.8. The non-zero discontinuity costs between neighboring pixels are often set by a Gaussian kernel $w_{pq} = \exp \frac{-\|I_p - I_q\|^2}{2\sigma^2}$ of some given bandwidth σ , which works as a soft threshold for detecting high-contrast intensity edges in the image. Thus, CRF loss encourages both the alignment of the segmentation boundary to contrast edges in the (high-resolution) input image and the shortness/regularity of this boundary.

The CRF loss functions can be formulated for continuous SoftMax predictions S_p produced by segmentation networks, e.g. [62, 77, 105]. It is natural to use relaxation forms of

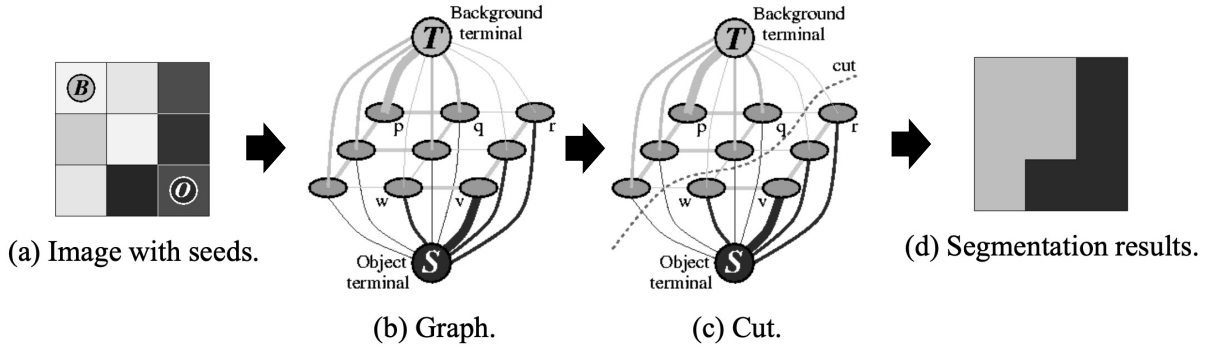


Figure 2.7: Graphical model for low-level binary segmentation. Figure (a) shows the input image. The brightness of pixels represents their intensities. Two letters in the circles represent two seeds corresponding to the background (B) and object (O). Figure (b) shows the graph. Each pixel is a graph node. “T” and “S” are two extra nodes indicating the class. Nodes are connected with the edges and the edge thickness denotes the weights showing affinity between graph nodes. The weights between seed pixels and their corresponding class node are infinity. Figure (c) optimized the binary segmentation problem via graph cut. Figure (d) demonstrates the final segmentation results. Image from [19]

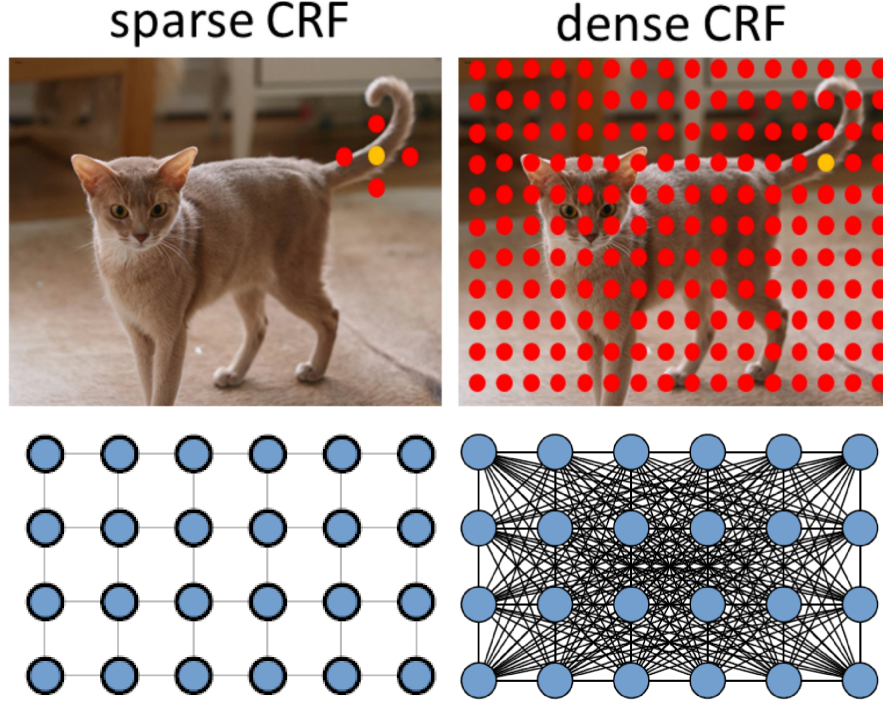


Figure 2.8: Illustration of sparse (left) and dense (right) CRF. The upper row uses one pixel as an example to show the difference between sparse and dense CRF. The neighborhood system of the yellow pixel is indicated by the red dots. The lower row shows the overview of sparse and dense connectivity in the CRF model, where the blue circles are pixels and the edges connect the pixels in the neighborhood. Image from [109, 63]

the standard discrete CRF models. For example, the quadratic formulation is

$$\sum_{p,q \in N} w_{pq} \|S_p - S_q\|^2, \quad (2.26)$$

Tang et al. [105] employed bilinear relaxation of the dense Potts model as a regularization loss to train scribble-supervised CNN segmentation models. Their loss function is formulated as

$$\mathcal{L}_{CRF} = \sum_k (\mathbf{1} - S^k)^\top W S^k \quad (2.27)$$

where $S^k := (S_p^k | p \in \Omega)$ is a vector of all pixel predictions specifically for class k . It is easy to interpret loss (2.27) assuming, for simplicity, that all pixels have confident *one-*

hot predictions S_p so that each S^k is a binary indicator vector for segment k . Then the loss sums all weights w_{pq} between the pixels in different segments. Thus, the weights are interpreted as discontinuity penalties. The loss minimizes the discontinuity costs [17].

2.3.2 Clustering techniques for segmentation

K-means

Given an image I containing unlabeled low-level features $\{f_1, \dots, f_{|\Omega|}\}$, the unsupervised segmentation problem can be solved by clustering methods. Let us first consider the basic *K-means* algorithm:

$$E(S, \mu) = \sum_{k=1}^K \sum_{p \in S^k} \|f_p - \mu^k\|^2, \quad (2.28)$$

where S is the segmentation variable defining K subsets of Ω , Ω is the image domain, and K is the predefined number of clusters. The output of the K-means algorithm is $S = \{S^1, \dots, S^K\}$. Segment/cluster centers (means) are denoted as $\mu = \{\mu^1, \dots, \mu^K\}$. K-means can be non-parametric clustering energy by substituting μ into (2.28). Then, K-means can be equivalently formulated as follows up to an additive constant:

$$E(S) = - \sum_{k=1}^K \sum_{p, q \in S^k} \frac{\langle f_p, f_q \rangle}{|S^k|}. \quad (2.29)$$

Additionally, the inner summation of (2.28) is closely related to sample variance. Hence, K-means can be shown as the *variance clustering* criteria:

$$E(S) = \sum_{k=1}^K |S^k| \text{var}(S^k), \quad (2.30)$$

where *var* is the sample variance of segment S^k . K-means objective (2.28) is optimized via block-coordinate descent iterating two steps: 1) fix segment S and compute cluster means $\mu^k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$; 2) fix μ^k and re-assign each point to the nearest mean.

Superpixel [2] applies K-means to low-level features combining pixel intensities (RGB) and coordinates (XY). Chan and Vese [23] combined K-means with the Potts model. The K-means is applied in image intensity space and boundary regularization is employed on XY coordinates:

$$E(S, \mu^0, \mu^1) = \sum_{k=0}^1 \sum_{p \in S^k} \|\mu^k - f_p\|^2 + \sum_{p, q \in N} W_{pq} [S_p \neq S_q] \quad (2.31)$$

Kernel K-means

Features f in low-dimensional space can be hard to partition. *Kernel K-means* [98] maps low-level features to a high-dimensional Hilbert space $\phi_p \equiv \phi(f_p)$ with mapping $\phi : R^N \rightarrow H$. The data can become linearly separable after the mapping. Then, the segmentation problem is solved in high-dimensional space by applying kernel trick to (2.29):

$$F(S) = - \sum_{k=1}^K \sum_{p,q \in S^k} \frac{k(f_p, f_q)}{|S^k|}, \quad (2.32)$$

where $k(f_p, f_q) := \langle \phi(f_p), \phi(f_q) \rangle$ is the pair-wise similarity of features. Now the high embedding ϕ is implicitly defined by the kernel k . The kernel operator can also be replaced by the affinity matrix $A = [A_{pq}]$, where $A_{pq} := k(f_p, f_q)$. The inner summation $\sum_{pq} A_{pq}$ is equivalent to matrix notation $S^{k'} A S^k$, where S^k the binary indicator vector and $S^{k'}$ corresponds to the transpose of the vector. Substituting these notations into (2.32), we obtain the *average association* [39] objective function as follows:

$$- \sum_{k=1}^K \frac{S^{k'} A S^k}{|S^k|} \quad (2.33)$$

Shi et al. [99] proposed a normalized version of (2.33) for spectral clustering, which can be seen as a weighted form of kernel K-means. The normalized cut is also closely related to the bilinear relaxation of CRF loss, see (2.27)

$$\sum_{k=1}^K \frac{S^{k'} A S^k}{\mathbf{1}' A S^k} \quad \text{Normalized association} \quad (2.34)$$

$$\sum_{k=1}^K \frac{S^{k'} A (\mathbf{1} - S^k)}{\mathbf{1}' A S^k} \quad \text{Normalized cut} \quad (2.35)$$

Entropy clustering

K-means objective (2.28) can be equivalently written as the sum of negative log-likelihood of Gaussian distribution up to an additive constant, which is known as the probabilistic K-means [56]:

$$E(S, \mu) = \sum_{k=1}^K \sum_{p \in S^k} -\log P(f_p | \mu^k), \quad (2.36)$$

where P is a Gaussian distribution with mean μ^k . The Gaussian distribution may fail to model complex features in natural images. It is common to use more sophisticated models (e.g. Gaussian mixture model, histogram) to fit the appearance:

$$E(S, \theta) = \sum_{k=1}^K \sum_{p \in S^k} -\log P(f_p | \theta^k), \quad (2.37)$$

where S_p is the segment indicator of pixel p , θ is the parameters of the appearance model. The model estimation can be seen as parameter estimation via maximum likelihood from the statistical point of view. On the other hand, the log-likelihood can be written as the cross-entropy between the true color distribution and the color model:

$$-\sum_{k=1}^K |S^k| \sum_i P_i^{S^k} \ln P_i^{\theta^k} \xrightarrow{\min \theta} \sum_{k=1}^K |S^k| H(S^k), \quad (2.38)$$

where i represents the color, P^{S^k} is the true color distribution in segment S^k , P^{θ^k} is the color model of segment S^k . Minimization of such energy results in the entropy of true color distribution $H(S^k)$. Hence, the log-likelihood segmentation energy can be interpreted as the energy clustering from the information-theoretic perspective. The entropy clustering criteria (2.38) can be juxtaposed with (2.30).

It is also common to combine appearance model fitting with boundary regularization in low-level image segmentation [130, 107, 93, 37]. Generally, the low-level segmentation tasks can be divided into two groups. The first group of tasks assumes the appearance model is fixed and given. The other group simultaneously estimated the appearance model and segmentation variables. Rother et al. [93] iteratively estimates the appearance model with the following loss:

$$E(S, \theta^0, \theta^1) = \sum_{p \in \Omega} -\log P(f_p | \theta^{S_p}) + \sum_{p, q \in N} W_{pq} [S_p \neq S_q]. \quad (2.39)$$

The iteration starts from the appearance models θ^0, θ^1 with colors inside and outside some given bounding box. It iterates the graph cut segmentation and appearance model re-estimation. Tang et al. [104] proposed a method representing the appearance model by

color histograms and optimizing (2.39) with one graph-cut. Zhu et al. [130] proposed a general unsupervised segmentation method:

$$E(S, \theta^0, \theta^1, \dots) = \sum_{p \in \Omega} -\log P(f_p | \theta^{S_p}) + \sum_{p, q \in N} W_{pq} [S_p \neq S_q] + |labels|, \quad (2.40)$$

where $|labels|$ is the cardinality of the label set. The optimization of the (2.40) also alternates segmentation with model re-estimation.

K-means (2.28) is widely used in deep learning to cluster deep features for segmentation [126, 58, 128]. However, the entropy clustering technique remains underexplored in the context of semantic segmentation. Addressing this gap, Chapter 4 introduces a *mutual information* (MI) loss, which can also be seen as a form of entropy clustering.

Chapter 3

Size Targets for Segmentation

We propose a new general form of image-level supervision for semantic segmentation based on approximate targets for the relative size of segments. At each training image, such targets are represented by a volumetric distribution. Soft-max prediction $S_p = (S_p^1, \dots, S_p^K)$ at any pixel p is a categorical distribution over K classes, including background. The average prediction of any image defined in (2.14), is also a categorical distribution over K classes. We assume that training images have approximate size targets represented by categorical distributions $v = (v_k)_{k=1}^K$, e.g. $v = (0, .15, 0, \dots, 0, .75)$ for the middle image in Fig.3.1 if “bird” is the second class and “background” is the last. For each training image, our *size-target loss*

$$L_{size} = KL(v \parallel \bar{S}) = \sum_k v_k \ln \frac{v_k}{\bar{S}^k} \quad (3.1)$$

is based on Kullback–Leibler (KL) divergence. Our total loss is very simple: it combines size-target loss (3.1) and a common CRF loss (2.27). It performs well on standard segmentation architectures. Figure 3.2(b) shows some results for a generic segmentation network (R101 backbone) trained on PASCAL VOC [42] using only image-level supervision with approximate size targets (8% mean relative errors).

In general, it would be great to have effective image-level supervision for segmentation that only uses barriers like (2.18) or (2.23) since they do not require any specific size targets. This corresponds to tag-only supervision. However, our empirical results for semantic segmentation using such barriers were poor and comparable with those in [86], see Figure 3.2(a). A number of more recent semantic segmentation methods for tag-level supervision have considerably improved such results [5, 128, 30, 129, 119, 95, 121, 120], but they introduce significantly more complex multi-stage training procedures and various

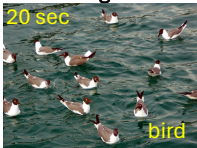
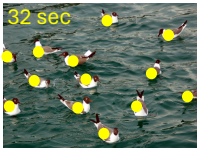
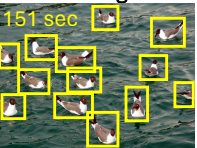
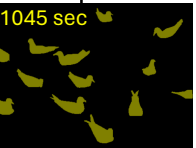
	Tag	Point	Size target (ours)	Bounding box	Full supervision
					
sec. per image:	20.0 sec	23.3 sec	36.2 sec	47.1 sec	239.7 sec
mIOU:	62.7% (WR38)	66.6% (R101)	72.6% (R101)	74.9% (R101)	78.2% (R101)

Figure 3.1: Labeling speed and accuracy for different forms of supervision on PASCAL VOC. The table shows ballpark estimates of average labeling time per image in the whole dataset. We use the data in [12], as well as our Table 1, and aggregate all labeling speeds from “per class”, “per instance”, or “per point” to “per image” using the average number of instances or classes in each image and the aggregation rules formulated in [12], see their Section 4. The top-left corner in each picture shows the corresponding estimated labeling times for the representative multi-instance image. All the labeling times above are only rough estimates, but they are intuitive. The relative costs for point supervision seem underestimated, but they follow [12]. The values of “point”, “size target” and “full supervision” accuracy (mIOU%) are based on the experiments in our main paper (Figs.4). For fairness, we compare these with end-to-end methods using similar ResNet backbones in *tag*- and *box*-supervision, [10, 67].

architectural modifications, which makes such methods hard to replicate, generalize, or to understand the results.

We are focused on general easy-to-understand end-to-end training methods based on volumetric or size targets, which we show can be approximate. Our main point is that this image-level supervision requires little extra information on top of image tags, compared to full pixel-accurate ground truth, but it produces a training quality comparable with full supervision using standard semantic segmentation architectures. Approximate size targets are relatively easy to get from human annotators. Our work also motivates the development of size predictor models, which will be discussed in Chapter 5.

3.1 Size-target loss and its properties

In contrast to our loss for image-level supervised segmentation (3.1), fairness (2.15) and balancing (2.16) losses in discriminative clustering [21, 65, 49] use the *reverse* KL divergence where the target distributions u or v appear as the second argument. As discussed later

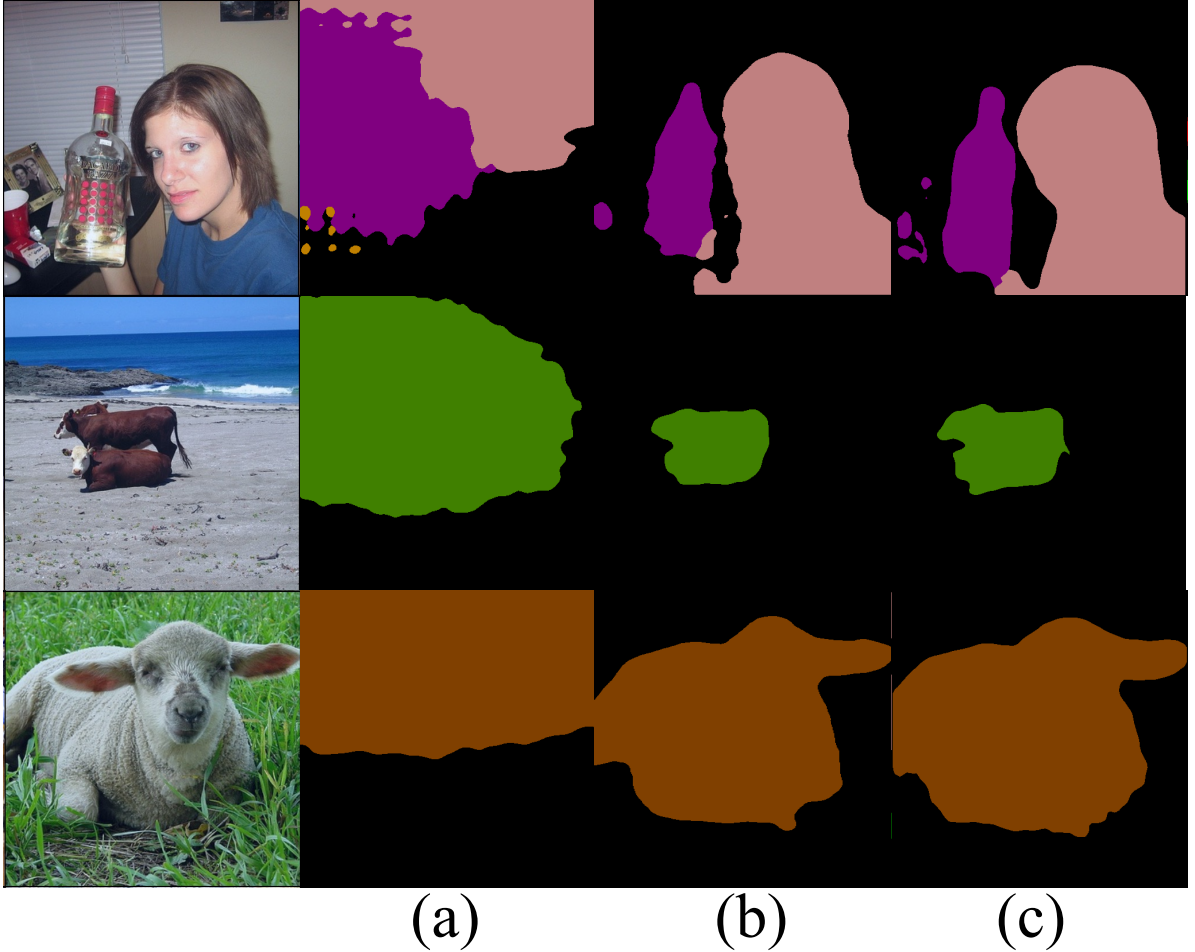


Figure 3.2: Semantic segmentation with DeepLabV3+(R101) models: PASCAL validation results for training with (a) log-barrier (2.18) using class tags, (b) KL-divergence (3.1) using our approximate size targets, (c) cross-entropy with full (mask) supervision.

in the paper, this difference is important and loss (2.16) is numerically unstable for image segmentation.

Our proposed total loss is very simple

$$L_{total} := L_{size} + L_{crf} \quad (3.2)$$

where the two terms come from our size target loss (3.1) using the *forward* KL-divergence and standard CRF loss as in (2.27). The core new component is our size loss. Several

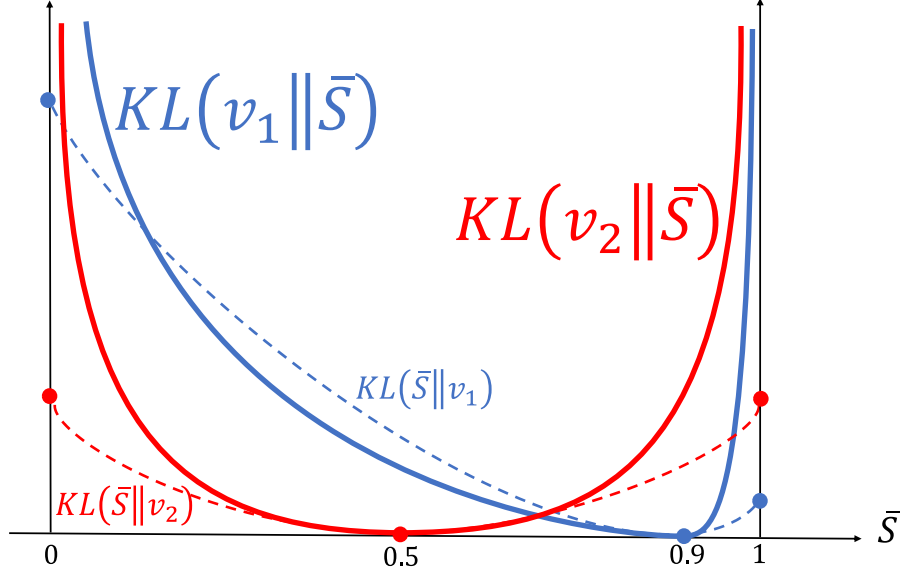


Figure 3.3: *Forward vs. reverse* KL divergence. Assuming binary classification $K = 2$, we can represent all possible probability distributions as points on the interval $[0,1]$. The solid curves illustrate our “strong” size constraint, i.e. the *forward* KL-divergence $KL(v\|\bar{S})$ for the average prediction $\bar{S}$. We show two examples of volumetric prior $v_1 = (0.9, 0.1)$ (blue curve) and $v_2 = (0.5, 0.5)$ (red curve). For comparison, the dashed curves represent reverse KL divergence $KL(\bar{S}\|v)$ commonly used in the prior art.

important properties of L_{size} are discussed below.

First of all, our size target loss (3.1) encourages specific approximate target volumes for image tag classes, but also encourages zero volumes for not-tag classes. The CRF loss also contributes to the suppression of redundant classes. Unlike most prior work on image-level supervision for semantic segmentation, e.g. [62, 10, 128], we do not need separate suppression loss terms like (2.17). We validated this claim experimentally, they do not change the results.

The size loss can also be integrated into other weakly-supervised settings, e.g. based on partial pixel supervision (seeds). Our size target loss (3.1) just needs to be combined with other losses used in these methods, e.g. (2.12). In particular, approximate size targets significantly improve the results in [105] in cases when seed scribbles are minimal.

$$L'_{total} := L_{size} + L_{crf} + L_{pce} \quad (3.3)$$

As well-known, KL divergence is not symmetric with respect to its arguments. In the context of our work on image-level supervision for semantic segmentation, the order of the estimate and target distributions matter significantly. When the estimated distribution, e.g. $\bar{S}$, is the first argument, this loss is the reverse KL divergence as used in many prior arts on image classification. For classification problems, there is no numerical issue with putting the target distribution in the second position, because the target distribution usually does not have zero support for any class to enforce balance among classes, see (2.16). However, the image tags often only contain a small subset of the whole categories. This makes the reverse KL divergence ill-defined for the image-level target distribution since there will be many zeros at the denominator inside the logarithm. This motivates us to use the forward KL divergence which is more numerically stable.

Moreover, the forward KL divergence has the so-called zero-avoiding property which is illustrated in Figure 3.3. The penalty will be infinite if any class corresponding to the non-zero target distribution gets zero prediction while the reverse KL divergence only enforces a finite penalty. Such property guarantees that we do not generate trivial solutions or miss any class labels from the image tag sets.

3.2 Human-annotated size targets

To assess the effectiveness of our method, we have annotated a subset of training images from the PASCAL VOC dataset. Size-target annotations are available for three classes: cat, dog, and bird. Our future work will involve annotating the entire dataset and conducting a comprehensive evaluation of the size-target method using more human-annotated data. A user interface with an assistance tool was developed to facilitate the annotation. The assistance tool overlays grid lines partitioning the image into 5×4 small rectangles or 3×3 large rectangles, see Figure 3.4. The user can enter the size for a given class in each image either by counting rectangles (fractions are OK) or by directly entering the corresponding percentage of the image size. Annotators can choose finer or coarser partitioning for each image depending on the object size. Empirical evidence shows that annotators are approximately two times more accurate with the assistance tool, especially for small objects in the image. We take *relative error* (RE) as the evaluation criterion to measure the accuracy of human annotations. The relative error has a straightforward interpretation, representing the error as a percentage of the true target. For each training image containing class k , the relative error for the size target v^k is defined as

$$RE(v^k) = \frac{|v^k - \hat{v}^k|}{\hat{v}^k}, \quad (3.4)$$

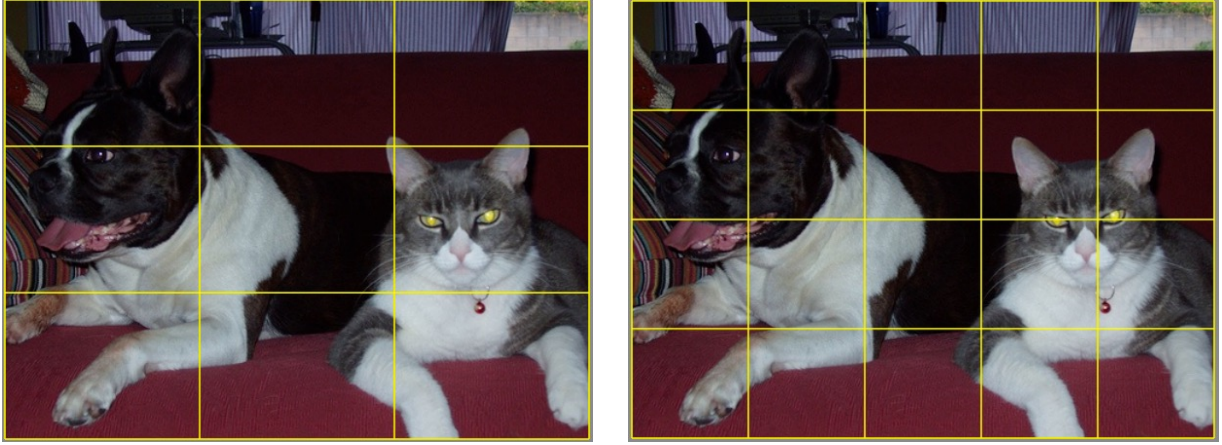


Figure 3.4: An illustration of the assistance tool for size-target supervision annotation. Image from Pascal dataset [42]

	speed	mRE
cat	12.6s	12.3%
dog	9.1s	16.6%
bird	15.2s	20.1%

Table 3.1: Human-annotated size targets. The table shows the average speed and relative errors (mRE) we obtained for human annotation of size targets for individual classes (cat, dog, bird) on PASCAL.

where $\hat{v}_k$ is the true size obtained from its corresponding ground truth segmentation mask. Let D denote the dataset we annotate. The *mean relative error* (mRE) is the average of RE over all images and all classes:

$$mRE = \frac{1}{|D|} \sum_{i \in D} \frac{1}{|T_i|} \sum_{k \in T_i} \frac{|v_i^k - \hat{v}_i^k|}{\hat{v}_i^k}, \quad (3.5)$$

where T_i represents the set of present classes (tag) in the i th image. Table 3.1 reports the annotation speed per image and the mean relative error (3.5) of each class. Figure 3.5 shows the histograms of relative errors for human annotations. The histograms illustrate that annotated size errors are mostly below 10%, but occasional large mistakes (*heavy tails*) raise the mean error. The total mRE is 15.9% for the three annotated classes.

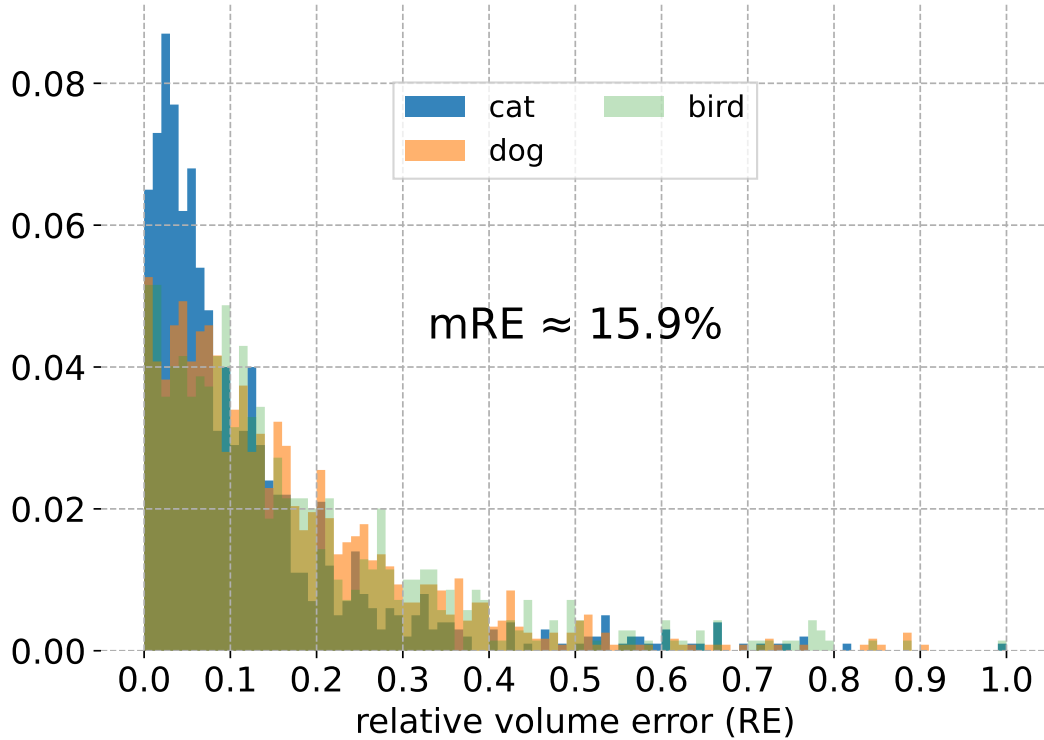


Figure 3.5: Quality of human annotations measured by relative errors for the dog, cat, and bird classes on the PASCAL dataset. The histograms are normalized by the number of images in each class. The mean relative error for the three classes is 15.9%.

3.3 Synthetic size targets with additive noise models

As we only annotate three classes (cat, dog, and bird) in the PASCAL VOC dataset, to comprehensively evaluate our approach on the whole dataset, we generate noisy synthetic size targets by corrupting the true size targets that are extracted from the segmentation ground truth masks. Notations $v = (v^k)_{k=1}^K$ and $\hat{v} = (\hat{v}^k)_{k=1}^K$ respectively represent the synthetic and true size targets of a training image in the dataset with K classes. We add Gaussian noise ϵ to the true size for corruption. Noise ϵ is only added to the size of tag classes (T) in the image, yielding the synthetic size target v^k of each tag class ($k \in T$):

$$v^k = \hat{v}^k + \epsilon^k \sim \mathcal{N}(0, \sigma(\hat{v}^k)) \quad (3.6)$$

We assume the standard deviation σ of additive error (AE) ϵ is a function of $\hat{v}$. To make the additive error (AE) closely match the error of human annotations, we study a variety of AE models from the basic linear one described in Section 3.3.1 to more advanced AE models detailed in Section 3.3.2.

In practice, the synthetic size targets v may not be a distribution that sums to 1 after corruption with ϵ . When training the neural networks, we re-normalize v . This re-normalization step is ignored when analyzing AE models.

3.3.1 Basic linear AE model

For simplicity, we omit the class index k in the notations, using v or $\hat{v}$ to denote one (predicted or actual) size target in the subsequent sections that discuss AE models. We start with a naive method assuming the standard deviation $\sigma(\hat{v})$ of the additive error ϵ grows linearly with $\hat{v}$. The standard deviation $\sigma(\hat{v})$ can be formulated as follows

$$\sigma(\hat{v}) = \gamma \cdot \hat{v}, \quad (3.7)$$

where γ is a parameter that controls the level of corruption. It is easy to find that the *conditional mean relative error* of v given $\hat{v}$ is a constant:

$$mRE(\hat{v}) = E\left(\frac{|v - \hat{v}|}{\hat{v}}\right) = \frac{MAD(\epsilon|\hat{v})}{\hat{v}} = \gamma\sqrt{\frac{2}{\pi}}, \quad (3.8)$$

where $MAD(\epsilon|\hat{v})$ is the *conditional mean absolute deviation* of ϵ given $\hat{v}$. *Mean absolute deviation* (MAD) is a standard statistical measure that is closely related to the variance, but MAD calculates the absolute difference rather than the squared difference:

$$MAD = \frac{1}{n} \sum_i |x_i - E(X)| \quad (3.9)$$

For Normal distribution, the ratio between MAD and the standard deviation is $\sqrt{\frac{2}{\pi}}$. Numerically, $MAD(\epsilon|\hat{v})$ is the same as the *conditional mean absolute error* ($mAE(\hat{v}) = E(|v - \hat{v}|)$) of v given $\hat{v}$. The minor discrepancy between the two terms is that $MAD(\epsilon|\hat{v})$ is a statistic of additive error ϵ , while $mAE(\hat{v})$ is a metric describing the synthetic size target v .

Parameter γ is calculated to match synthetic size target mRE with human mRE (3.5). For linear AE model, the synthetic size target mRE equals $mRE(\hat{v})$. As a result, it is

easy to get parameter $\gamma = \sqrt{\frac{2}{\pi}} \cdot mRE_{human} \approx \frac{mRE_{human}}{0.8}$, where mRE_{human} represents the mRE of human annotations. The motivation for matching mean relative error is due to its straightforward interpretation, describing the error as the percentage of the true target.

Figure 3.6 compares the conditional $MAD(\epsilon|\hat{v})$ plots of human-annotated size and synthetic size generated by the AE model. The parameter γ of the AE model is set to match the mRE_{human} . The human curve shows that the standard deviation $\sigma(\epsilon)$ does not grow linearly with $\hat{v}$, which inspired us to study advanced AE models.

3.3.2 Advanced AE models

We only conduct a theoretical analysis of the following advanced AE models. The empirical evaluation of the advanced AE models is on a subset of the Pascal dataset, see Figure 3.14. It is our future work to perform a comprehensive study of the advanced AE model on the whole Pascal dataset.

Triangular Model. Figure 3.6 shows that human error does not increase with $\hat{v}$ after $\hat{v}$ exceeds 50%. This behavior is reasonable, as annotators tend to estimate the size of other areas $(1 - \hat{v})$ when the class of interest occupies more than half of the image. This strategy is adopted because predicting the sizes of smaller areas is typically easier. This can also explain why human error drops when $\hat{v}$ is greater than 50%. Inspired by this observation, we propose the triangular model. The word “triangular” stems from the triangular shape of $\sigma(\hat{v})$, see the left plot of Figure 3.8. We define the standard deviation $\sigma(\hat{v})$ of triangular model as

$$\sigma(\hat{v}) = \gamma \cdot \min\{\hat{v}, 1 - \hat{v}\}. \quad (3.10)$$

Bernoulli Model. Bernoulli model may more accurately reflect the patterns of human errors. We define S_p as the segment indicator at pixel p , which is treated as a random variable uniformly distributed over the image grid Ω . Then, the random variable S_p corresponds to binary Bernoulli trials with distribution

$$E(S_p) \approx \frac{1}{|\Omega_{samp}|} \sum_{p \in \Omega_{samp}} S_p = v_S, \quad (3.11)$$

where v_S represents the predicted volume of segment S . We conjecture that humans predict volume v_S for any class based on sampling S_p at pixels inside and outside of segment S . The sum of Bernoulli trials S_p follows a binomial distribution, which (by the law of large numbers) can be approximated by a Normal distribution. The uncertainty (variance) of

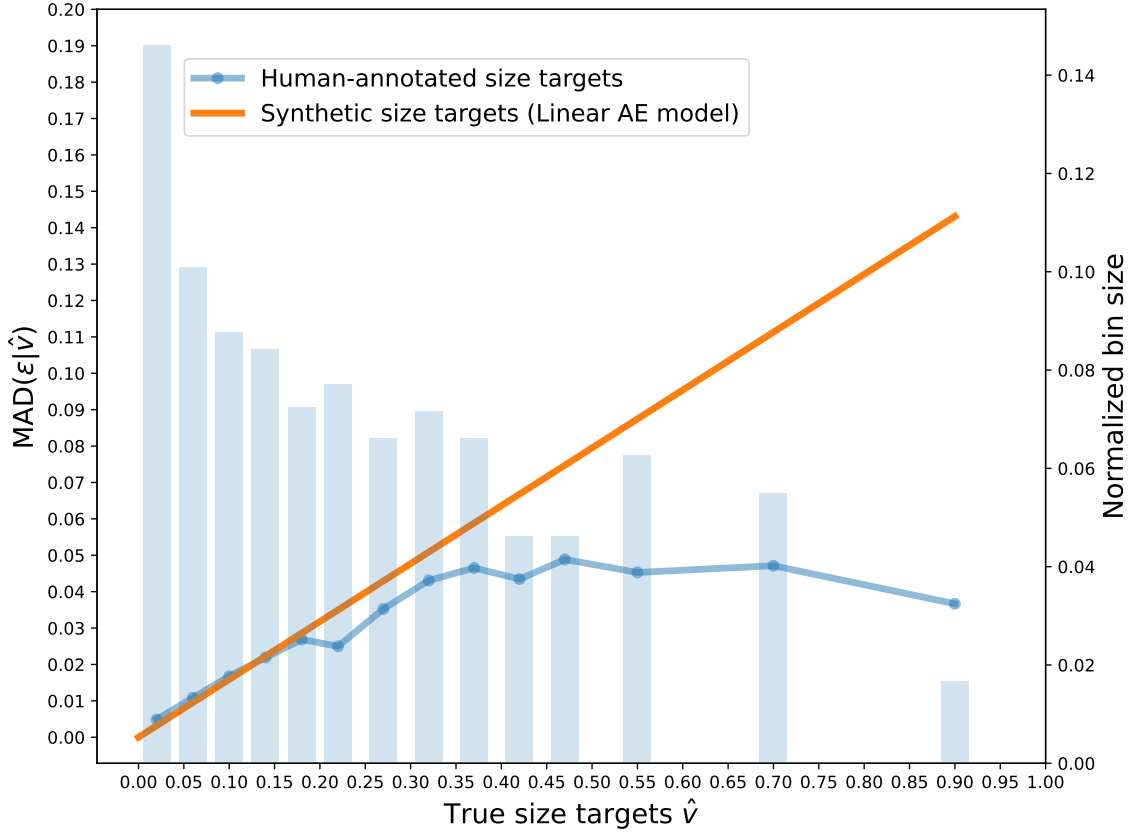


Figure 3.6: Comparison of the conditional mean absolute deviation ($MAD(\epsilon|\hat{v})$) between the linear AE model and human annotations. We match the mRE of the linear AE model with human annotations (15.9%) on the annotated PASCAL dataset by tuning the γ . The blue curve represents the MAD of human annotations including all three classes (cat, dog, and bird). The MAD of human-annotated size is computed within bins. The normalized bin sizes are shown as the bar plot.

v_S should match the uncertainty (variance) of the binomial distribution, justifying the

following AE model:

$$\sigma \left(\frac{1}{|\Omega_{samp}|} \sum_{p \in \Omega_{samp}} S_p \right) \approx \sqrt{\frac{\hat{v}(1 - \hat{v})}{|\Omega_{samp}|}}. \quad (3.12)$$

The standard deviation of Bernoulli trials can be matched with $\sigma(\hat{v})$ of Gaussian additive errors for some γ as follows

$$\sigma(\hat{v}) = \gamma \sqrt{\hat{v}(1 - \hat{v})}. \quad (3.13)$$

Entropic Model. Furthermore, we conjecture that the standard deviation of human-annotated sizes is proportional to the uncertainty (entropy) of Bernoulli trials $(\hat{v}, 1 - \hat{v})$. This is motivated by “Bernoulli model” where $\sigma(\hat{v})$ is near zero when $\hat{v}$ is zero or one and $\sigma(\hat{v})$ is the highest at $\hat{v} = 0.5$. Therefore, we propose the quadratic AE model, where the standard deviation of ϵ is proportional to the quadratic entropy. The standard deviation $\sigma(\hat{v})$ is defined as follows

$$\sigma(\hat{v}) = \gamma \cdot QH(\hat{v}, 1 - \hat{v}) \equiv \gamma \cdot \hat{v}(1 - \hat{v}), \quad (3.14)$$

In general, we can use *Rényi entropy* to model the uncertainty of ϵ . Rényi entropy is a concept in information theory that generalizes Shannon entropy. Rényi entropy provides a family of measures that offer different perspectives on uncertainty. For a discrete random variable with a probability mass function $P(X)$, the Rényi entropy of order α is defined as:

$$H_\alpha(X) = \frac{1}{1 - \alpha} \log \sum_x P(x)^\alpha \quad (3.15)$$

Rényi model incorporates all previous AE models as special cases (approximately). The standard deviation $\sigma(\hat{v})$ of ϵ is defined as

$$\sigma_\alpha(\hat{v}) = \gamma \cdot \frac{1}{1 - \alpha} \ln(\hat{v}^\alpha + (1 - \hat{v})^\alpha), \quad (3.16)$$

Gamma estimation. We estimate the parameter γ in the AE model to match with human *mRE*. The mean relative error of a general AE model is

$$\begin{aligned} mRE_{AE} &= \int_0^1 mRE(\hat{v}) \cdot p(\hat{v}) d\hat{v} \\ &= \sqrt{\frac{2}{\pi}} \int_0^1 \frac{\sigma(\hat{v})}{\hat{v}} \cdot p(\hat{v}) d\hat{v}, \end{aligned} \quad (3.17)$$

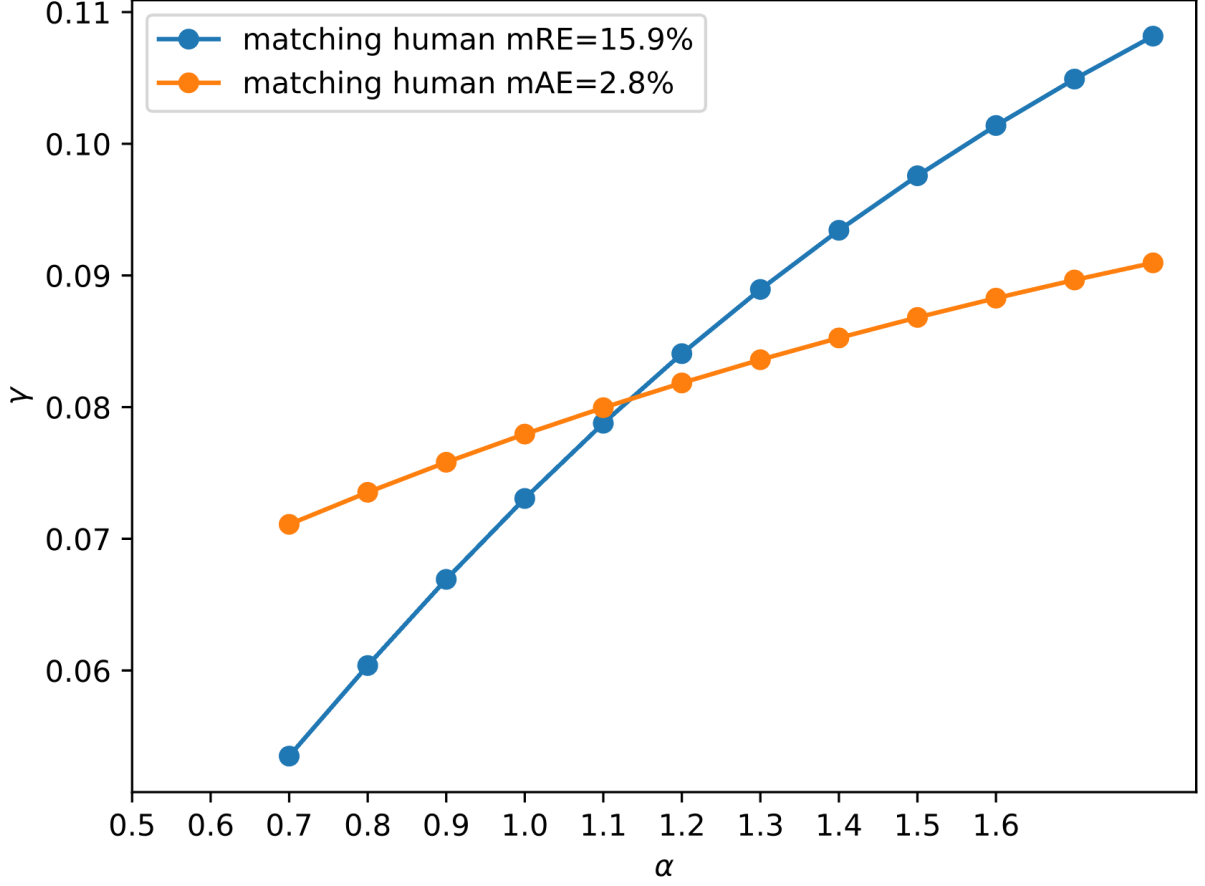


Figure 3.7: Alpha value selection for the Renyi model. The blue and orange lines show the Gamma values that respectively match with human mRE 15.9% and human mAE 2.8% across different α . We select the $\alpha = 1.1$ that can match best with both human mRE and mAE.

where $p(\hat{v})$ is the probability density of the size $\hat{v}$. For the α -Rényi model, we define

$$\begin{aligned}
mRE_{\alpha} &:= \frac{1}{1-\alpha} \sqrt{\frac{2}{\pi}} \int_0^1 \frac{\ln(\hat{v}^{\alpha} + (1-\hat{v})^{\alpha})}{\hat{v}} p_{\hat{v}}(\hat{v}) d\hat{v} \\
&\approx \frac{1}{1-\alpha} \sqrt{\frac{2}{\pi}} \sum_{\hat{v} \in D} \frac{\ln(\hat{v}^{\alpha} + (1-\hat{v})^{\alpha})}{|D| \cdot \hat{v}}.
\end{aligned} \tag{3.18}$$

The second step in (3.18) uses Monte-Carlo estimation to approximate the integral over $\hat{v}$

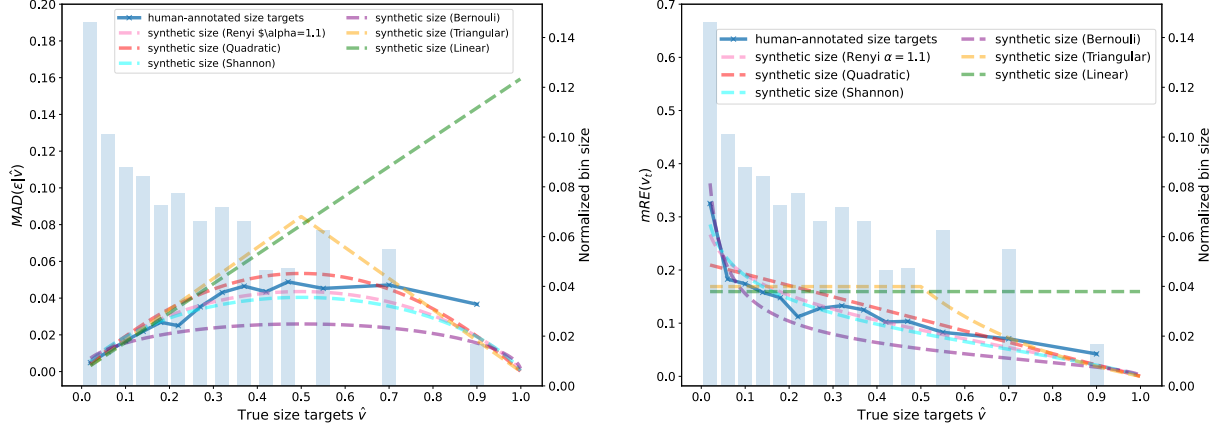


Figure 3.8: The $MAD(\epsilon|\hat{v})$ (left figure) and $mRE(\hat{v})$ (right figure) plots of human (blue solid line) and all AE models (dashed line). We match the mRE of the AE models with the human mRE (15.9%). The $MAD(\epsilon|\hat{v})$ and $mRE(\hat{v})$ of human annotations are computed within bins. The normalized bin sizes are shown as the bar plot.

with the samples in a dataset D . It is easy to check that the mean relative error of the α -Rényi model is $\gamma_\alpha \cdot mRE_\alpha$. Now, we can estimate γ_α with

$$\gamma_\alpha = \frac{mRE_{human}}{mRE_\alpha}, \quad (3.19)$$

where mRE_{human} is the mRE of human annotations.

Alpha estimation. We select the parameter α to best match with human errors. Our approach is to select from a range of discrete α values such that the Rényi model matches with human mRE and mAE simultaneously. For the annotated dataset (cat, dog, bird), We choose $\alpha = 1.1$, see Figure 3.7. A comparison of all AE models is shown in Figure 3.8.

3.4 Experimental results

3.4.1 Experimental settings

Datasets. We evaluate our approach on PASCAL VOC 2012 [42], MS COCO 2014 segmentation datasets [79], and 2017 ACDC Challenge¹ medical dataset. PASCAL dataset

¹<https://www.creatis.insa-lyon.fr/Challenge/acdc/>

contains 21 classes. We adopt the augmented training set with 10,582 images [44], following the common practice [25, 62]. Validation and testing contain 1449 and 1456 images. Scribble supervision of PASCAL dataset is from [77]. COCO has 81 classes with 80K training and 40K validation images. ACDC Challenge is to segment the left ventricular endocardium. We use the same experimental setup as [57], where the training and validation sets contain 1674 and 228 images. Exact size targets are from the ground truth (GT) masks.

Evaluation metrics for segmentation. We employ mean Intersection-over-Union (mIoU) as our evaluation criteria for PASCAL and COCO datasets. The performance of the PASCAL test set is evaluated on the official PASCAL VOC online evaluation server. Performance on ACDC dataset is measured by the mean Dice similarity coefficient (DSC).

Implementation details. We evaluate our approach with two types of ResNet-based and one ViT-based [40] segmentation models on the PASCAL and COCO datasets. ResNet-based models follow the implementation of DeepLabV3+ [28] using the backbone of ResNet101 (R101) or the backbone of WR38 [114]. For brevity, we name them R101-based or WR38-based DeepLabV3+ models. For the ViT-based network, We follow the implementation of Segmenter [101], adopting its ViT-B/16 backbone and linear decoder. For experiments on the ACDC datasets, we use MobileNetV2-based [97] DeepLabv3+ model. The R101, WR38, and MobileNetV2 backbones are ImageNet [96] pre-trained. ViT-B/16 backbone is pre-trained on ImageNet-21K [38] and fine-tuned on ImageNet-1k [96]. We directly evaluate our size-target approach on top of the standard architectures commonly used in full supervision without introducing any modification.

For training and testing, PASCAL and COCO images are resized to 512×512 , and ACDC medical images are resized to 256×256 . We employ color jittering and horizontal flipping for data augmentation. The models are trained using stochastic gradient descent (SGD): 60 epochs for PASCAL and COCO, and 200 epochs for ACDC, with a polynomial learning rate scheduler (power of 0.9). Batch sizes are set to 16 for ResNet and 20 for ViT models on PASCAL, 12 on ACDC, and 12 (ResNet) and 16 (ViT) for MS COCO. The initial learning rate (lr) is 0.005 for ACDC and PASCAL’s ResNet model, and 0.0005 for PASCAL’s ViT model. Initial lr on COCO is 0.0005 for ResNet and 0.0001 for ViT models. Loss function (3.2) is employed for size-target supervision. Loss (3.3) is only used for scribble supervision. The implementation of CRF loss (2.27) is the same as [105]. Size-target loss (3.1) and pCE (2.12) are used for medical images. Note that experiments on the whole datasets (Pascal and COCO) are conducted with synthetic size targets produced by the linear AE model. A comprehensive study of the advanced AE models on the whole PASCAL and COCO datasets will be our future work.

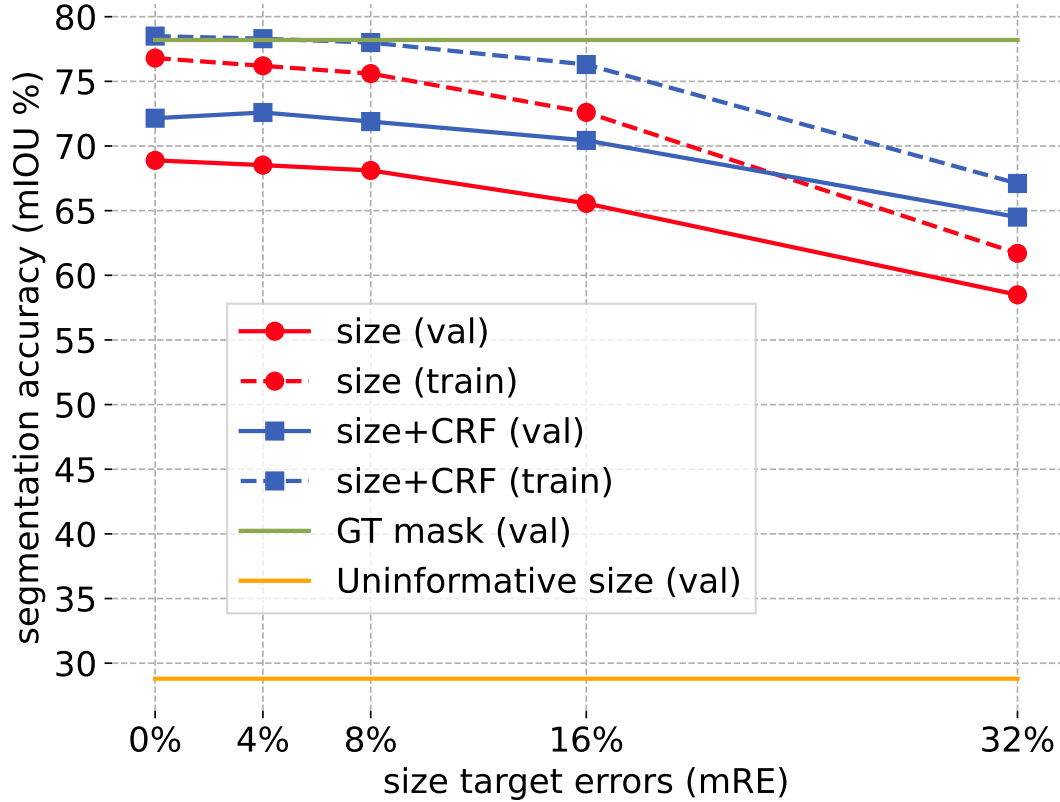


Figure 3.9: Segmentation accuracy for our approximate size-target supervision on PASCAL’s training (dashed lines) and validation data (solid lines). The segmentation is trained using losses (3.1) (red) or (3.2) (blue), where size targets are subject to various levels of corruption (3.6,3.8). The green line indicates the segmentation accuracy for full ground truth mask supervision. The orange line shows the segmentation accuracy obtained with uninformative (uniformly distributed) size targets with log barrier loss (2.18).

3.4.2 Approximate and uninformative size targets

Our approach works well with approximate size targets. Figure 3.9 illustrates the robustness of our approach to size errors. We train with synthetic size targets generated by varying corruption levels as defined in (3.8) using the linear AE model (3.7). The validation accuracy (solid red line) only drops slightly when mRE (3.5) remains below 16%. Utilizing

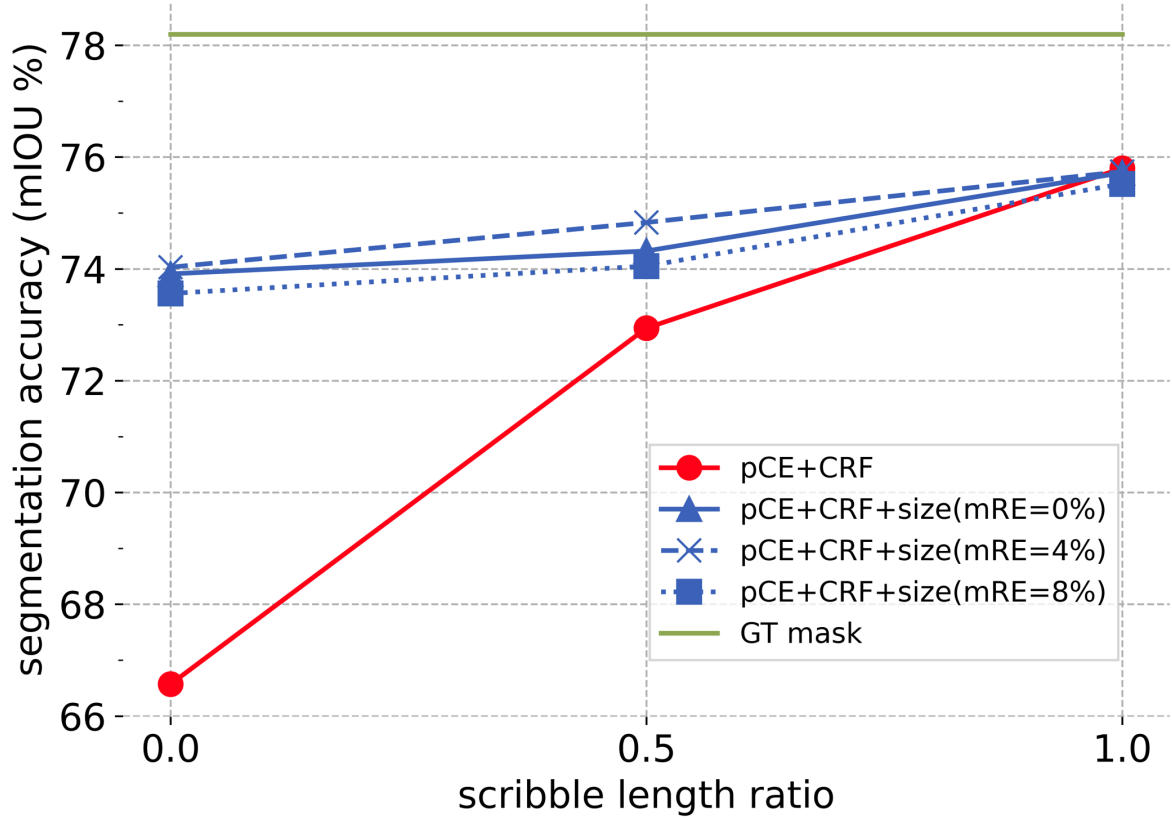


Figure 3.10: Segmentation accuracy on PASCAL validation set for scribble supervision of varying lengths with (blue curve) and without (red curve) size-target supervision. The line styles of the blue curves differentiate among various levels of corruption

CRF loss (2.27) further enhances the segmentation model’s robustness to noise (solid blue line). When mRE is 4%, there is a noticeable increase in validation accuracy. The downward trend of the training accuracy (dashed blue line) suggests that the observed increases in validation accuracy at $mRE = 4\%$ stem from improved neural network generalization quality. Without approximate size-target supervision, the performance of segmentation models trained with log barrier (2.18) collapses, see the orange line in Figure 3.9.

3.4.3 Enhancing scribble-based segmentation with size targets

Our size-target approach can be integrated with partial ground truth mask supervision (scribbles/seeds). Figure 3.10 shows the results of scribble-supervised semantic segmentation with or without size-target supervision. Synthetic size targets are generated by varying corruption levels as defined in (3.8) using the linear AE model (3.7). Size targets can significantly enhance performance, especially when the scribble length is short. The performance of the segmentation model trained without size targets degrades dramatically with the reduction of scribble length. In the extreme case, only one pixel is labeled for each object (scribble length ratio = 0.0), our size-target approach can boost the accuracy from 66.6% to 74%, which is close to the performance of full scribble supervision (scribble length ratio = 1.0).

3.4.4 Comparison with the state-of-the-art methods

Quantitative results. Our general training losses are applied to three standard architectures for semantic segmentation (R101-DeepLabV3+, WR38-DeepLabV3+, and ViT-Linear) as is, without any modification. Our results are highlighted in Table 3.2. Since our approach is completely general, it is possible to use it in specialized architectures or complex training procedures. Likely, this would further improve the results, but this is not the focus of our work. The rest of Table 3.2 shows the results for semantic segmentation methods (of different complexities) for weak and full supervision. Methods are divided into multi-stage and single-stage methods, grouped by their backbones. The multi-stage methods for weak supervision enhance their accuracy through complicated training schemes. Typical single-stage methods improve their results using complex architectural or training modifications like additional branches, extra refinement modules, or specialized training strategies. However, we achieve state-of-the-art using only standard segmentation architectures, commonly used in full supervision. A standard R101-based DeepLabV3+ model trained with approximate size targets ($mRE = 8\%$) approaches 92% (71.9/78.2) and 75% (45.0/60.4) of its full supervision performance on PASCAL and COCO datasets, respectively. The WR38-based DeepLabV3+ model trained with approximate size-target supervision surpasses other methods employing the same backbone by approximately 10% without any architectural or training modification. Using the standard vision transformer architecture with a simple linear decoder [101], size-target supervision achieves approximately 96% of the full supervision performance on the Pascal dataset. The size-target approach outperforms other complex single-stage methods on both datasets despite its simplicity.

Qualitative results. Figure 3.11 presents the qualitative examples of our method

Backbone	Decoder	Architectural modification	Supervision	PASCAL Val	PASCAL Test	COCO Val
<i>Multi-stage methods</i>						
R101	DeepLabV2	CLIMS [116]	tags	70.4	70.0	-
R101	DeepLabV2	SANCE [75]	tags	70.9	72.2	44.7
R101	DeepLabV3+	MARS [53]	tags	77.7	77.2	49.4
R101	DeepLabV2	BECO [89]	tags	72.1	71.8	45.1
R101	DeepLabV2	MatLabel [110]	tags	73.0	72.7	45.6
WR38	LargeFOV	MCT [118]	tags	71.9	71.6	42.0
WR38	LargeFOV	MCTOCR [32]	tags	72.7	72.0	42.5
SWIN	DeepLabV2	ReCAM [31]	tags	71.8	72.2	47.9
ViT-S	“Grad-clip”	WeakTr [129]	tags	78.4	79.0	50.3
<i>Single-stage (end-to-end) methods</i>						
R101	DeeplabV3+	-	size (8%)	71.9	72.4	45.0
R101	DeeplabV3+	-	full	78.2	78.2	60.4
WR38	DeepLabV3+	SSSS [10]	tags	62.7	64.3	-
WR38	Conv	RRM [123]	tags	62.6	62.9	-
WR38	DeeplabV3+	-	size (8%)	72.7	72.6	-
MiT-B1	MLP	AFA [94]	tags	66.0	66.3	38.9
ViT-B	DeepLabV2	ViTPCM [92]	tags	70.3	70.9	45.0
ViT-B	LargeFOV	ToCo [95]	tags	71.1	72.2	42.3
ViT-B	Conv	SeCo [121]	tags	74.0	73.8	46.7
ViT-B	LargeFOV	CoSA [120]	tags	76.2	75.1	51.0
ViT-B	Linear	-	size (8%)	78.1	78.2	56.3
ViT-B	Linear	-	full	81.4	80.7	-

Table 3.2: Semantic segmentation results (mIOU%) on PASCAL and COCO. The supervision column indicates a form of supervision: image-level class *tags*, *size* targets (our highlighted results), or *full* supervision with pixel-accurate masks. The percentage after “size” is the accuracy (mRE) of our corrupted size targets (3.6,3.8). Our approach does not require any complex architectural modification or multi-stage training procedures needed for tag supervision, see “Modification” column. We achieve state-of-the-art in image-level supervision using standard segmentation architectures common in full supervision, e.g. DeepLabV3+ (R101, WR38) [28] or ViT-B (Linear) [101]. Likely, our general approach can further improve by architectural modifications, but it is not our focus.

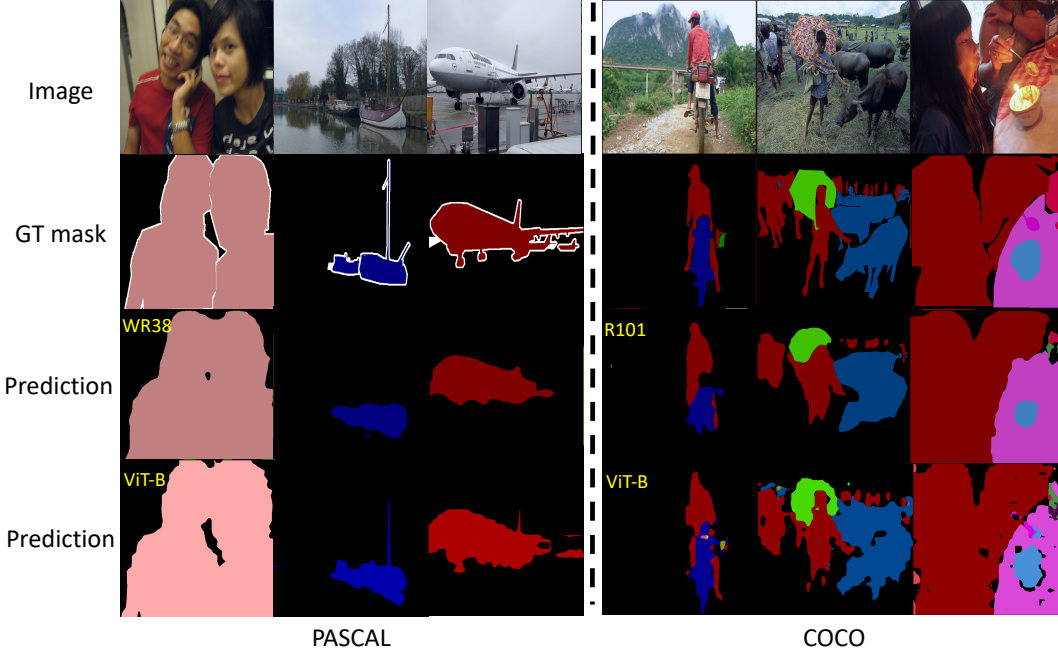


Figure 3.11: Segmentation examples using size-target supervision ($mRE = 8\%$). Model backbones are shown in the top-left corner of the predictions, see Table 3.2 for decoders.

on PASCAL (left) and COCO (right) validation sets. Despite size targets providing only image-level information, segmentation models can precisely identify object locations, eliminating the need for localization methods like CAM.

3.4.5 Medical data: size-target vs. size-barrier

Our method is also promising for medical image segmentation, benefiting from the consistency in object sizes across similar medical images, which healthcare professionals can easily estimate. Here methods are evaluated on the medical ACDC dataset. We compare our size-target approach with the thresholded size-barrier technique [57], proposed for the weakly supervised medical image semantic segmentation. The size-barrier loss enforces inequality size constraints. Given the lower bound of each class, the thresholded size-barrier loss is

$$\mathcal{L}_{size_barrier}(S) = \sum_k (\max\{a_k - \bar{S}^k, 0\})^2, \quad (3.20)$$

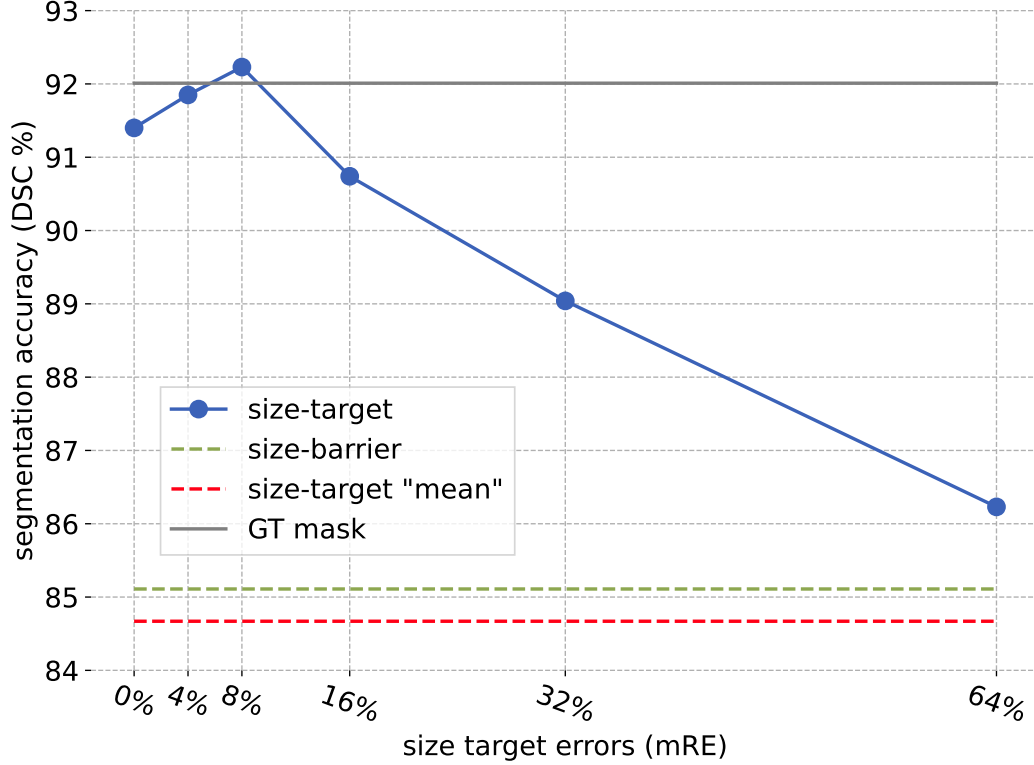


Figure 3.12: Size-targets (3.1) vs. size-barriers (3.20) on ACDC dataset. The accuracy of the binary segmentation models (MobileNetV2-based DeeplabV3+) is measured by *Dice coefficient* (DSC). The blue curve shows size-target accuracy against various levels of size corruption. The dashed green line shows the accuracy of the size-barrier technique [57]. The dashed red line shows the accuracy using the mean size target for all training images. The gray line indicates the full supervision result..

where a_k is a lower bound of class k . We train binary segmentation models with a combination of partial cross-entropy loss (2.12) and size constraint loss: size-target (3.1) or size-barrier (3.20). Seeds used in the experiments are obtained using the same method provided in [57]. The object and background barrier, a_{obj} and a_{bg} , are set based on [57]. In the size-barrier experiments, similarly to [57], we suppress the non-tag classes, using the loss $L_{sup}(S) = (\bar{S}^{obj})^2$. Conversely, size-target loss automatically suppresses non-tag classes, as discussed in Sec. 3.1. Figure 3.12 displays the segmentation accuracy against different

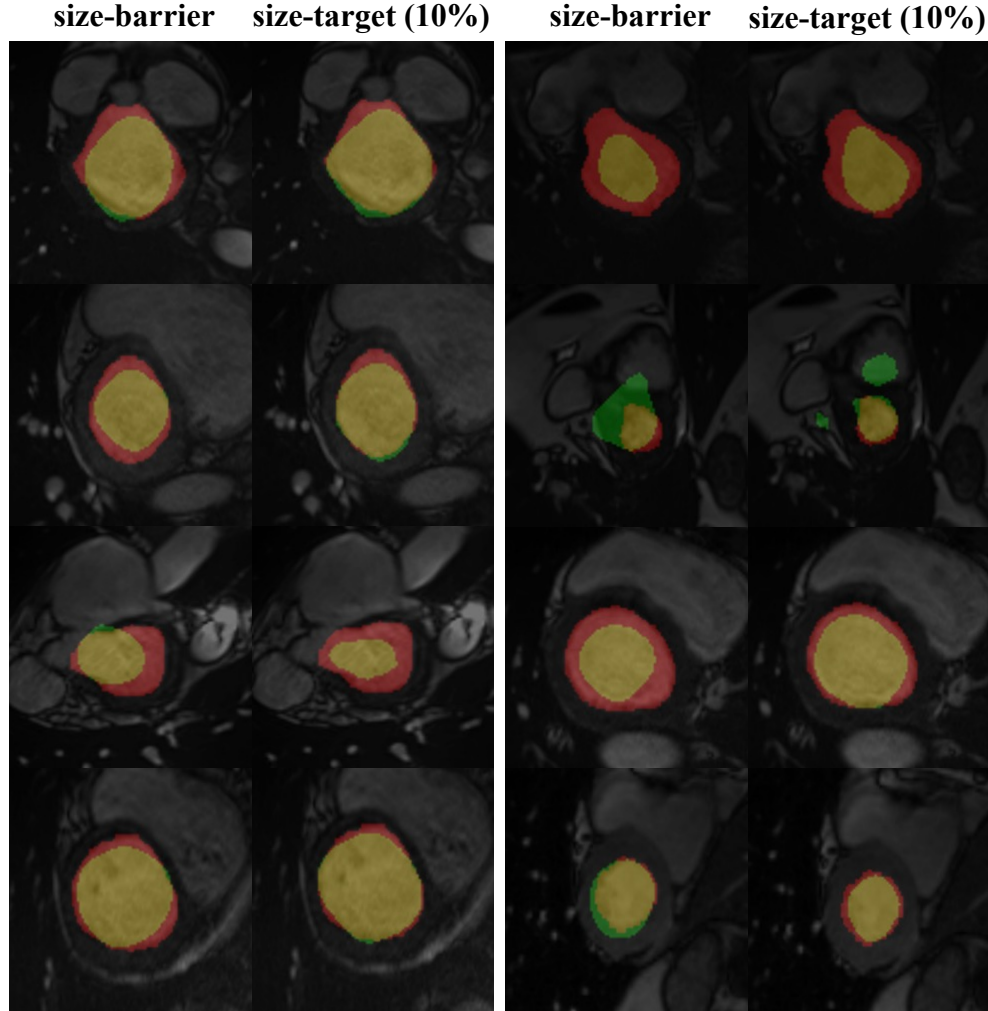


Figure 3.13: Randomly selected results comparing size-barrier (3.20) in [57] and our approximate size targeting (3.1) with $mRE = 8\%$. Yellow shows true positive pixels, green is false positive, and red is false negatives. Size barrier results tend to under-segment the objects, which may explain the differences shown in Fig.3.12.

levels of size target corruption. Our size-target loss consistently outperforms size-barrier loss, maintaining its superiority even when using highly noisy size targets. A peak in the accuracy curve is observed, which aligns with experimental results in Sec.3.4.2, showing that the noise within size targets can enhance the generalization ability of neural networks.

	ground truth masks	ground truth size targets	human-annotated size targets
cat	90.6	88.8	88.0
dog	88.1	84.3	84.5
bird	88.8	86.2	86.4

Table 3.3: Binary semantic segmentation accuracy (mIOU%) of three classes (cat, dog, bird). The table compares the accuracy of binary segmentation trained with three types of supervision (ground truth segmentation masks, ground truth size targets, and human-annotated size targets).

Our result using size targets with relative errors of 8% surpasses the full supervision performance indicated by the gray line. The dashed red line in the plot shows that using a fixed average size target across all training images can yield performance comparable to the size-barrier method. Figure 3.13 shows some qualitative examples of both methods. Size-target results are generally better than those of size barrier, which tends to under-segment the objects.

3.4.6 Segmentation with human-annotated size

Segmentation models trained with human-annotated size targets show robustness to human “heavy tail” errors. We compare the accuracy of segmentation models trained with human-annotated size targets and synthetic size targets in Figure 3.14. The accuracy for human-annotated size (indicated by the red star in the plot) approaches 97.2% (89.6%/92.2%) of the full supervision performance. Binary segmentation accuracy for each class is reported in Table 3.3. The binary segmentation results are consistent with the multi-class results. Binary segmentation models trained with human-annotated size targets are comparable to or outperform models trained with precise size-target supervision.

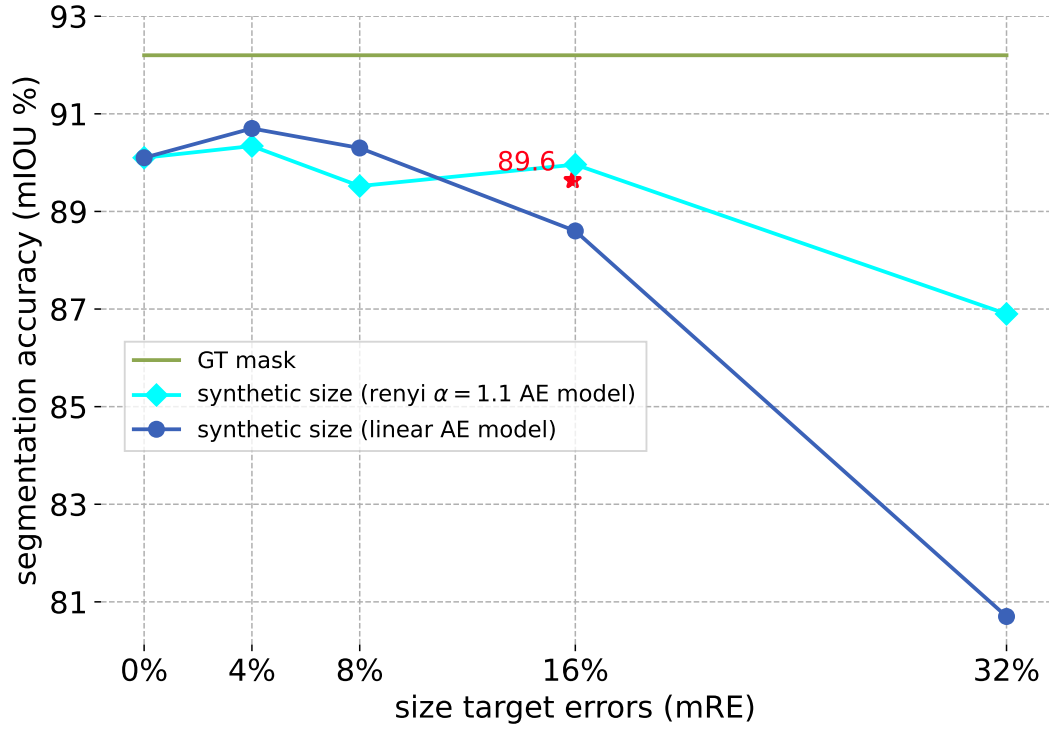


Figure 3.14: Segmentation accuracy with size-target supervision on three PASCAL classes (cat, dog, bird). Unlike Table 3.3, here we train multi-class segmentation models. The red star corresponds to human-annotated sizes ($mRE = 15.6\%$) and the blue curve indicates the accuracy for synthetic size targets generated by the linear AE model with different levels of corruption (3.7, 3.8). The cyan curve indicates segmentation performance for the synthetic size targets generated by Rényi AE model ($\alpha = 1.1$) with varying levels of corruption (3.15). The green line shows the segmentation accuracy for full supervision.

Chapter 4

Size Bias in MI Loss and Weakly-supervised Segmentation

Many low-level segmentation methods [130, 23, 93] estimate the appearance of segments with maximum likelihood. This standard log-likelihood term (2.37) can be interpreted as the entropy of colors in the segment (2.38). Optimizing the log-likelihood term breaks the image into coherent segments with low entropy of colors, which corresponds to entropy clustering criteria that encourage a simple appearance within the segment. This segmentation energy is connected to an information-theoretic approach: *mutual information* (MI) which measures the dependency between two random variables. From the MI point of view, the log-likelihood term encourages the alignment of the segmentation predictions and the color regions. Maximizing MI reduces the uncertainty of colors given the segmentation. Despite the popularity of MI in low-level segmentation, we found that it was underexplored in semantic segmentation. In this chapter, we propose using MI as an unsupervised loss for weakly-supervised semantic segmentation.

Here is the organization of this chapter. In Section 4.1, we introduce MI and its related work. The related work is separated by the two perspectives of MI: generative and discriminative view. The discriminative view of MI is closely related to its application in deep clustering, while the generative view of MI, entropy clustering, is widely used in low-level segmentation. In addition, we will discuss the strong volumetric bias in the MI-based loss found by prior work. In Sec. 4.2, we propose the MI loss for semantic segmentation. First, we propose an architectural design called the conditional posterior model that works well with the MI loss in semantic segmentation. Next, we discuss the volumetric bias in the MI loss and a solution inherited from [16]. Experimental results are shown in Sec. 4.3.

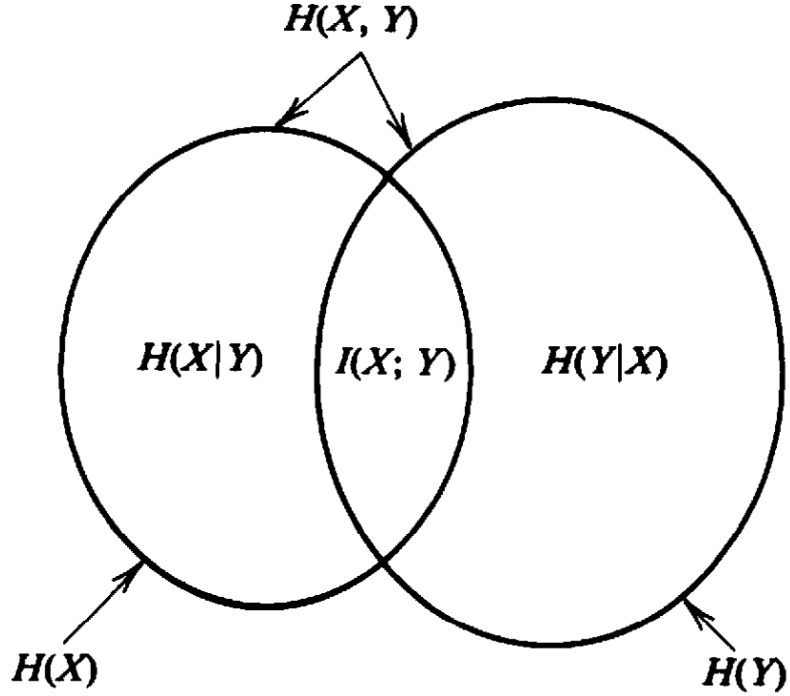


Figure 4.1: Relationship between entropy, conditional entropy, and mutual information. X and Y are two random variables. The figure shows that mutual information $I(X; Y)$ is the reduced uncertainty of X given Y and the reduced uncertainty of Y given X . Image from [35]

4.1 Mutual information loss

4.1.1 Formulation and related work

Formulation

Mutual information (MI) quantifies the shared information between two random variables, capturing how much knowing one variable reduces uncertainty about the other. For discrete random variables X and Y , with joint probability distribution $P(X, Y)$ and marginal probability distributions $P(X), P(Y)$, mutual information $I(X, Y)$ is defined through the concept of Kullback-Leibler (KL) divergence, which measures the difference between two probability distributions. Specifically, mutual information is the KL divergence between the joint distribution $P(X, Y)$ and the product of the marginal distributions $P(X)P(Y)$.

MI can be mathematically expressed as:

$$\mathcal{I}(X, Y) = \sum_{x \in X} \sum_{y \in Y} P_{(X,Y)}(x, y) \log \frac{P_{(X,Y)}(x, y)}{p_X(x)p_Y(y)} \quad (4.1)$$

From the MI definition, we can derive the following property, also see Figure 4.1.1, revealing its relationship with the entropy and the conditional entropy:

$$\mathcal{I}(X, Y) \equiv \underbrace{H(X) - H(X|Y)}_{\text{generative view}} \equiv \underbrace{H(Y) - H(Y|X)}_{\text{discriminative view}}, \quad (4.2)$$

where $H(\cdot)$ represents the entropy and $H(\cdot|\cdot)$ represents the conditional entropy.

Boudiaf et al. [14] referred to (4.2) as the two perspectives of MI, the generative view on the left and discriminative view on the right, in the context of representation learning. The generative view focuses on data distribution, while the discriminative view includes the probabilities of labels and the posterior probabilities. For representation learning, the X represents learned features that are movable and Y denotes predictions. The conditional entropy $H(X|Y)$ indicates features within the same cluster should be proximal, while $H(X)$ encourages all features to spread out in the feature space. The entropy $H(X)$ is crucial to avoid trivial representation learning solutions where all features collapse into one. In contrast, the X represents fixed features in the context of clustering. Thus, $H(X)$ remains constant. Note that the two views of MI are only theoretically equivalent, their practical implementations may diverge significantly. The generative approach utilizes density models to approximate the feature distribution, whereas the discriminative approach relies on posterior models for prediction. In practice, the equivalence between the two views is hard to achieve because of their different assumptions about model capacity and the hypothesis space.

Related discriminative work

Bridle and MacKay et al. [21] proposed to train neural networks with discriminative MI for unsupervised clustering. Neural networks output categorical distributions σ generated by the SoftMax function (2.4). The output $\sigma = \{\sigma^1, \dots, \sigma^K\}$ are interpreted as the posterior probabilities. MI measures the reduced uncertainty of the output given the input. Network parameters are optimized to maximize the MI between the input and output. The MI loss is formulated in Equation (4.3). See Theorem (4.1) for the proof.

Theorem 4.1 (MI loss for clustering in discriminative view [21]). *Notations X and K represent discrete random variables representing features and clusters. The σ is the output posterior $\Pr(k|x)$, and $\bar{\sigma}$ is the average of σ over all images in the dataset D . MI loss is the difference between the entropy of the average output $H(\bar{\sigma})$ and the average entropy of the output $\frac{1}{|D|} \sum_{i \in D} H(\sigma_i)$.*

$$\mathcal{I}(X; K) = H(\bar{\sigma}) - \overline{H(\sigma_i)} \quad (4.3)$$

Proof. From the definition of MI, we have

$$\begin{aligned} \mathcal{I}(X; K) &= \sum_{i=1}^N \sum_{k=1}^K \Pr(x_i, k) \log \frac{\Pr(x_i, k)}{\Pr(x_i) \Pr(k)} \\ &= \sum_{i=1}^N \sum_{k=1}^K \Pr(x_i) \Pr(k|x_i) \log \frac{\Pr(k|x_i)}{\Pr(k)} \\ &= \sum_{i=1}^N \Pr(x_i) \sum_{k=1}^K (\Pr(k|x_i) \log \Pr(k|x_i) - \Pr(k|x_i) \log \Pr(k)) \end{aligned}$$

Assume the neural network output σ represent posterior $p(k|x)$

$$\mathcal{I}(X; K) = \sum_{i=1}^N p(x_i) \sum_{k=1}^K (\sigma_i^k \log \sigma_i^k - \sigma_i^k \log \bar{\sigma}^k)$$

Apply Monte Carlo estimation

$$\begin{aligned} &\approx \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \sigma_i^k \log \sigma_i^k - \sum_{k=1}^K \bar{\sigma}^k \log \bar{\sigma}^k \\ &= -\overline{H(\sigma_i)} + H(\bar{\sigma}), \end{aligned}$$

□

Maximizing the MI encourages the neural network output to be as dependent on the input as possible. The two entropy terms in (4.3) can also be interpreted on their own. Minimizing $\overline{H(\sigma_i)}$ encourages the “decisiveness” or confidence of the predictions. The average entropy $H(\bar{\sigma})$ is minimized when the output is a one-hot distribution. Maximizing

$H(\bar{\sigma})$ encourages “fairness”. The entropy of average output $H(\bar{\sigma})$ is maximized when the neural network predicts equal-size clusters. The fairness term is important for avoiding trivial solutions in unsupervised clustering. Without the fairness term, the neural network can group all data into one cluster to satisfy the decisiveness. On top of (4.3), [65] proposed a regularization term such that the cluster boundary is restricted to be simple. These two works [21, 65] were before the deep learning era. Inspired by them, MI loss was widely used in deep clustering [22, 49, 50, 125].

Related generative work

The generative view is closely related to the entropy clustering which was detailed in Section 2.3.2. For clustering or segmentation, the entropy of features $H(X)$ remains constant. Maximizing the MI is equivalent to minimizing the conditional entropy $H(X|Y)$. Note that $H(X|Y)$ directly relates to entropy clustering (2.38), differing only by a constant factor. The generative view of MI is common in low-level segmentation as discussed in Sec 2.3.2. We show the generative MI is equivalent to entropy clustering in Theorem 4.2 in the context of low-level segmentation. For simplicity, we assume the low-level features are discrete RGB colors.

Theorem 4.2 (MI criterion in generative view). *Notations X and K are random variables representing colors and clusters. The S_p^k is the segmentation indicator at pixel p . The P^k represents the color density model. The Ω is the image domain. MI loss is equivalent to the entropy clustering (2.37)*

$$H(X|K) = -\frac{1}{|\Omega|} \sum_k \sum_{p \in \Omega} S_p^k \log P^k(x_p) \quad (4.4)$$

Proof. From the definition of conditional entropy, we have

$$H(X|K) = -\sum_k \Pr(k) \sum_x \Pr(x|k) \log \Pr(x|k)$$

Apply Monte Carlo estimation and approximate the feature density with P^k

$$\approx -\sum_k \frac{\Pr(k)}{|S^k|} \sum_{p \in S^k} \log P^k(x_p)$$

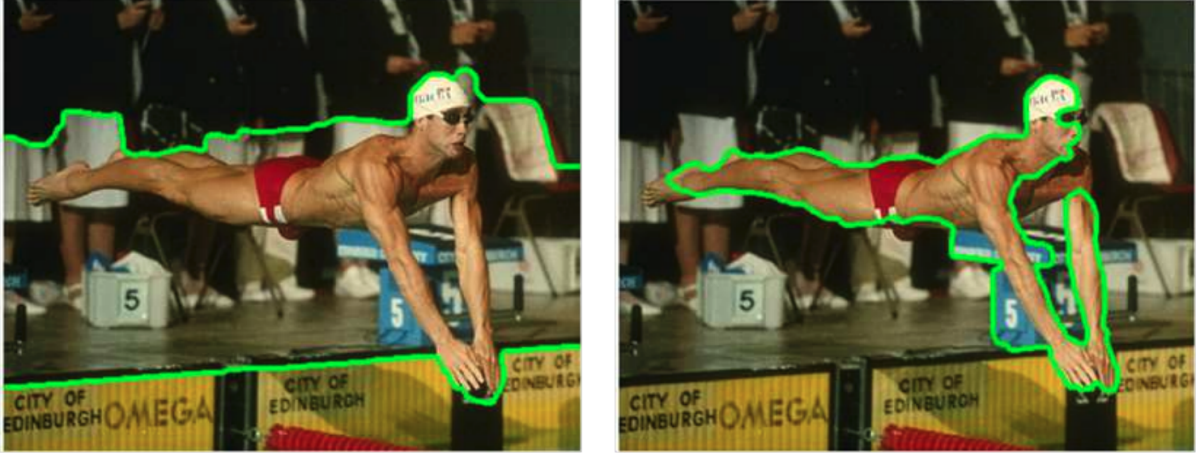


Figure 4.2: The left figure presents the low-level segmentation result by optimizing the standard log-likelihood energy (2.37). The green lines split the foreground and background areas in the image. It shows the volumetric bias toward equal-size segments in the standard entropy clustering energy or generative MI. The right figure presents the low-level segmentation result after fixing the volumetric bias with unbiased energy derived by [16]. Image from [16]

Introduce segmentation indicator S_p^k

$$\begin{aligned}
 &= - \sum_k \frac{\Pr(k)}{|S^k|} \sum_{p \in \Omega} S_p^k \log P^k(x_p) \\
 &= - \frac{1}{|\Omega|} \sum_k \sum_{p \in \Omega} S_p^k \log P^k(x_p)
 \end{aligned}$$

□

4.1.2 Uniform size bias in the context of low-level segmentation

The standard log-likelihood term (2.37) is closely associated with probabilistic K-means (2.36). Kearns et al. [56] pointed out this energy has a volumetric bias towards equal-size clusters. Boykov et al. [16] observed the same bias in low-level segmentation, see Figure 4.2. In the context of segmentation, the volumetric bias in generative MI loss (4.4) can be revealed by decomposing the MI:

$$\begin{aligned}
\mathcal{L}_{MI} &= -\frac{1}{|\Omega|} \sum_k \sum_{p \in \Omega} S_p^k \log P^k(x_p) \\
&= -\sum_k \frac{|S^k|}{|\Omega|} \sum_{p \in \Omega} \frac{S_p^k}{|S_k|} \log P^k(x_p) \\
&= \sum_k \frac{|S^k|}{|\Omega|} \mathbb{E}_{x_p \sim X^k} [-\log P^k(x_p)] \\
&= \sum_k \frac{|S^k|}{|\Omega|} \mathbb{E}_{x_p \sim X^k} [\log \frac{X^k(x_p)}{P^k(x_p)} - \log X^k(x_p)] \\
&= \sum_k \frac{|S^k|}{|\Omega|} H(X^k) + \frac{|S^k|}{|\Omega|} KL(X^k || P^k) \\
&= H(X|K) + \sum_k \frac{|S^k|}{|\Omega|} KL(X^k || P^k) \\
&= H(X) - H(K) + H(K|X) + \sum_k \frac{|S^k|}{|\Omega|} KL(X^k || P^k) \\
&\stackrel{c}{=} -H(K) + H(K|X) + \sum_k \frac{|S^k|}{|\Omega|} KL(X^k || P^k),
\end{aligned} \tag{4.5}$$

where X^k is the true color distribution, S_p^k is the segmentation indicator, Ω represents the image domain, and P^k is the color density model. We only need to focus on the last row of the decomposition (4.5) to identify the volumetric bias. It is easy to find that the last row of the decomposition (4.5) corresponds to the discriminative view of MI. The conditional entropy $H(K|X)$ encourages the decisiveness of the predictions given color X . It prefers each color to be uniquely associated with one class. The KL divergence term encourages the color model to estimate the true color density. The entropy term $H(K)$ is the one that causes the equal-size bias. Maximizing the entropy $H(K)$ is equivalent to minimizing the KL divergence between the distribution of K and the uniform distribution, as explained in (2.15).

Boykov et al. [16] proposed two solutions to the volumetric bias. The first method fixes the volumetric bias with the “correct” bias or size targets. The correct volumetric bias is enforced by reverse KL divergence as defined in (2.16). However, this method requires prior knowledge of true size targets. The second method eliminates the volumetric bias by

using weighted likelihood:

$$\mathcal{L}_{unbiased} = -\frac{1}{|\Omega|} \sum_k \sum_{p \in \Omega} S_p^k \log w^k P^k(x_p) \quad (4.6)$$

The two MI losses defined in (4.4) and (4.6) differ by a weight w^k in the log-likelihood. The weights w are free variables that can be estimated from the predictions. The optimal weights are the average of the current predictions: $S_p = (\bar{S}^1, \dots, \bar{S}^K)$ as defined in (2.14). Decomposing the unbiased log-likelihood, we have

$$\begin{aligned} \mathcal{L}_{unbiased} &= -\frac{1}{|\Omega|} \sum_{k \in K} \sum_{p \in \Omega} S_p^k \log w^k P^k(x_p) \\ &= \mathcal{L}_{MI} - \frac{1}{|\Omega|} \sum_{k \in K} \sum_{p \in \Omega} S_p^k \log w^k \\ &= \mathcal{L}_{MI} - \sum_{k \in K} \frac{|S^k|}{|\Omega|} \log w^k \\ &= \mathcal{L}_{MI} + \sum_{k \in K} \frac{|S^k|}{|\Omega|} \log \frac{|\Omega|}{w^k} - \sum_{k \in K} \frac{|S^k|}{|\Omega|} \log \frac{|S^k|}{|\Omega|} \\ &= \mathcal{L}_{MI} + KL(K||W) + H(K) \\ &\stackrel{c}{=} \sum_{k \in K} \frac{|S^k|}{|\Omega|} KL(X^k||P^k) + KL(K||W) + H(K|X) \end{aligned} \quad (4.7)$$

Comparing the decomposition of standard (4.5) and unbiased MI (4.7), it is easy to find that the entropy $H(K)$ is replaced with KL divergence $KL(K||W)$. When weights w are free variables, the KL divergence term $KL(K||W)$ can be optimized to be zero, eliminating the volumetric bias caused by $H(K)$. Note that unbiased MI only encourages the “decisiveness” with $H(K|X)$ and does not care about “fairness”.

4.2 MI for deep segmentation

MI loss (4.4) is commonly used in low-level segmentation [23, 93, 104, 130]. We propose to use it in weakly-supervised semantic segmentation as an unsupervised loss to align the segmentation predictions with the color features. In Section 4.2.1, we propose a conditional posterior model as an alternative to the standard SoftMax posterior model. This architectural modification is tailored to enhance the compatibility with the MI loss in semantic

segmentation. In Section 4.2.2, we follow the second solution proposed by [16] and present the unbiased MI loss for semantic segmentation.

4.2.1 Conditional posterior model and MI loss

Before introducing the conditional posterior model, we first attempt to use MI loss (4.4) with the standard SoftMax model commonly used in semantic segmentation. SoftMax models are trained on a dataset with a set of predefined classes K . Given an input image, the SoftMax model generates a categorical distribution for each pixel. The pixel-wise prediction S_p^k for class k is generated by the SoftMax function:

$$S_p^k = \frac{e^{l_p^k}}{\sum_{i \in K} e^{l_p^i}}, \quad (4.8)$$

where l is the logits. Such prediction is interpreted as the posterior probability $\Pr(k|x_p)$, where x_p is the color features of pixel p . The density models P^k in MI (4.4) are fitted by colors sampled from the posterior distribution. Specifically, for each pixel, a class k is sampled from the posterior distribution S_p , representing that this color will be used to fit the density model of segment k . During training, the MI loss (4.4) is optimized via block coordinate descent by alternating two steps. In the first step, the network predictions are fixed, and the density models P^k are optimized using colors sampled from the posterior distribution predicted by the network. Next, density models are fixed and the network parameters are optimized via gradient descent to minimize the MI loss. There are a few issues in the above algorithm.

First, as tag classes in an image are a subset of the predefined classes, fitting color models for the absent classes in the image is unnecessary. In addition, when sampling colors from the posterior distribution generated by the SoftMax model, many pixels are wasted because they are sampled into the set of non-tag classes. These colors do not contribute to the color model fitting that is useful for the segmentation. The two issues are caused by neglecting the tag information provided by the image. To fix these issues, we design a conditional posterior model and redefine the MI loss over the tag classes.

As tag supervision is available in weakly-supervised semantic segmentation, we define posterior probabilities conditioned on given image tags. The conditional prediction is formulated as follows

$$S_p^{k|T} = \frac{e^{l_p^k}}{\sum_{i \in T} e^{l_p^i}}, \quad (4.9)$$

where T is the set of tag classes, The prediction of tag classes S_p^T is defined as

$$S_p^T = \frac{\sum_{k \in T} e^{l_p^k}}{\sum_{i \in K} e^{l_p^i}}. \quad (4.10)$$

The conditional prediction $S_p^{k|T}$ can be interpreted as the conditional posterior:

$$\Pr(k|k \in T, x_p) = \frac{\Pr(k|x_p)}{\Pr(k \in T|x_p)} = \frac{S_p^k}{S_p^T} \quad (4.11)$$

We derive the conditional MI loss over the tag classes:

$$\mathcal{L}_{\text{MI}} = - \sum_{k \in T} \sum_{p \in \Omega} S_p^{k|T} \log \Pr(x_p|k). \quad (4.12)$$

The derivation process is the same as Theorem 4.2. Note that only the color models of the tag classes are required for this loss (4.12). It is easy to find that logits of non-tag classes do not affect the training when using conditional MI loss from (4.9). As a result, conditional MI loss doesn't guide the predictions of non-tag classes. We introduce suppression loss to prevent the neural networks from predicting non-tag classes. Suppression loss is formulated as follows:

$$\mathcal{L}_{\text{sup}} = - \sum_{p \notin \Omega_L} \log(1 - \sum_{k \notin T} S_p^k), \quad (4.13)$$

where Ω_L represents the seed pixels on partial ground truth masks (scribbles). The suppression loss is only applied to non-labeled pixels when partial ground truth masks are available. The MI loss (4.12) can be used together with size-target loss (3.1) when size-target supervision is available. When combining size-target loss with MI, the suppression loss can be dropped, because the size-target loss can suppress non-tag classes on its own, and we have tested that the results are similar when the suppression loss is not used.

4.2.2 Unbiased MI loss

The equal-size bias over the tag classes in the conditional MI loss (4.12) can be revealed by decomposing the loss function, like the decomposition shown in (4.5).

$$\begin{aligned}
\mathcal{L}_{MI} &= -\frac{1}{|\Omega|} \sum_{k \in T} \sum_{p \in \Omega} S_p^{k|T} \log P^k(x_p) \\
&= -\sum_{k \in T} \frac{|S^k|}{|\Omega|} \sum_{p \in \Omega} \frac{S_p^{k|T}}{|S^k|} \log P^k(x_p) \\
&= \sum_{k \in T} \frac{|S^k|}{|\Omega|} \mathbb{E}_{x_p \sim X^k} [-\log P^k(x_p)] \\
&= \sum_{k \in T} \frac{|S^k|}{|\Omega|} \mathbb{E}_{x_p \sim X^k} [\log \frac{X^k(x_p)}{P^k(x_p)} - \log X^k(x_p)] \\
&= \sum_{k \in T} \frac{|S^k|}{|\Omega|} H(X^k) + \frac{|S^k|}{|\Omega|} KL(X^k || P^k) \\
&= H(X|K) + \sum_{k \in T} \frac{|S^k|}{|\Omega|} KL(X^k || P^k) \\
&= H(X) - H(K) + H(K|X) + \sum_{k \in T} \frac{|S^k|}{|\Omega|} KL(X^k || P^k) \\
&\stackrel{c}{=} -H(K) + H(K|X) + \sum_{k \in T} \frac{|S^k|}{|\Omega|} KL(X^k || P^k),
\end{aligned} \tag{4.14}$$

where S is the segment, P is the color model, and X is the true color density. We inherit the solution to the volumetric bias from [16], and develop the unbiased conditional MI as follows:

$$\mathcal{L}_{\text{unbiased MI}} = -\sum_{k \in T} \sum_{p \in \Omega} S_p^{k|T} \log w^k \Pr(x_p|k), \tag{4.15}$$

where weights w are free variables that can be estimated from the average predictions. The following decomposition of the unbiased conditional MI shows that the volumetric bias is eliminated.

$$\begin{aligned}
\mathcal{L}_{\text{unbiased MI}} &= -\frac{1}{|\Omega|} \sum_{k \in T} \sum_{p \in \Omega} S_p^{k|T} \log w^k P^k(x_p) \\
&= \mathcal{L}_{\text{MI}} - \frac{1}{|\Omega|} \sum_{k \in T} \sum_{p \in \Omega} S_p^{k|T} \log w^k \\
&= \mathcal{L}_{\text{MI}} - \sum_{k \in T} \frac{|S^k|}{|\Omega|} \log w^k \\
&= \mathcal{L}_{\text{MI}} - \sum_{k \in T} \frac{|S^k|}{|\Omega|} \log w^k \tag{4.16} \\
&= \mathcal{L}_{\text{MI}} + \sum_{k \in T} \frac{|S^k|}{|\Omega|} \log \frac{S^k}{|\Omega| w^k} - \sum_{k \in T} \frac{|S^k|}{|\Omega|} \log \frac{|S^k|}{|\Omega|} \\
&= \mathcal{L}_{\text{MI}} + KL(K||W) + H(K) \\
&\stackrel{c}{=} \sum_{k \in T} \frac{|S^k|}{|\Omega|} KL(X^k||P^k) + KL(K||W) + H(K|X).
\end{aligned}$$

Both (equal-size) biased and unbiased condition MI losses are evaluated in the following experimental section.

4.3 Experimental results

4.3.1 Experimental settings

The experiments are conducted on the PASCAL VOC 2012 dataset [42]. Scribble supervision of the PASCAL dataset is from [77]. We employ mIoU (2.10) as our evaluation criteria and test the conditional MI losses with DeepLabV3+ semantic segmentation models using the backbone of ResNet101 (R101). The R101 backbone is pre-trained on the ImageNet dataset. Images are randomly rescaled and cropped into the size of 512×512 for training and testing for scribble-supervised semantic segmentation. For size-target supervision, we resize the images to 512×512 so that the size distribution of an image remains the same through training. We employ horizontal flipping and random Gaussian blur for data augmentation. Segmentation models are trained with stochastic gradient descent (SGD) for 60 epochs with a polynomial learning rate scheduler using a power of 0.9. The initial learning rate is 0.005. Segmentation models are trained with a combination of the conditional

MI loss (4.12), suppression loss (4.13), and pCE loss (2.12) on scribbles. For size-target supervision, we combine the conditional MI loss and the size-target loss (3.1).

4.3.2 Scribbles vs. size targets

We evaluate conditional MI losses on three forms of weak supervision for semantic segmentation: scribbles, tags, and size targets. We compare the effectiveness of the conditional unbiased and biased MI losses against the standard CRF loss. For simplicity, the word “conditional” will be dropped when we discuss MI for semantic segmentation, as it is obvious we are referring to the MI losses defined in (4.12) and (4.15). In addition, we are also interested in comparing the performance of scribble and size-target supervision for semantic segmentation. Figure 4.3 combines the MI results in Chapter 4 and size-target results from Chapter 3, providing a comprehensive evaluation of the impact of size constraints in semantic segmentation, which is the core of this thesis.

MI with scribbles. The experimental results are presented with different lengths of scribble supervision. For full scribble supervision, the unbiased MI outperforms the biased MI and is comparable with the CRF loss. However, the performance of unbiased MI and CRF loss collapses quickly with the decrease in the scribble length. It shows that the segmentation models tend to over-segment the background when the scribbles are short, see qualitative results in Figure 4.4. Note that this behavior is not penalized by the unbiased MI as the entropy term encouraging “fairness” shown in (4.14) is eliminated. The shrinking bias is also in CRF loss [15]. In contrast, the biased MI is more robust to weak scribble supervision. The biased MI loss can still achieve 70% accuracy for point supervision.

MI with tags. Without using any size prior, segmentation models trained by only suppressing non-tag classes generate trivial solutions where all pixels are predicted as the background class, see 1st column in Figure 4.4. This is likely due to the imbalance of the dataset, where the background class is prevalent in the dataset. Unbiased MI and CRF losses do not penalize such trivial solutions. The fairness term in biased MI discourages the trivial solutions, but as it is based on the reverse KL divergence (2.15), it is not strong enough to prevent the background class from taking over all pixels. Using log barrier (2.18) can avoid such trivial solutions preventing the collapse in performance. However, due to the strong equal-size volumetric bias in the log barrier loss (2.18), the segmentation performance is relatively low, approximately 30% mIoU, for all three types of regularization. This equal-size bias is obvious in the qualitative examples in Figure 4.4 (2nd column). Pathak et al. [86] obtained slightly higher accuracy, 36% mIoU, with tag supervision. However,

this could be attributed to a more powerful, but more expensive, self-labeling optimization technique using auxiliary variables (pseudo labels) and an EM algorithm. In contrast, we use the plain gradient descent for evaluating our losses. It is also possible to achieve stronger segmentation results with tag supervision using complex architectural modifications or multi-stage training schemes, but it is not the focus of this thesis either. For fairness, we only compare single-stage methods with standard segmentation architectures (e.g. DeepLab [28]), commonly used in fully-supervised semantic segmentation.

MI with size targets. The segmentation results for size-target supervision are shown in the right three bars in Figure 4.3. In Comparison with tag supervision, size targets address the detrimental equal-size bias and dramatically boost the segmentation performance. The biased MI achieves 73.8% accuracy with approximate size targets ($mRE = 8\%$), surpassing unbiased MI and CRF.

As shown by the experiments in Chapter 3, size targets have the potential to improve the performance of scribble-supervised segmentation. The hollow bars overlaid on the solid color bars signify the increase in accuracy achieved by incorporating size-target supervision. Notably, when scribbles and size targets are combined, the biased MI loss demonstrates higher accuracy compared to the other two regularization losses. Overall, the MI loss performs comparably to or surpasses the CRF loss across all three types of supervision.

Scribbles vs. size targets Now, let us reassess the size target approach discussed in Chapter 3. Here, our focus is on comparing size targets with the commonly used scribble supervision for semantic segmentation.

We compare the highest segmentation results achieved by any of the three regularization losses between size targets and scribbles. The size-target approach (with biased MI) attains 97.2% (73.7/75.8) performance of full scribble supervision despite size targets being much cheaper image-level supervision than scribbles.

Moreover, segmentation models trained on size targets outperform scribble-supervised segmentation models, when scribble lengths are one click or 30%. These experimental findings further validate the effectiveness of the size-target approach and underscore the importance of volumetric cues in semantic segmentation.

In conclusion, both location cues (seeds) and volumetric cues (size targets) are sufficient for training semantic segmentation models, enabling them to achieve performance comparable to segmentation models trained on full supervision. However, obtaining satisfactory results without any information on objects’ location or size (e.g., tag supervision) is challenging. Using complex architectural modifications or multi-stage training schemes can boost performance, but this is outside the scope of this thesis.

The core contribution of this thesis lies in providing a new option of weak supervision for the community. The introduction of size targets as a new form of supervision underscores the effectiveness of volumetric information, which is as potent as conventional location cues.

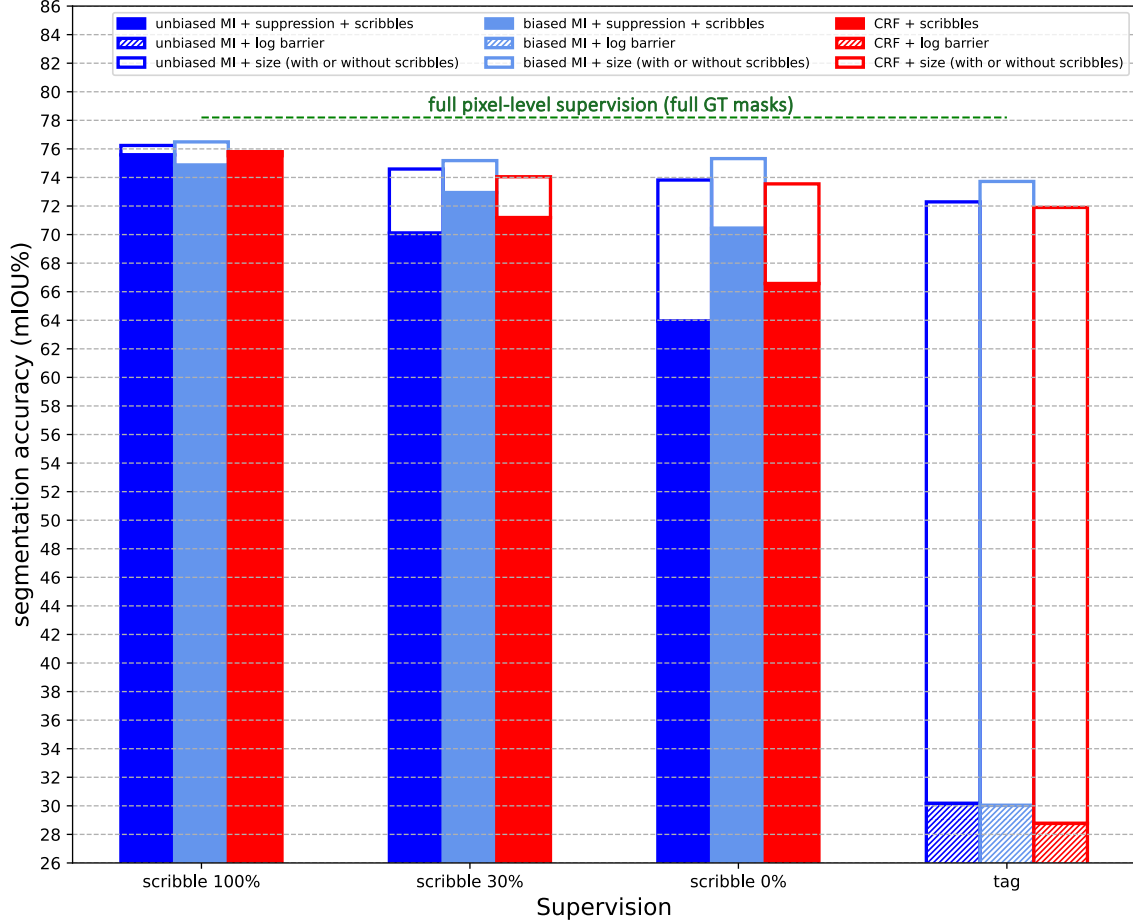


Figure 4.3: Performance (mIoU%) comparison of R101-based DeeplabV3+ segmentation models trained on scribble or image-level supervision. Segmentation models are evaluated on varying scribble lengths from full-length scribble (100%) to one click (0%). We use pCE loss (2.12) on scribbles (solid color) and log barrier (2.18) on tag supervision (shaded color), respectively. For each type of supervision shown in the horizontal axis, we employ three types of regularization loss: unbiased MI (blue) (4.15), biased MI (light blue) (4.12), and CRF loss (red) (2.27). Note that MI is used hand-in-hand with suppression loss when only scribbles are available. The hollow bars indicate the performance increase using size-target loss (3.1) with approximate size targets ($mRE = 8\%$). The log barrier is replaced with the size-target loss when size targets are available. For scribble supervision, size-target loss is employed together with pCE loss. As a baseline reference, we provide the green line indicating the segmentation accuracy for full pixel-level supervision with complete pixel-precise ground truth (GT) masks.

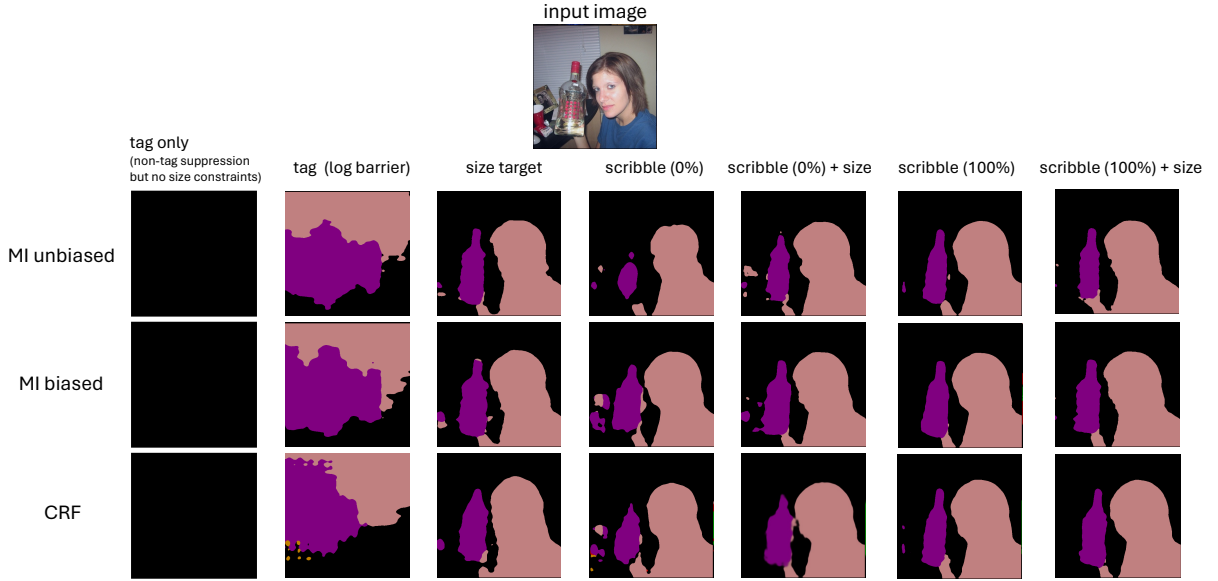


Figure 4.4: Qualitative examples of segmentation models trained with different supervision are presented in different columns. Different rows display various types of regularization losses. The first column shows “tag only” supervision, indicating the use of the tag class without any enforced size constraints. In contrast, the second column assumes a uniform volumetric distribution by utilizing the log barrier loss (2.18) on tag classes. The “size target” or “+ size” label indicates the utilization of approximate size-target supervision ($mRE = 8\%$). Lastly, the “scribble” column represents models trained with scribble supervision. The percentage in parentheses denotes the length of the scribbles.

Chapter 5

Size Prediction: Between Regression and Classification

In Chapter 3 and 4, we demonstrate that approximate size targets are effective for semantic segmentation. In this chapter, we aim to streamline the size annotation process and predict the size targets of an image with neural networks. We know size targets (2.14) are by-products of segmentation models. However, we focus on cheaper architectures without any decoder (e.g. classification networks) to directly predict sizes. We treat segmentation models as the upper bound baseline for our size predictors.

Here is the setup of our size prediction problem. Let D be the training dataset for size prediction models $\phi_{[W,\theta]}$, where θ is the parameters of the neural network backbone and W is the parameters of the linear prediction head. The training set D has K pre-defined classes. Each data in D is a pair $(I_i, \hat{v}_i)$, where I_i is the i th input image and $\hat{v}_i \in \mathbb{R}^K$ represents the ground truth size targets of the image I . The goal is to learn neural network parameters (W and θ) on dataset D so that it can be used to predict the size targets of an unseen image.

The organization of this chapter is as follows. In Section 5.1, we first assess the naive methods that directly use regression or classification architectures for size prediction. Then, we propose an effective architectural modification. Section 5.2 presents two losses used for size prediction: *mean squared error* (MSE), and KL divergence. The experimental results are in Section 5.3.

5.1 Architectures

Size prediction can be approached as either a regression or a “soft” classification problem, differing in how predictions are made. In regression, the linear prediction head directly produces size predictions of all classes. Conversely, in “soft” classification, a non-linear function converts the prediction scores into a categorical distribution. However, unlike standard classification tasks, where the ground truth target is a one-hot distribution indicating a single correct class, the ground truth labels $\hat{v}$ are soft targets.

In Section 5.1.1, we explore the naive approaches, attempting to solve size prediction via regression or “soft” classification based on global features. In Section 5.1.2, we propose aggregating predictions derived from local features. This shift highlights the architectural change from averaging features to averaging predictions.

5.1.1 Naive approaches: regression and “soft” classification

Regression formulation. Let $\bar{f}$ be the spatially averaged (global) deep features encoded by the network backbone ψ , the size prediction of class k is

$$v_i^k = \langle W^k, \bar{f}_i \rangle, \quad (5.1)$$

where W is the weights of the linear prediction head, v^k is the predicted size of the k th class. Specifically, $\bar{f}_i$ is defined as

$$\bar{f}_i = \text{GAP}(\psi_\theta(I_i)), \quad (5.2)$$

where ψ is the backbone with parameters θ and GAP denotes the global average pooling layer. Note that the predictions generated by (5.1) do not necessarily form a distribution that sums to one.

“Soft” classification formulation. In “soft” classification, the prediction is

$$v_i^k = \sigma^k(\langle W, \bar{f}_i \rangle), \quad (5.3)$$

where σ is the SoftMax function (2.4). Because of the SoftMax function, prediction v_i must be a distribution. We can employ a KL-based loss function to predict a distribution v_i that closely approximates the true size target $\hat{v}_i$. While the “soft” classification model may seem like a reasonable solution to the size prediction problem, it in fact fails to achieve satisfactory results. Unlike standard classification tasks in which the linear classifier learns

decision boundaries, the weights W of the size prediction head must coordinate across all classes to precisely predict the size. While weights W may successfully predict one size target, they may struggle to predict others. This issue arises from the SoftMax function, which causes the prediction of one class to depend on the predictions of others.

To mitigate the heavy lifting of the size prediction head, we can use the Sigmoid function (2.3) to predict the size target of each class independently. The predictions are formulated as

$$v_i^k = s^k(\langle W, \bar{f}_i \rangle), \quad (5.4)$$

where s is the Sigmoid function. Since the Sigmoid function s^k is solely associated with W^k , it decouples the size prediction v^k from the predictions of other classes.

5.1.2 From averaging features to averaging predictions

In the naive approaches, we average local features before predicting size targets with the linear prediction head. The averaging operation can lead to information loss. To fix this problem, we adjust the architectures. Unlike standard classification networks that average features before making predictions, We propose to derive predictions from local features and average the predictions. Our model is defined as

$$v_i^k = GAP(s^k(\langle W, f_i \rangle)), \quad (5.5)$$

where $f_i \in \mathbb{R}^{d \times m \times n}$ is the deep features with spatial resolution $m \times n$ and feature dimensions d , W is the weights of the linear prediction head, s is the Sigmoid function, and GAP is the global average pooling layer. This architecture offers some advantages. Firstly, prediction is made directly using local features, eliminating any loss of information through feature aggregation. Furthermore, the local prediction can be interpreted as classifying local features, which enables us to use the SoftMax function. As an analogy to segmentation, when the spatial resolution of deep features f is large, it is reasonable to associate local features with one correct class, making the SoftMax function a suitable choice. The average prediction naturally represents the size distribution defined in (2.14). With SoftMax, our model is formulated as

$$v_i^k = GAP(\sigma^k(\langle W, f_i \rangle)), \quad (5.6)$$

where σ is the SoftMax function. Note that such architectural modification is not necessary for the regression approach. As no non-linear function is applied in the regression prediction head, averaging features and averaging predictions are equivalent.

5.2 Losses for supervision with “soft” distribution targets

In the “soft” classification approach (naive or averaging predictions), we employ forward KL divergence loss when using the SoftMax activation function. The forward KL divergence loss is defined as

$$\mathcal{L}_{\text{KL}} = - \sum_k \hat{v}^k \log v^k. \quad (5.7)$$

Alternatively, we can use the binary KL divergence (BKL) loss with the Sigmoid activation function:

$$\mathcal{L}_{\text{BKL}} = - \sum_{k \in T} \hat{v}^k \log v^k + (1 - \hat{v}^k) \log(1 - v^k), \quad (5.8)$$

where T denotes the set of tag classes. Technically, the above loss should be referred to as K-class binary KL divergence. However, since we are only comparing three losses and there should be no confusion, we simplify it by dropping the “K-class” designation.

We adopt mean squared error (MSE) loss in the regression approach (naive or averaging predictions). The MSE loss is defined as

$$\mathcal{L}_{\text{MSE}} = \sum_{k \in T} (v^k - \hat{v}^k)^2. \quad (5.9)$$

MSE and BKL losses only involve tag classes because we observe that summing over all classes leads to a strong bias toward underestimating the size. This bias may be attributed to over-fitting the zero size targets of non-tag classes.

5.3 Experimental results

5.3.1 Experimental settings

Our size prediction algorithms are evaluated on the PASCAL VOC 2012 [42] dataset. The ground truth size targets are extracted from the ground truth masks. We employ mean relative error (mRE) and root mean squared error (L2) as our evaluation criteria. We use R101-based DeepLabV3+ semantic segmentation models trained on size-target loss and CRF as the upper bound baseline. We adopt the R101 as the backbone and the

linear layer as the prediction head for our size prediction models. The spatial resolution of deep features is controlled by using more convolutional layers whose stride is 2. We build two types of architectures based on averaging features and averaging predictions. As described in Section 5.1.2, the two architectures differ in the position of the GAP layer. For averaging-feature architectures, we put the GAP layer after the backbone followed by the linear layer, which is common for classification networks. For averaging-prediction architectures, the linear layer is after the backbone, followed by the GAP layer which is applied to predictions. All R101 backbones are pre-trained on the ImageNet dataset. Images are resized to 512×512 for training and testing. We employ horizontal flipping and random Gaussian blur for data augmentation during training. Models are trained with stochastic gradient descent (SGD) for 60 epochs with a polynomial learning rate scheduler using a power of 0.9. The initial learning rate is 0.005.

5.3.2 Size prediction quality

Table 5.1 compares our size prediction methods. The KL divergence loss performs the worst on the naive architecture. Its mean L2 error is much higher than other methods. Using the same naive architecture, the BKL loss (2nd row) effectively decouples the prediction of each class, resulting in a significantly lower L2 error. Generally, the results from the naive architecture (average features) are inferior to the architectures that average predictions. The increase in accuracy may be attributed to the elimination of information loss via feature averaging. Additionally, some spatial resolution of the features can benefit size prediction.

For architectures that average predictions (4 - 7 rows), KL loss shows a lower mRE, while BKL performs better with L2 error. Mean relative error places more weight on small objects compared to the L2 error. Therefore, the KL loss works better for small objects, while BKL performs well in predicting the size of large objects. The KL result in the 7th row indicates that we achieve performance close to the segmentation baseline without using any decoder in the architecture or regularization loss.

Architecture	Loss	mRE	mean L2 error
Avg features (1×1)	KL	1.923	0.236
Avg features (1×1)	BKL	2.099	0.095
Avg features (1×1)	MSE	2.219	0.080
Avg predictions (16×16)	BKL	1.251	0.068
Avg predictions (64×64)	BKL	1.310	0.069
Avg predictions (16×16)	KL	0.645	0.105
Avg predictions (64×64)	KL	0.515	0.089
Segmentation (512×512)	Size target + CRF	0.415	0.082

Table 5.1: Comparison of size prediction models. Two types of architectures are shown in the architecture column: average features and average predictions. The number in the parenthesis represents the spatial resolution of the deep features. The last row is our upper bound baseline, which is the segmentation model with an encoder-decoder architecture. Loss functions are shown in the second column. The mRE and mean L2 error of the validation set are reported in the right two columns.

Chapter 6

Conclusion and Future Work

In this thesis, we studied three related problems about size constraints in segmentation and size predictions. First, we proposed a new image-level supervision for semantic segmentation: size targets. Such targets could be approximate. In fact, our results suggest that some errors can benefit generalization. The size-target annotation requires little extra effort compared to the standard image-level tags. Annotators are simply tasked with predicting the area fractions of each class, which is much cheaper than labeling the full pixel-accurate ground truth masks. On top of this size-target supervision, we proposed an effective size-target loss based on KL divergence between the approximate size targets and the average prediction. In combination with the standard CRF-based regularization loss, our approximate size-target supervision on standard segmentation architectures (DeepLab and ViT) achieves segmentation accuracy comparable to the quality of the full pixel-precise supervision of the same models. Using standard semantic segmentation architectures, our general easy-to-reproduce approach outperforms significantly more complex weakly-supervised techniques based on model modifications and multi-stage training procedures.

In the second project, we proposed an unsupervised loss for semantic segmentation based on mutual information. Compared with CRF loss, the biased mutual information loss is more robust when the scribble supervision is weak. Overall, the MI loss performs comparably to or surpasses the CRF loss for size-target and scribble supervision. Furthermore, we reassess the effectiveness of size targets in Figure 4.3. We find that size targets are a low-cost alternative to scribbles. The segmentation performance with size targets is comparable to the models trained on full supervision. The performance decline observed with the decrease in scribble lengths does not occur under size-target supervision.

Lastly, we proposed a simple architectural modification for size prediction, which averages predictions rather than features. Combined with the standard KL divergence loss, our size prediction model achieved performance comparable to the upper bound baseline that uses more complex segmentation architectures.

In the future, we plan to annotate size targets of the whole Pascal training dataset, which will allow us to comprehensively assess the size-target approach. Additionally, it enables us to evaluate our advanced AE models with more human-labeled data. We also aim to further improve the size prediction model to ultimately automate the size target annotation process.

References

- [1] Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.
- [3] Abien Fred Agarap. Deep learning using rectified linear units (relu), 2019.
- [4] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2209–2218, 2019.
- [5] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4981–4990, 2018.
- [6] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.
- [7] Luis B Almeida. Multilayer perceptrons. In *Handbook of Neural Computation*, pages C1–2. CRC Press, 2020.
- [8] Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. Fluid annotation: A human-machine collaboration interface for full image annotation. In *Proceedings of the 26th ACM international conference on Multimedia*. ACM, oct 2018.

- [9] Mykhaylo Andriluka, Jasper RR Uijlings, and Vittorio Ferrari. Fluid annotation: a human-machine collaboration interface for full image annotation. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1957–1966, 2018.
- [10] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020.
- [11] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [12] A. Bearman, O. Russakovsky, V. Ferrari, and F. Li. Semantic segmentation with point supervision. In *ECCV*, 2015.
- [13] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What’s the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.
- [14] Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *European conference on computer vision*, pages 548–564. Springer, 2020.
- [15] Boykov and Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proceedings Ninth IEEE international conference on computer vision*, pages 26–33. IEEE, 2003.
- [16] Yuri Boykov, Hossam Isack, Carl Olsson, and Ismail Ben Ayed. Volumetric bias in segmentation and reconstruction: Secrets and solutions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1769–1777, 2015.
- [17] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.
- [18] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 105–112. IEEE, 2001.

- [19] Yuri Y Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images.
- [20] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 105–112 vol.1, 2001.
- [21] John Bridle, Anthony Heading, and David MacKay. Unsupervised classifiers, mutual information and phantom targets. *Advances in neural information processing systems*, 4, 1991.
- [22] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.
- [23] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.
- [24] Arslan Chaudhry, Puneet K Dokania, and Philip HS Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1707.05821*, 2017.
- [25] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.
- [26] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan 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, 2017.
- [27] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [28] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

- [29] Qi Chen, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4288–4298, 2022.
- [30] Zhaozheng Chen and Qianru Sun. Extracting class activation maps from non-discriminative features as well. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3135–3144, 2023.
- [31] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2022.
- [32] Zesen Cheng, Pengchong Qiao, Kehan Li, Siheng Li, Pengxu Wei, Xiangyang Ji, Li Yuan, Chang Liu, and Jie Chen. Out-of-candidate rectification for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23673–23684, 2023.
- [33] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16794–16804, 2021.
- [34] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [35] Thomas M Cover, Joy A Thomas, et al. Entropy, relative entropy and mutual information. *Elements of information theory*, 2(1):12–13, 1991.
- [36] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.
- [37] Andrew DeLong, Anton Osokin, Hossam N Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *International journal of computer vision*, 96:1–27, 2012.

- [38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [39] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, 2004.
- [40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [41] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4320–4329, 2022.
- [42] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308, 2009.
- [43] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [44] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 international conference on computer vision*, pages 991–998. IEEE, 2011.
- [45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [46] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1945–1954, 2018.
- [47] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7334–7344, 2019.

- [48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [49] Mohammed Jabi, Marco Pedersoli, Amar Mitiche, and Ismail Ben Ayed. Deep clustering: On the link between discriminative models and k-means. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1887–1896, 2019.
- [50] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.
- [51] Peng-Tao Jiang, Yuqi Yang, Qibin Hou, and Yunchao Wei. L2g: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16886–16896, 2022.
- [52] Huang Jin Jie and Putra Wanda. Runpool: A dynamic pooling layer for convolution neural network. *Int. J. Comput. Intell. Syst.*, 13(1):66–76, 2020.
- [53] Sanghyun Jo, In-Jae Yu, and Kyungsu Kim. Mars: Model-agnostic biased object removal without additional supervision for weakly-supervised semantic segmentation. *arXiv preprint arXiv:2304.09913*, 2023.
- [54] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency for object segmentation from image level labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4410–4419, 2017.
- [55] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X Yu. Adaptive affinity fields for semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 587–602, 2018.
- [56] Michael Kearns, Yishay Mansour, and Andrew Y Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. *Learning in graphical models*, pages 495–520, 1998.
- [57] Hoel Kervadec, Jose Dolz, Meng Tang, Eric Granger, Yuri Boykov, and Ismail Ben Ayed. Size-constraint loss for weakly supervised CNN segmentation. In *Medical Imaging with Deep Learning*, 2018.

- [58] Amjad Rehman Khan, Siraj Khan, Majid Harouni, Rashid Abbasi, Sajid Iqbal, and Zahid Mehmood. Brain tumor segmentation using k-means clustering and deep learning with synthetic data augmentation for classification. *Microscopy Research and Technique*, 84(7):1389–1399, 2021.
- [59] Beomyoung Kim, Sangeun Han, and Junmo Kim. Discriminative region suppression for weakly-supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1754–1761, 2021.
- [60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [61] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [62] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 695–711. Springer, 2016.
- [63] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.
- [64] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, pages 513–521. PMLR, 2013.
- [65] Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. *Advances in neural information processing systems*, 23, 2010.
- [66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [67] V. Kulharia, S. Chandra, A. Agrawal, P. Torr, and A. Tyagi. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *ECCV’20*.

- [68] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [69] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series.
- [70] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [71] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 34:27408–27421, 2021.
- [72] Minhyun Lee, Dongseob Kim, and Hyunjung Shim. Threshold matters in wsss: manipulating the activation for the robust and accurate segmentation model against thresholds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4330–4339, 2022.
- [73] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5495–5505, 2021.
- [74] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *2009 IEEE 12th international conference on computer vision*, pages 277–284. IEEE, 2009.
- [75] Jing Li, Junsong Fan, and Zhaoxiang Zhang. Towards noiseless object contours for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16856–16865, 2022.
- [76] Yi Li, Yiqun Duan, Zhanghui Kuang, Yimin Chen, Wayne Zhang, and Xiaomeng Li. Uncertainty estimation via response scaling for pseudo-mask noise mitigation in weakly-supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1447–1455, 2022.
- [77] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016.

- [78] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.
- [79] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [80] Sheng Liu, Kangning Liu, Weicheng Zhu, Yiqiu Shen, and Carlos Fernandez-Granda. Adaptive early-learning correction for segmentation from noisy annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2606–2616, 2022.
- [81] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [82] ML4Devs.com. An engineer’s trek into machine learning.
- [83] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.
- [84] Junwen Pan, Qi Bi, Yanzhan Yang, Pengfei Zhu, and Cheng Bian. Label-efficient hybrid-supervised learning for medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2026–2034, 2022.
- [85] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015.
- [86] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1796–1804, 2015.
- [87] Renfrey Burnard Potts. Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press, 1952.

- [88] Jie Qin, Jie Wu, Xuefeng Xiao, Lujun Li, and Xingang Wang. Activation modulation and recalibration scheme for weakly supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2117–2125, 2022.
- [89] Shenghai Rong, Bohai Tu, Zilei Wang, and Junjie Li. Boundary-enhanced co-training for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19574–19584, 2023.
- [90] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- [91] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [92] Simone Rossetti, Damiano Zappia, Marta Sanzari, Marco Schaerf, and Fiora Pirri. Max pooling with vision transformers reconciles class and shape in weakly supervised semantic segmentation. In *European Conference on Computer Vision*, pages 446–463. Springer, 2022.
- [93] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ” grabcut” interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.
- [94] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: end-to-end weakly-supervised semantic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16846–16855, 2022.
- [95] Lixiang Ru, Heliang Zheng, Yibing Zhan, and Bo Du. Token contrast for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2023.
- [96] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

- [97] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
- [99] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- [100] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [101] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.
- [102] W. Sun, Z. Liu, Y. Zhang, Y. Zhong, and N. Barnes. An alternative to wsss? an empirical study of the segment anything model on weakly-supervised semantic segmentation problems, 2023.
- [103] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1818–1827, 2018.
- [104] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *Proceedings of the IEEE international conference on computer vision*, pages 1769–1776, 2013.
- [105] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 507–522, 2018.
- [106] T. Tieleman. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, 2012.

- [107] Philip HS Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998.
- [108] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [109] Olga Veksler and Yuri Boykov. Sparse non-local crf. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4493–4503, 2022.
- [110] Changwei Wang, Rongtao Xu, Shibiao Xu, Weiliang Meng, and Xiaopeng Zhang. Treating pseudo-labels generation as image matting for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 755–765, 2023.
- [111] Yude Wang, Jie Zhang, Meina Kan, S. Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12272–12281, 2020.
- [112] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2314–2320, 2016.
- [113] Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th international conference on Machine learning*, pages 1168–1175, 2008.
- [114] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition, 2016.
- [115] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.
- [116] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: Cross language image matching for weakly supervised semantic segmentation. In *Proceedings of the*

- IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4483–4492, 2022.
- [117] Jia Xu, A. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3781–3790, 2015.
 - [118] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4310–4319, 2022.
 - [119] Xiaobo Yang and Xiaojin Gong. Foundation model assisted weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 523–532, 2024.
 - [120] Xinyu Yang, Hossein Rahmani, Sue Black, and Bryan M Williams. Weakly supervised co-training with swapping assignments for semantic segmentation. *arXiv preprint arXiv:2402.17891*, 2024.
 - [121] Zhiwei Yang, Kexue Fu, Minghong Duan, Linhao Qu, Shuo Wang, and Zhijian Song. Separate and conquer: Decoupling co-occurrence via decomposition and representation for weakly supervised semantic segmentation. *arXiv preprint arXiv:2402.18467*, 2024.
 - [122] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, and Lihe Zhang. Joint learning of saliency detection and weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7223–7233, 2019.
 - [123] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12765–12772, 2020.
 - [124] Xiangrong Zhang, Zelin Peng, Peng Zhu, Tianyang Zhang, Chen Li, Huiyu Zhou, and Licheng Jiao. Adaptive affinity loss and erroneous pseudo-label refinement for weakly supervised semantic segmentation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5463–5472, 2021.
 - [125] Zhongwen Zhang and Yuri Boykov. Discriminative entropy clustering and its relation to k-means and svm, 2023.

- [126] Xin Zheng, Qinyi Lei, Run Yao, Yifei Gong, and Qian Yin. Image segmentation based on adaptive k-means algorithm. *EURASIP Journal on Image and Video Processing*, 2018(1):1–10, 2018.
- [127] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.
- [128] Tianfei Zhou, Meijie Zhang, Fang Zhao, and Jianwu Li. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4299–4309, 2022.
- [129] Lianghui Zhu, Yingyue Li, Jieming Fang, Yan Liu, Hao Xin, Wenyu Liu, and Xing-gang Wang. Weaktr: Exploring plain vision transformer for weakly-supervised semantic segmentation. *arXiv preprint arXiv:2304.01184*, 2023.
- [130] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 18(9):884–900, 1996.