Soft Labels for Training and Evaluating Semantic Segmentation Models

by

Nima Jamali

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, 2026

© Nima Jamali 2026

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

Despite notable advances in network architectures and representation learning, most semantic segmentation pipelines continue to rely on hard ground-truth labels and evaluation metrics originally designed for binary masks. This assumption is misaligned with real-world data, where object boundaries are often ambiguous, annotations are noisy, spatial downsampling aggregates multiple semantic classes, and uncertainty is frequently encoded through void labels that are ignored during training and evaluation. As a result, both learning objectives and evaluation criteria do not faithfully reflect the underlying semantic structure and uncertainty of the data.

In this thesis, we study semantic segmentation in settings where the ground-truth labels are not necessarily hard, but instead may be uncertain. We refer to this formulation as *soft-label semantic segmentation*. We treat this problem in a unified, end-to-end manner encompassing label generation, training, and evaluation.

To generate soft labels, we propose a geometry-aware downsampling strategy called *Weighted Average Pooling* (*WAP*) for semantic segmentation masks. WAP produces smooth and probabilistically valid soft labels at arbitrary resolutions by constructing spatially varying weights based on geometric relationships and spatial proximity. As a result, the generated soft labels are resolution-agnostic, preserve the underlying probabilistic structure of the annotations, and avoid artifacts commonly introduced by conventional downsampling methods.

In addition, this work motivates the need for evaluation metrics that operate directly on probabilistic segmentation outputs. To this end, we introduce several principled relaxations of the soft intersection-over-union (soft IoU) metric that provide faithful extensions of standard IoU to soft-label settings. We further introduce void replacement strategies that assign soft class distributions to void pixels based on spatial context, enabling uncertain regions such as ambiguous boundaries and thin structures to be incorporated directly into the supervision signal.

Extensive experiments on the PASCAL VOC 2012 dataset demonstrate that WAP produces more faithful soft labels than conventional approaches, particularly in scenarios involving thin structures and complex spatial arrangements. The proposed soft IoU relaxations offer improved interpretability and better alignment with hard-label evaluation, while the void replacement strategies perform comparably to hard-label baselines, indicating that incorporating soft supervision in uncertain regions does not compromise segmentation quality. Together, these contributions establish a principled framework for generating, training, and evaluating soft labels in semantic segmentation.

Acknowledgements

I would like to sincerely thank my supervisors, Prof. Olga Veksler and Prof. Yuri Boykov, for their invaluable guidance and support throughout my Master's studies. Their expertise, dedication, and thoughtful feedback have profoundly shaped the direction of this thesis and my development as a researcher. Beyond their supervision, I am especially grateful for the many insightful discussions we shared, through which I learned tremendously and strengthened my understanding of research and critical thinking.

I am deeply appreciative of my thesis committee members, Prof. Jesse Hoey and Prof. Hongyang Zhang, for their thoughtful review and for generously dedicating their time to the evaluation of this thesis.

I extend my gratitude to Indy Ng and Lori Paniak from CSCF for their continuous assistance with computational resources and server support, which were essential to the experiments conducted in this thesis.

Special gratitude is owed to Matina for her unwavering support, encouragement, and constant presence throughout this journey, which played a meaningful role in sustaining me during this period. I also wish to acknowledge Jiahao and the other members of the Computer Vision Group and AI Lab for their insightful discussions and collaborative spirit, as well as my friends for their continued encouragement and companionship.

Finally, I offer my deepest thanks to my parents, Afsaneh and Peyman, for their unwavering belief in me, their countless sacrifices, and their unconditional love and support. All that I have achieved rests upon their strength and encouragement.

Dedication

To my family, for being my home across every distance.

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 Background & Related Work 6

2.1 Semantic Segmentation 6

2.1.1 Levels of Supervision in Semantic Segmentation 8

2.1.2 Deep Learning-Based Approaches 10

2.1.3 Foundation-Model-Based Approaches 12

2.2 Soft Labels and Label Uncertainty 14

2.3 Downsampling Approaches 15

2.3.1 Nearest-Neighbor Downsampling 15

2.3.2 Bilinear Downsampling . . . 16
2.3.3 Area-Based Downsampling . . . 16
2.3.4 Adaptive Average Pooling . . . 17
2.3.5 Summary . . . 17
2.4 Loss Functions for Semantic Segmentation . . . 18
2.4.1 Cross-Entropy (CE) Loss . . . 19
2.4.2 Kullback-Leibler (KL) Divergence Loss . . . 20
2.4.3 Collision Cross-Entropy (CCE) Loss . . . 21
2.4.4 Collision Divergence (CD) Loss . . . 22
2.5 Evaluation Metrics for Segmentation . . . 22
2.5.1 Pixel-Level Metrics . . . 22
2.5.2 Overlap-Based Metrics . . . 24
2.5.3 Soft Evaluation Metrics . . . 26

3 Proposed Downsampling Approach **28**
3.1 Problem Statement . . . 28
3.2 Method Overview . . . 29
3.3 Components of the Method . . . 29
3.3.1 Positional Encoding . . . 30
3.3.2 Upsampling . . . 31
3.3.3 Normalization . . . 31
3.3.4 Weighted Aggregation . . . 32
3.4 Advantages . . . 33
3.5 Summary . . . 34

4 Relaxations of Soft-mIoU for Evaluation **36**
4.1 Symmetric Difference Relaxation . . . 36
4.2 Square-Root Relaxation . . . 38
4.3 Second-Order (Squared) Relaxation . . . 39
4.4 Summary . . . 41

5 Void Pixels as Uncertain Supervision **42**
5.1 Distance-Based Void Replacement . . . 42
5.2 Gaussian Void Replacement . . . 46
5.3 Summary . . . 48

6 Experiments **50**
6.1 Dataset . . . 50
6.2 Implementation Details . . . 50
6.3 Handling of Void Pixels at Coarse Resolution . . . 51
6.4 Quantitative Results . . . 52
6.4.1 Effect of Downsampling Strategy . . . 52
6.4.2 Effect of the Loss Function under Soft Supervision . . . 53
6.4.3 Fine-Level Hard Supervision vs Coarse-Level Soft Supervision . . . 54
6.4.4 Comparison of Soft IoU Relaxations . . . 55
6.4.5 Comparison of Void Replacement Strategies . . . 57
6.5 Qualitative Results . . . 59
6.5.1 Qualitative Analysis of Downsampling Approaches . . . 59
6.5.2 Qualitative Comparison Across Training Strategies . . . 62

7 Conclusion and Future Work **67**

References **69**

List of Figures

1.1 Illustration of common computer vision tasks. 2

1.2 Illustrative examples of annotation errors in the PASCAL VOC dataset. . . 5

2.1 Segmentation outputs for a street scene. 7

2.2 Illustration of semantic segmentation under varying levels of supervision. . 9

2.3 Overview of FCN architecture. 10

2.4 Overview of ViT architecture. 12

2.5 Example from the PASCAL VOC dataset illustrating annotation ambiguity arising from thin structures and semantic uncertainty. 14

2.6 Comparison of standard downsampling methods. 18

2.7 Illustration of the relationship between predicted and ground-truth regions for a single class. 23

3.1 Overview of the Weighted Average Pooling (WAP) method for downsampling. 30

3.2 Example of bilinear upsampling and normalization of a positional encoding slice. 32

4.1 Illustration of the standard soft IoU formulation for a single pixel with two classes. 37

4.2 Visualization of the symmetric-difference soft IoU for a two-class, single-pixel setting. 38

4.3 Visualization of the square-root soft IoU for a two-class, single-pixel setting. 39

4.4 Visualization of the second-order soft IoU for a two-class, single-pixel setting. 40

5.1 Example from the PASCAL VOC dataset illustrating void pixels in the annotation mask. . . . 43
5.2 Illustration of the proposed distance-based void replacement strategy on PASCAL VOC examples. . . . 44
5.3 Magnified visualization of boundary behavior under distance-based void replacement. . . . 45
5.4 Overview of Gaussian void replacement pipeline. . . . 46
5.5 Illustration of the proposed Gaussian void replacement strategy on PASCAL VOC examples. . . . 48
5.6 Magnified visualization of boundary behavior under Gaussian void replacement. . . . 49

6.1 Qualitative comparison of downsampling approaches on PASCAL VOC ground-truth masks. . . . 60
6.2 Qualitative comparison of downsampling approaches on a synthetic two-stripe pattern. . . . 61
6.3 Qualitative comparison of downsampling approaches on a synthetic barcode pattern. . . . 61
6.4 Qualitative comparison of predictions obtained by models trained with different downsampling approaches for soft-label supervision. . . . 63
6.5 Qualitative comparison of predictions obtained using different loss functions for soft supervision with an MLP segmentation head. . . . 65
6.6 Qualitative comparison of predictions obtained by training at different resolution levels using linear and MLP segmentation heads. . . . 66

List of Tables

6.1 Comparison of different downsampling approaches under hard and soft supervision at coarse resolution, evaluated using a linear and an MLP segmentation head. 53

6.2 Effect of the training loss function under coarse-level soft supervision. . . . 54

6.3 Comparison of training with fine-level hard labels, coarse-level hard labels, and coarse-level soft labels. 55

6.4 Comparison of IoU relaxations for evaluating probabilistic segmentation outputs. 56

6.5 Quantitative comparison of void replacement approaches on the full PASCAL VOC training set using a linear segmentation head. 57

6.6 Quantitative comparison of void replacement approaches on the boundary-focused subset of PASCAL VOC, where large void regions are excluded. . . 59

Chapter 1

Introduction

The goal of semantic segmentation is to assign a semantic label to every pixel in an image. Unlike image-level classification or object detection, which operate on global or region-based representations, semantic segmentation requires dense, pixel-level understanding of visual scenes, as Figure 1.1 illustrates. It therefore forms a foundational component of the real-world systems that require precise, pixel-level spatial reasoning.

Over the past decade, semantic segmentation has undergone rapid development driven by advances in deep learning. Early approaches based on hand-crafted features and graphical models [26, 5, 4] have largely been replaced by convolutional neural networks (CNNs) [35, 44, 8] and, more recently, transformer-based architectures [63, 10] capable of learning hierarchical representations directly from the input images. Encoder–decoder designs [44, 2, 8], multi-scale feature aggregation [62], attention mechanisms [63, 48], and large-scale pretraining [24, 40] have significantly contributed to improvements in both accuracy and generalization. These developments have led to segmentation models that perform reliably across diverse visual domains and increasingly complex scenes.

Despite this progress, several fundamental assumptions underlying modern segmentation pipelines remain less examined. One such assumption is that ground-truth annotations are inherently *hard*, meaning that each pixel is assigned exclusively to a single semantic class. While convenient, this assumption does not reflect the true nature of visual data. In practice, however, object boundaries in natural images rarely align perfectly with pixel grids. Edge pixels may contain parts of two or even more classes, while hard annotation requires a single class assignment. Annotation noise, human error, and inter-annotator variability also commonly occur, especially for thin objects or low-contrast regions where boundaries are visually unclear, as depicted in Figure 1.2. Forcing such inherently uncer-

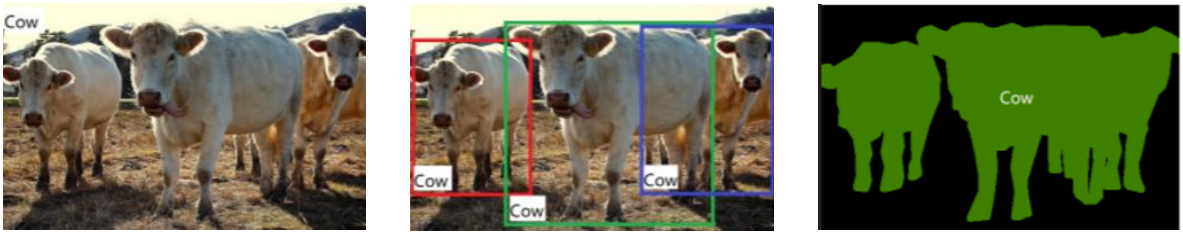


Figure 1.1: Illustration of common computer vision tasks. Image classification assigns a single label to the entire image. Object detection identifies object instances using bounding boxes. Semantic segmentation produces dense, pixel-level class predictions for the image (Adapted from [57]).

tain regions into hard class assignments introduces artificial discontinuities and may bias both training and evaluation.

These limitations motivate the use of *soft* labels in semantic segmentation. Soft labels explicitly represent uncertainty and mixed-class membership by assigning each pixel a probability distribution over semantic classes. They arise naturally in several vision tasks, due to the aforementioned reasons. Another source of soft labels emerges from rescaling operations in modern segmentation pipelines. Feature extractors such as CNNs or vision transformers (ViTs) operate at spatial resolutions much lower than the input image. When high-resolution ground-truth masks are downsampled to these coarser resolutions, multiple classes may be aggregated into a single pixel. Using soft labels is thus a mathematically consistent way to represent the resulting mixed-class targets. Incorporating soft labels into segmentation pipelines therefore offers both conceptual and practical advantages.

However, adopting soft labels introduces new challenges that are not adequately addressed by existing methods. The first challenge concerns how soft labels should be generated, especially when they originate from downsampling operations. Standard downsampling techniques were initially proposed for natural images or feature tensors and, when applied to semantic masks, reveal several shortcomings. Nearest-neighbor interpolation preserves sharp edges but suffers from aliasing and block artifacts. Bilinear interpolation produces smooth outputs but aggregates information from only a small, fixed neighborhood, regardless of the scaling factor. Area-based method and adaptive pooling perform regional averaging that accounts for the scaling factor, but rely on hard rectangular bins and uniform weighting, ignoring spatial proximity and geometry.

Consequently, existing downsampling approaches often fail to produce soft labels that

are both spatially coherent and probabilistically meaningful. Such methods either oversimplify the aggregation process or impose discontinuities that distort the semantic structure. These issues are particularly problematic when soft labels are used as training targets, where small inconsistencies can propagate through the optimization process and negatively affect learning.

A second, closely related challenge occurs at the evaluation stage. Semantic segmentation models are typically evaluated using overlap-based metrics such as intersection-over-union (IoU) or Dice coefficient. These metrics were originally designed for hard (binary) masks and are based on set-theoretic notions of intersection and union. Although soft extensions of these metrics have been proposed, the most widely used formulation of soft IoU suffers from a fundamental inconsistency. It fails to achieve its maximum value when the predicted distribution exactly matches the ground-truth distribution. This behavior contradicts a basic requirement of a similarity measure and results in misleading evaluations when soft labels are involved.

Another related challenge concerns how annotation uncertainty encoded through *void* pixels (regions where annotators could not assign a confident or unique semantic class) should be handled during training, as simply masking these regions out prevents the model from exploiting spatial context in such pixels.

The central premise of this thesis is that soft-label semantic segmentation should be approached as a unified, end-to-end problem that spans label representation, downsampling strategies, training objectives, and evaluation metrics. These components are tightly coupled. The soft-label generation process directly influences the suitability of the loss functions used during training, while the choice of evaluation metrics determines how faithfully model performance is assessed. Treating any of these components in isolation increases the risk of introducing inconsistencies that can obscure the true behavior of segmentation models.

To this end, this thesis makes three complementary contributions:

1. We propose **a geometry-aware downsampling framework for semantic segmentation masks** that produces smooth, resolution-agnostic soft labels while preserving their probabilistic nature. Rather than relying on fixed neighborhoods or hard spatial regions, the proposed method constructs spatially varying weights that encode the contribution of each input pixel to the computation of each output spatial location.

2. We introduce **consistent relaxations of the soft IoU metric** that satisfy a fundamental consistency property, namely that the metric attains its maximal value when

the predicted distribution perfectly matches the ground-truth distribution. These relaxations preserve the behavior of the standard IoU in the hard-label setting while providing meaningful and well-behaved evaluations for probabilistic segmentation outputs.

3. We develop **void replacement strategies for semantic segmentation** that replace void labels with soft class distributions derived from spatial context. These strategies enable uncertain regions, such as ambiguous boundaries and thin structures, to be incorporated into the supervision signal rather than discarded, while remaining compatible with standard training pipelines.

Together, these contributions establish a principled framework for working with soft labels in semantic segmentation. The proposed downsampling approach ensures that soft labels are generated in a geometrically meaningful manner, while the proposed relaxations of soft IoU enable these labels to be evaluated consistently and fairly. Empirical experiments on the PASCAL VOC 2012 dataset demonstrate the practical relevance of these ideas in comparison with conventional approaches.

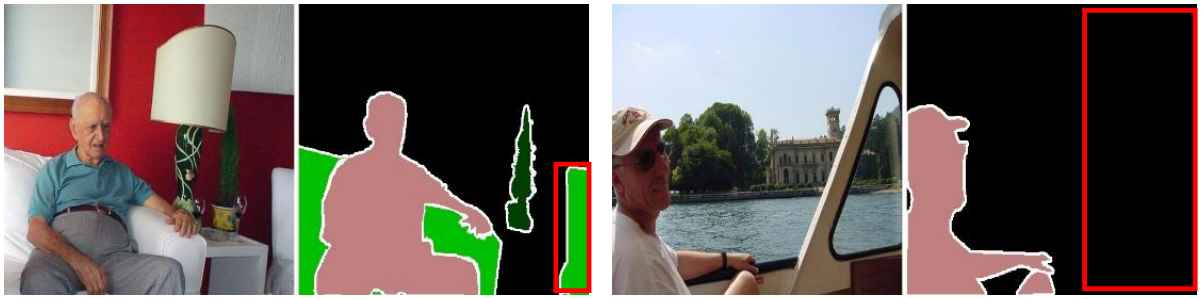
The remainder of this thesis is organized as follows. Chapter 2 reviews background material and related work on semantic segmentation, soft labels, downsampling methods, loss functions, and evaluation metrics. Chapter 3 presents the proposed downsampling approach in detail. Chapter 4 provides the proposed relaxations of soft-mIoU and analyzes their theoretical properties. Chapter 5 presents strategies for handling void annotations through soft-label replacement. Chapter 6 reports the experimental results, and Chapter 7 concludes with a discussion of directions for future work.



(a) Human Annotation Errors



(b) Inter-annotator Inconsistency



(c) Over-confidence in Labeling

Figure 1.2: Illustrative examples of annotation errors in the PASCAL VOC dataset. Regions associated with annotation uncertainty are outlined with red bounding boxes. (a) Human annotation errors, where clearly visible foreground class instances are labeled as background or void. (b) Inter-annotator inconsistency, showing inconsistent labeling practices across annotators for similar objects. (c) Over-confidence in labeling, where ambiguous or partially visible objects are assigned hard class labels despite uncertain class membership.

Chapter 2

Background & Related Work

2.1 Semantic Segmentation

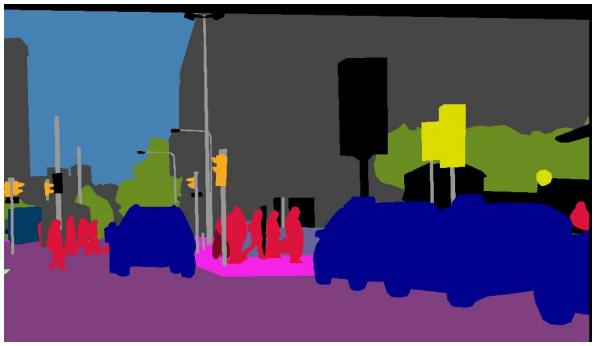
Semantic segmentation is a fundamental task in computer vision that focuses on understanding images at the pixel level. The objective is to assign each pixel a semantic label such as "person", "car", or "background", producing a dense map that captures the structure and composition of the scene. Consequently, semantic segmentation can be formulated as a per-pixel classification problem that aims to learn a mapping $f : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{H\times W\times C}$, where H and W denote the spatial resolution of the image and C represents the number of semantic classes. The output is typically a probability map over classes for each pixel.

Semantic segmentation differs from related tasks such as instance segmentation and panoptic segmentation [23], as illustrated in Figure 2.1. Instance segmentation focuses on distinguishing between individual occurrences of the same class, while semantic segmentation groups all instances of a class into a single category. Panoptic segmentation combines both perspectives by predicting semantic labels and separating individual instances at the same time.

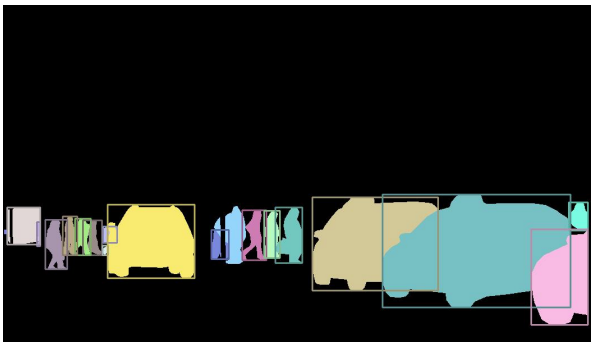
Semantic segmentation has become a critical component of many real-world applications that require a detailed understanding of visual scenes. In autonomous driving, a safety-critical domain, it is used to identify pedestrians, vehicles, roads, and traffic signs to support navigation and decision-making [6, 41, 19, 28]. Its applications also extend to medical imaging, where semantic segmentation is employed to delineate tumors, organs, and other anatomical structures from MRI, CT, or histopathology scans [44, 16, 34, 56]. Similarly, semantic segmentation contributes to robotics and augmented reality, where



(a) Input Image



(b) Semantic Segmentation



(c) Instance Segmentation



(d) Panoptic Segmentation

Figure 2.1: Segmentation outputs for a street scene [23]. The input image is shown alongside its semantic segmentation map, instance segmentation prediction, and corresponding panoptic segmentation, which unifies both semantic and instance-level information.

pixel-level perception enables localization and scene reconstruction [37, 59]. Beyond these fields, semantic segmentation is also used in remote sensing and aerial imagery, agriculture, environmental monitoring, microscopic analysis, facial segmentation, and image editing [58, 68, 27, 21, 45, 32].

Although semantic segmentation may appear trivial to human perception, it remains an inherently challenging problem for machines. This difficulty has motivated extensive research in the field over the past several decades. Early approaches relied on hand-crafted features and traditional computer vision techniques [5, 33, 26], while most modern methods presented in the last decade leverage deep neural networks capable of learning strong feature representations directly from data and generalizing to unseen scenes [35, 44, 8].

More recently, the emergence of large-scale foundation models has further advanced the field by providing transferable high-quality visual representations that can be effectively adapted for semantic segmentation tasks [66, 38, 40, 24, 53].

The level of supervision available during the training phase also plays a major role in how segmentation models are designed and evaluated. Depending on the annotation quality, methods can be fully supervised, weakly supervised, semi-supervised, or even unsupervised.

The following sections review these aspects in more detail. Subsection 2.1.1 discusses the different levels of supervision used in semantic segmentation, Subsection 2.1.2 revisits the main deep learning–based approaches, and Subsection 2.1.3 summarizes recent methods built upon large-scale foundation models.

2.1.1 Levels of Supervision in Semantic Segmentation

Semantic segmentation methods can be categorized according to the level and type of supervision available during training. The achievable accuracy and the cost of dataset preparation is determined based on the form and precision of the supervision.

Fully supervised methods rely on dense, pixel-level annotations in which each pixel is assigned a corresponding semantic class. This setup generally achieves the highest accuracy but comes with substantial costs. Creating and labeling large-scale datasets such as PASCAL VOC [14], ADE20K [65] or MS COCO [30] requires significant human effort to precisely delineate object boundaries at pixel-level.

Weakly supervised approaches reduce annotation costs by leveraging sparse or indirect forms of supervision, such as bounding boxes [11], scribbles [29], point annotations [3], or image-level tags [25, 1, 31]. Instead of relying on dense annotations, these methods typically generate pixel-level masks from sparse supervision using Class Activation Maps (CAMs) [64], pixel affinities, or saliency information.

Semi-supervised segmentation uses a small set of annotated data together with a larger set of unlabeled images. Typical strategies in this realm include pseudo-labeling and teacher–student frameworks that propagate supervision to unlabeled samples [69, 22, 50].

Finally, unsupervised and self-supervised methods aim to learn pixel-wise semantics without relying on any kinds of human annotation. While unsupervised approaches often employ clustering or feature-matching methods to group visually similar regions, self-supervised methods leverage contrastive learning to obtain transferable visual representa-



Figure 2.2: Illustration of semantic segmentation under varying levels of supervision (Adapted from [46]).

tions [38, 7]. These learned representations can subsequently be fine-tuned for downstream semantic segmentation tasks.

Figure 2.2 compares the training data available under different levels of supervision. Overall, as the level of supervision decreases, the segmentation problem becomes more ambiguous and the model's uncertainty increases, while the cost and effort of data annotation are substantially reduced.

Although different levels of supervision have been explored in this subsection, the work presented in this thesis mainly focuses on fully supervised semantic segmentation. Accordingly, the following subsections review prominent deep learning-based approaches developed under full supervision and discuss their evolution toward more generalizable segmentation models.

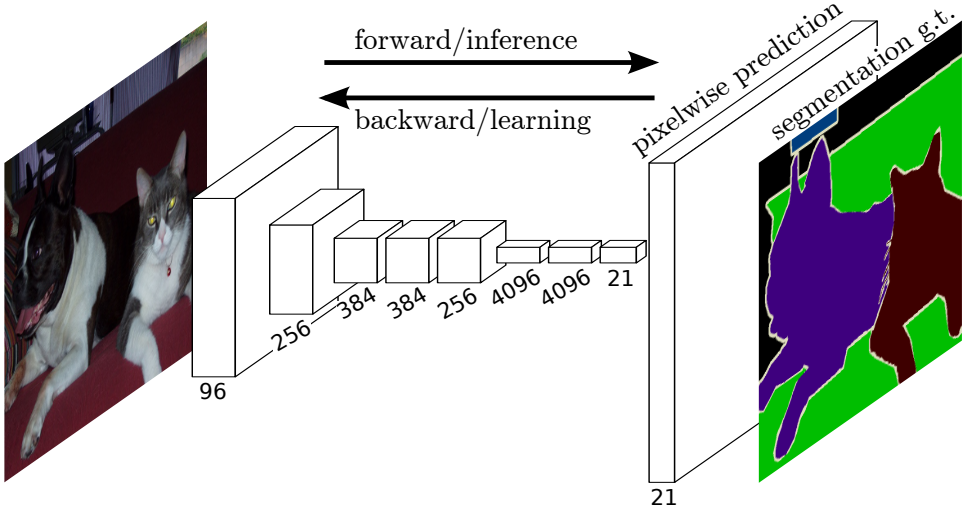


Figure 2.3: Overview of FCN architecture [35]. FCN transforms a conventional classification CNN into a fully convolutional structure by replacing fully connected layers with convolutional ones.

2.1.2 Deep Learning-Based Approaches

In recent years, with the rapid development of neural networks, deep learning-based semantic segmentation methods have evolved into a more advanced stage of research. The evolution of Convolutional Neural Networks (CNNs) and transformer architectures has made them the mainstream choices for modern computer vision applications.

Fully Convolutional Networks (FCN) [35] was a pioneering work in this area, proposing the idea of replacing fully connected layers with convolutional ones to extend classification CNNs to per-pixel predictions tasks. The architecture of FCN is depicted in Figure 2.3.

Since the introduction of FCN, subsequent research has proposed encoder–decoder backbones, which have become a dominant design paradigm in semantic segmentation. These networks typically consist of two complementary components: an encoder that captures high-level semantic features by downsampling the input image, and a decoder that incrementally upsamples these representations to recover original spatial resolution and generate pixel-level predictions. The encoder is generally composed of convolutional and pooling layers that capture contextual information, whereas the decoder utilizes upsampling, deconvolution, or interpolation operations, often combined with skip connections, to reconstruct spatial details lost during encoding.

Among encoder–decoder-based networks, SegNet [2] introduced an efficient decoding

mechanism that reuses the encoder's pooling indices for upsampling, making it well-suited for real-time applications. Originally developed for biomedical image segmentation, U-Net [44] popularized the usage of skip connections between corresponding encoder and decoder layers to preserve spatial information, becoming one of the most influential architectures in the field. Pyramid Scene Parsing Network (PSPNet) [62] employs a Pyramid Pooling Module (PPM) that aggregates contextual features from multiple spatial scales using pooling with different kernel sizes, and then concatenates them into a unified multi-scale representation that captures both local and global information (extracted from an encoder-decoder module). The DeepLab family [8, 9] introduced atrous convolutions to enlarge the receptive field without reducing spatial resolution. The latest variant, DeepLab v3+ [9], integrates an encoder–decoder design to refine object boundaries and improve segmentation accuracy.

The encoder component in most segmentation networks commonly relies on a feature extraction backbone adapted from image classification architectures pre-trained on large-scale datasets such as ImageNet [12]. Early models like VGG [47] and ResNet [17] served as standard backbones, providing strong hierarchical representations for downstream segmentation tasks. In addition, lightweight architectures including MobileNet [20] and EfficientNet [52] have been adopted to enable real-time segmentation in resource-limited settings, striking a trade-off between accuracy and computational efficiency.

While convolutional networks dominated semantic segmentation for several years, the emergence of transformer architectures has marked a major shift in visual representation learning. Originally designed for sequence modeling in natural language processing, transformers rely on a self-attention mechanism [54] that allows each element in an input sequence to attend to all others, effectively modeling long-range contextual dependencies. Inspired by this idea, Vision Transformer (ViT) [13] was the first to successfully apply a purely transformer-based architecture to image analysis by treating an image as a sequence of non-overlapping fixed-size patches (e.g., 16×16), analogous to word tokens in text. Figure 2.4 illustrates the overview of such a transformer model. When trained on large-scale datasets, ViT demonstrated that attention-based architectures can achieve performance on par with or exceeding their convolutional counterparts, leading to the development of transformer-based approaches for dense prediction tasks.

SETR (Segmentation Transformer) [63] was one of the earliest attempts to adapt ViT for semantic segmentation. It directly applied a ViT backbone to encode contextual relationships across the entire image, but its high computational cost and quadratic complexity motivated further architectural refinements. Segmenter [48] formulated segmentation as a set prediction problem and improved efficiency by introducing a mask transformer decoder that projects patch representations into class-specific masks, while Mask2Former [10] unified semantic, instance, and panoptic segmentation under a common mask-based frame-

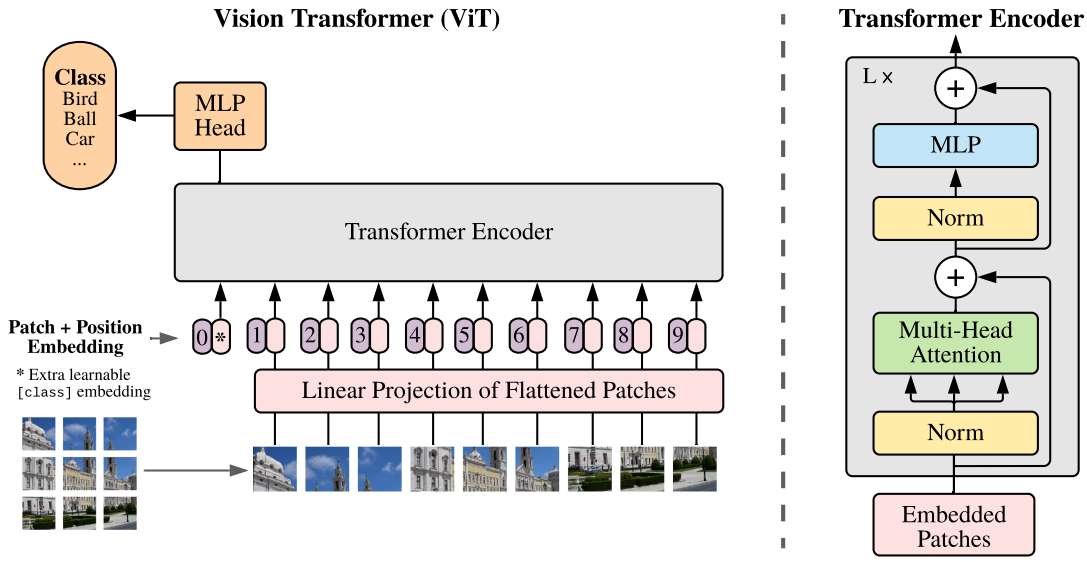


Figure 2.4: Overview of ViT architecture [13]. The input image is divided into fixed-size patches that are flattened and linearly projected into embeddings augmented with positional encodings and a learnable class token. These embeddings are then processed by a stack of standard transformer encoder blocks consisting of multi-head self-attention and feed-forward layers. The final class token output is fed into a classification head to produce the prediction.

work using multi-scale attention. These advances established transformers as competitive replacements for convolutional networks, offering strong context modeling and scalability across various segmentation settings.

In summary, the evolution from CNN-based to transformer-based architectures has reshaped semantic segmentation by shifting from localized feature extraction to global, attention-driven understanding. Modern models increasingly integrate the advantages of both paradigms, using convolutional layers for spatial encoding and transformers for capturing long-range dependencies.

2.1.3 Foundation-Model-Based Approaches

Recent advances with large-scale foundation models have reshaped the semantic segmentation landscape by introducing extensive pretraining on diverse visual data. Unlike conventional networks trained from scratch or fine-tuned on a single dataset, foundation models

such as CLIP [42], DINO/DINOv2 [7, 40], and Segment Anything Model (SAM) [24] learn highly transferable representations through self-supervised or vision-language alignment objectives. These models generalize effectively across datasets and tasks with minimal fine-tuning, which makes them well-suited for robust zero-shot or few-shot segmentation performance.

CLIP [42] applies contrastive learning between images and their paired natural language descriptions to learn a joint visual–text embedding space. It consists of a Vision Transformer (ViT) as the vision encoder and a transformer-based text encoder, which are jointly trained to bring matching image–text pairs closer in the embedding space while pushing non-matching pairs apart. For semantic segmentation, CLIP's pretrained vision encoder provides powerful and transferable features that can be fine-tuned under full supervision. MaskCLIP [66] and CLIPSeg [36] are among approaches that use CLIP's cross-modal alignment to enhance pixel-level understanding, either by fine-tuning the vision encoder on segmentation masks or conditioning predictions on text-based class prompts.

DINO [7] and DINOv2 [40] are self-supervised vision transformers trained without labels through a teacher–student distillation framework. The student network in this setup learns to reproduce the teacher's embeddings for different augmented views of the same image, which encourages invariance to geometric and photometric transformations. For fully supervised semantic segmentation, the features learned by DINO and DINOv2 serve as strong pretrained backbones, which often outperform those of purely supervised CNNs once fine-tuned on labeled datasets.

The Segment Anything Model (SAM) [24] is a promptable segmentation model trained on a large-scale dataset that contains over a billion masks. SAM uses a ViT-based image encoder to extract features, a flexible prompt encoder to process input cues (points, bounding boxes, masks, or text), and a mask decoder that predicts segmentation masks conditioned on both image and prompt embeddings. Although SAM was originally designed for zero-shot and interactive segmentation, it can also be adapted to fully supervised settings by fine-tuning its encoder or mask decoder on specific datasets.

Although these foundation models differ in their objectives, they all demonstrate strong generalization capabilities with minimal task-specific supervision. Among these approaches, DINOv2 [40] stands out for its stability, scalability, and high-quality dense features, which consistently translate into strong performance when fine-tuned for pixel-level prediction. For this reason, the work presented in this thesis relies on DINOv2 as the primary feature representation.



(a) Input image



(b) Ground-truth mask

Figure 2.5: Example from the PASCAL VOC dataset illustrating annotation ambiguity arising from thin structures and semantic uncertainty. Thin components of the bicycle are difficult to align with the pixel grid and are often only partially represented in the ground-truth mask. Additionally, semantic ambiguity is visible for small or distant individuals, which may be labeled as *void*.

2.2 Soft Labels and Label Uncertainty

Soft labels represent a probability distribution over possible classes, instead of assigning each pixel to a single category with absolute certainty. Unlike hard one-hot labels, which force the annotation of every pixel into a single class even when the visual evidence is ambiguous, soft labels make uncertainty explicit and provide a more realistic description of the underlying scene. Soft labels have been widely explored in machine learning, including early work on label smoothing [51, 39] and later in soft-target learning frameworks such as knowledge distillation [18, 67]. These advances demonstrate that replacing hard labels with soft ones can reduce overconfidence and improve generalization.

Soft labels arise naturally in many computer vision tasks due to several sources of ambiguity, such as unclear object boundaries, human annotation noise, and rescaling operations, as illustrated in Figure 2.5. As an example of how soft labels are used in semantic segmentation, [61] proposes a weakly supervised framework that treats segmentation as a relaxed Potts energy problem, allowing pixel labels to take soft, continuous values rather than discrete ones. The method involves a soft pseudo-labeling strategy that iteratively updates pixel-level class distributions by combining image evidence with spatial consis-

tency. Through a normalized-quadratic relaxation of the Potts model, the optimization operates directly on soft labels, enabling the representation of uncertainty in regions with weak supervision or ambiguous boundaries. This work illustrates how soft labels can be incorporated into the core mathematical structure of an energy-based segmentation formulation.

[60] further extends the role of soft labels by proposing a principled loss function, Collision Cross-Entropy (CCE), designed specifically for probabilistic label assignments. Instead of forcing predictions toward hard one-hot vectors, CCE encourages alignment between two probability distributions, making it ideal for clustering and segmentation scenarios where labels evolve gradually. This objective (discussed in detail in Subsection 2.4.3) generalizes classical cross-entropy to cases in which both predictions and targets are soft, providing a mathematically consistent loss function for soft-label segmentation.

As the following sections discuss, this shift toward soft ground truth also necessitates loss functions and evaluation metrics that operate directly on distributions, ensuring that both training and evaluation remain consistent with the underlying soft-label paradigm.

2.3 Downsampling Approaches

Downsampling is a fundamental operation in image processing and deep learning, used to reduce spatial resolution while preserving as much relevant context as possible. In semantic segmentation, downsampling is required both for feature extraction and for generating soft labels at lower resolutions. This section reviews the most commonly used downsampling methods in the literature: nearest-neighbor, bilinear interpolation, area-based averaging, and adaptive average pooling, and discusses their limitations when applied to binary masks or probability maps.

2.3.1 Nearest-Neighbor Downsampling

Nearest-neighbor interpolation assigns each output pixel the value of the closest input pixel. Although it is fast and preserves sharp edges, it introduces aliasing and blocky artifacts. As Figure 2.6 illustrates, another limitation is that it uses strictly local sampling, as it copies the value from a single input location and ignores all neighboring pixels. Consequently, it cannot produce a meaningful soft distribution when the input mask is hard.

2.3.2 Bilinear Downsampling

Bilinear interpolation computes each output pixel as a weighted combination of the four nearest input pixels:

$$\mathcal{B}(Y)_i = \sum_{j \in \mathcal{N}(i)} w_{ij}\, Y_j, \tag{2.1}$$

where Y represents the input soft mask, $\mathcal{N}(i)$ denotes the 2×2 neighborhood surrounding the continuous sampling location corresponding to output pixel i, and the weights w_{ij} are bilinear functions of the relative distances.

While bilinear interpolation produces smooth results for natural images, it has notable limitations as a downsampling method. It does not aggregate information over the full spatial region that an output pixel represents. Regardless of the downsampling factor, it always considers at most a 2×2 neighborhood, as depicted in Figure 1. Moreover, it assumes that values vary smoothly in Euclidean space, an assumption that does not hold for semantic labels or soft probability maps.

2.3.3 Area-Based Downsampling

Area-based downsampling computes each output pixel as the average of all input pixels that fall within the spatial region it represents. When the scaling factor is an integer, all pixels within the corresponding region contribute equally. For fractional scaling factors, contributions are weighted proportionally to the area of overlap between the input pixel grid and the spatial region associated with the output pixel.

Formally, letting $\mathcal{R}_i$ denote the set of input coordinates that map to output pixel i, area downsampling function $\mathcal{A}$ is defined as

$$\mathcal{A}(Y)_i = \frac{1}{|\mathcal{R}_i|} \sum_{j \in \mathcal{R}_i} Y_j, \tag{2.2}$$

where $|R_j|$ is the effective number of contributing input pixels for the lower-resolution pixel i.

Figure 1 shows that area-based downsampling performs true averaging over spatial regions and therefore offers better anti-aliasing than nearest-neighbor or bilinear interpolation; however, it remains limited when applied to binary masks or soft probability maps.

As it uniformly averages over rectangular regions, it can oversmooth semantic boundaries and distort soft labels, especially under fractional scaling ratios. The uniform weighting also ignores spatial distances between pixels, treating all locations within the region as equally relevant.

2.3.4 Adaptive Average Pooling

Adaptive Average Pooling (AAP) partitions the input into a fixed number of spatial bins and computes the average value within each bin. Unlike standard average pooling, which works with fixed kernel sizes and strides and therefore only supports integer downsampling factors, AAP accommodates arbitrary scaling ratios by selecting bin boundaries automatically so that the output reaches the desired resolution. Each output pixel is therefore the mean of all input pixels assigned to its corresponding bin, as expressed by

$$\mathcal{P}(Y)_i = \frac{1}{|B_i|} \sum_{j \in B_i} Y_j. \tag{2.3}$$

However, this approach shares many of the same limitations as area-based averaging. The hard partitioning of the input into rectangular bins can introduce discontinuities at bin boundaries, and the uniform weighting applied to all pixels within a bin ignores spatial proximity. Furthermore, when the scaling factor is non-integer, the bins can vary in size, leading to uneven aggregation across the image.

2.3.5 Summary

The discussed downsampling approaches differ substantially in how they aggregate spatial information. Nearest-neighbor and bilinear interpolation rely on local sampling and therefore do not integrate information over the full region represented by an output pixel. Area-based downsampling and adaptive average pooling perform regional averaging but impose hard rectangular partitions and apply uniform weights inside each region.

Figure 2.6 illustrates these differences. For a single output location (right), the contributing input pixels under each method (left) show distinct spatial patterns. These variations highlight the limitations of existing approaches and motivate the need for a geometry-aware downsampling method that outputs smooth, spatially consistent soft labels.

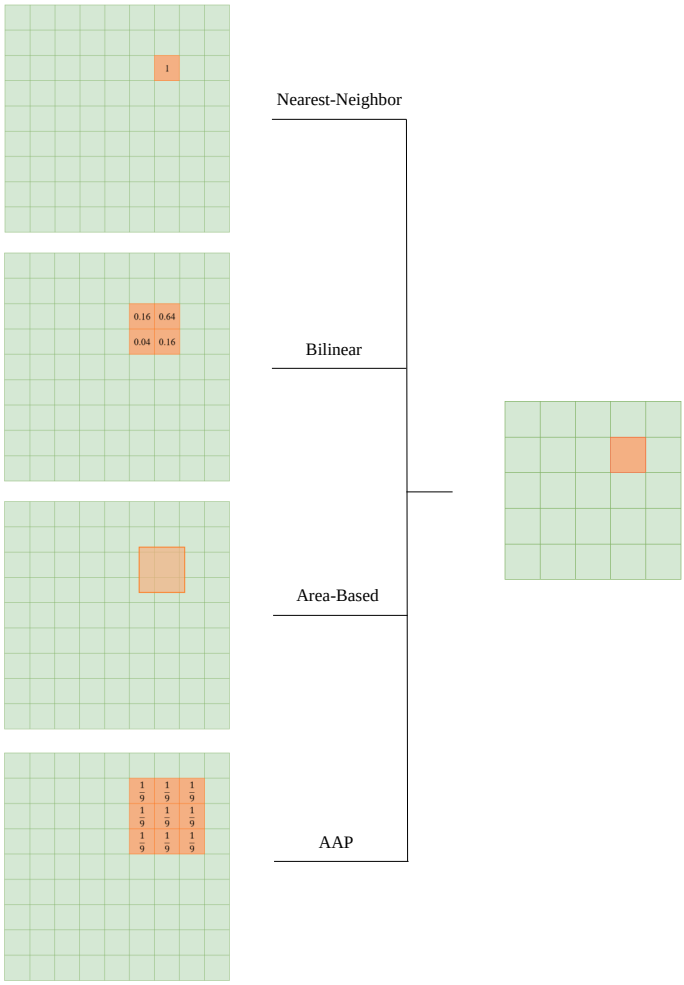


Figure 2.6: Comparison of standard downsampling methods. For a single output location (right), each method aggregates a different set of input pixels (left): nearest-neighbor samples one pixel, bilinear mixes at most four pixels, area-based downsampling averages all pixels in the geometric region, and adaptive average pooling averages all pixels within its assigned bin.

2.4 Loss Functions for Semantic Segmentation

Loss functions serve as key elements in training segmentation networks by guiding the optimization process toward correct pixel-wise predictions. Since semantic segmentation

requires assigning a semantic label to each pixel, an effective loss must penalize discrepancies between dense prediction maps and ground-truth annotations while remaining robust to class imbalance, soft labels, and ambiguity near object boundaries.

To establish a unified notation, we denote the ground-truth target distribution per pixel over the C semantic classes by

$$q = (q_1, q_2, \ldots, q_C) \in \Delta^{C-1}, \tag{2.4}$$

where q_c represents the probability that the given pixel belongs to class c. Hard labels are a special case of this formulation: if the correct class at a given pixel is c^*, then $q = e_{c^*}$, where e_{c^*} is the one-hot vector with a 1 in the c^*-th position and zeros elsewhere.

Similarly, the network predicts a probability distribution

$$p = (p_1, p_2, \ldots, p_C) \in \Delta^{C-1}, \tag{2.5}$$

obtained by applying the softmax function to the network's output (logits) $f \in \mathbb{R}^C$:

$$p_c = \frac{\exp(f_c)}{\sum_{k=1}^{C} \exp(f_k)}, \qquad c \in \{1, 2, \ldots, C\}. \tag{2.6}$$

With this unified representation of both hard and soft labels and predictions, the following subsections review the four families of loss functions considered in the experimental part of this thesis.

2.4.1 Cross-Entropy (CE) Loss

Shannon's Cross-Entropy (CE) loss is the standard objective for fully supervised semantic segmentation and remains the most widely adopted choice in modern dense prediction pipelines. Given a ground-truth distribution q and a predicted per-pixel distribution q, the per-pixel CE loss is defined as

$$\mathcal{L}_{\mathrm{CE}}(q, p) = -\sum_{c=1}^{C} q_c \log p_c. \tag{2.7}$$

In the case of hard labels where $q = e_{c^*}$, this expression simplifies to the negative log-likelihood of the correct class,

$$\mathcal{L}_{\mathrm{CE}}(q, p) = -\log p_{c^*}, \tag{2.8}$$

as all other entries of q are zero.

Although CE is considered highly effective for training with hard labels and provides stable gradients when combined with softmax operation, it becomes less suitable in settings where supervision is provided through soft labels. The inherent asymmetry of CE hinders it from measuring the overall similarity between two distributions q and p, which can make the loss overly harsh or misaligned when the targets encode uncertainty

2.4.2 Kullback-Leibler (KL) Divergence Loss

Kullback–Leibler (KL) divergence is a fundamental measure from information theory that quantifies how one probability distribution diverges from another. The forward KL divergence loss at a pixel is defined as

$$\mathcal{L}_{\mathrm{KL}}(q, p) = D_{\mathrm{KL}}(q \,\|\, p) = \sum_{c=1}^{C} q_c \log \frac{q_c}{p_c}. \tag{2.9}$$

This expression can be decomposed as

$$\mathcal{L}_{\mathrm{KL}}(q, p) = \underbrace{\sum_{c=1}^{C} q_c \log q_c}_{-H(q)} \underbrace{- \sum_{c=1}^{C} q_c \log p_c}_{\mathcal{L}_{\mathrm{CE}}(q,p)}, \tag{2.10}$$

where $H(q)$ is the entropy of target distribution, which does not depend on the model parameters and therefore remains constant during training. As a result, minimizing the forward KL divergence is equivalent to minimizing CE loss:

$$\arg\min_{p} \mathcal{L}_{\mathrm{KL}}(q, p) \equiv \arg\min_{p} \mathcal{L}_{\mathrm{CE}}(q, p). \tag{2.11}$$

However, KL divergence is asymmetric; in general,

$$D_{\mathrm{KL}}(q \,\|\, p) \neq D_{\mathrm{KL}}(p \,\|\, q). \tag{2.12}$$

The reverse KL divergence loss is given by

$$\mathcal{L}_{\mathrm{RKL}}(q,p) = D_{\mathrm{KL}}(p \,\|\, q) = \sum_{c=1}^{C} p_c \log \frac{p_c}{q_c}. \tag{2.13}$$

Although the RKL loss is typically not preferred for semantic segmentation due to its numerical instability when $q \to 0$, it is included in the experiments as a comparison baseline.

2.4.3 Collision Cross-Entropy (CCE) Loss

Collision Cross-Entropy (CCE) is based on the classical notion of collision probability, defined as the inner product of two distributions q and p. The collision probability corresponds to the probability that two random variables drawn from distributions q and p being equal. A collision event is not guaranteed even when the distributions are identical ($p = q$). The collision probability equals 1 only if both q and p are the same one-hot vectors. However, for uniform distributions, the collision probability reduces to $\frac{1}{C}$.

Collision Cross-Entropy (CCE) [60] applies a negative logarithm to the collision probability, yielding the loss

$$\mathcal{L}_{\mathrm{CCE}}(q,p) = -\log \sum_{c=1}^{C} q_c p_c \,. \tag{2.14}$$

CCE encourages high overlap between the predicted and target distributions. When q and p assign the probability mass to the same classes, their inner product increases, thus reducing the loss. In contrast, disagreement between the distributions reduces the collision probability and results in a larger penalty.

Expanding the CCE loss, it can be equivalently written as

$$\begin{aligned}
\mathcal{L}_{\mathrm{CCE}}(q,p) &= -\log(q^\top p) \\
&\equiv -\log \|q\| - \log \|p\| - \log\cos(q,p) \\
&\equiv \underbrace{-\log \sqrt{\sum_{c=1}^{C} q_c^2}}_{\frac{1}{2}H_2(q)} \underbrace{-\log \sqrt{\sum_{c=1}^{C} p_c^2}}_{\frac{1}{2}H_2(p)} - \log\cos(q,p) \\
&\equiv \frac{H_2(q) + H_2(p)}{2} - \log\cos(q,p),
\end{aligned} \tag{2.15}$$

where $H_2(q) = -\log \sum_{c=1}^{C} q_c^2$ is the second-order Rényi entropy term, and $\cos(q, p)$ is the cosine of the angle between q and p in $\mathbb{R}^C$. The logarithmic-cosine term—also referred to as Collision Divergence, as discussed in Subsection 2.4.4—can be interpreted as a distance between the two distributions when they are viewed as vectors in the unit sphere in $\mathbb{R}^C$.

Because CCE is only based on the dot product $q^\top p$, it treats the two distributions in a symmetric manner. This property is particularly advantageous for soft labels, where no single class can be regarded as the unique ground-truth label.

2.4.4 Collision Divergence (CD) Loss

Collision Divergence (CD) generalizes the idea of collision-based similarity by normalizing the raw collision probability $q^\top p$ using the magnitudes of the individual distributions. This produces a cosine-style similarity between the two distributions. CD loss is defined as the negative logarithm of their cosine similarity:

$$\mathcal{L}_{\mathrm{CD}}(q, p) = -\log \cos(q, p) = -\log \frac{q^\top p}{\|q\| \, \|p\|}. \tag{2.16}$$

CD loss is symmetric and equals zero only when the two distributions are identical.

2.5 Evaluation Metrics for Segmentation

Semantic segmentation models are typically evaluated by comparing predicted masks against ground-truth annotations at the pixel level. Over time, various evaluation metrics have been proposed, differing in whether they operate at the pixel level or capture region-level overlap. This section reviews the most commonly used metrics in the literature, highlights their assumptions, and and motivates the need for soft-label evaluation measures that can more effectively assess probabilistic masks.

2.5.1 Pixel-Level Metrics

As the name implies, pixel-level metrics treat each pixel as an independent classification decision. Accordingly, they are computed by counting pixel-wise true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These four regions correspond to the overlap and disagreement between the predicted and ground-truth masks, as illustrated in Figure 2.7.

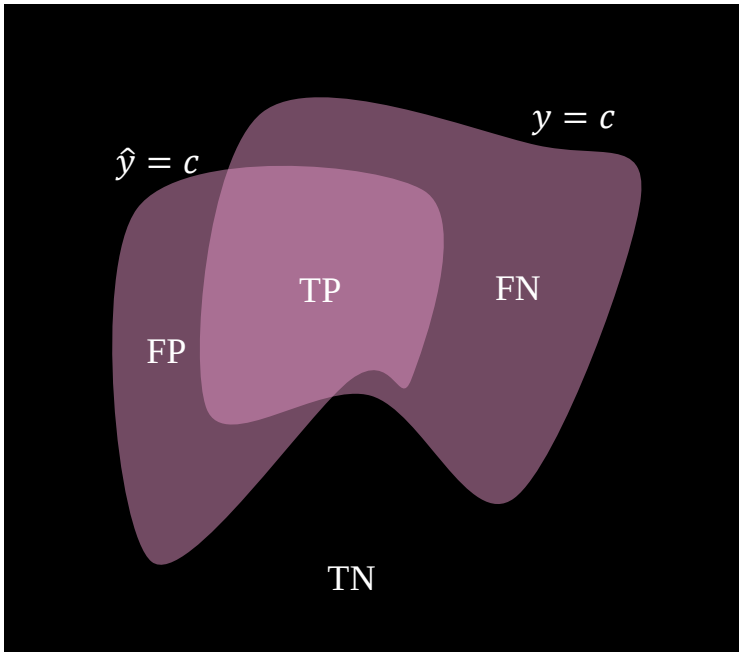


Figure 2.7: Illustration of the relationship between predicted and ground-truth regions for a single class. The overlap between the two shapes corresponds to true positives (TP), while the non-overlapping parts represent false positives (FP) and false negatives (FN). Areas outside both regions correspond to true negatives (TN).

Pixel Accuracy (PA)

This metric measures the proportion of pixels in a dataset that are correctly classified. Given a ground-truth mask y and a predicted segmentation mask $\hat{y}$, pixel accuracy is defined as

$$\text{PA}(y, \hat{y}) = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\sum_i \mathbf{1}[\hat{y}_i = y_i]}{N} \tag{2.17}$$

where $\mathbf{1}[\hat{y}_i = y_i]$ denotes an indicator function that is equal to 1 if pixel i is correctly predicted and 0 otherwise, and N is the total number of pixels in the dataset.

While simple and intuitive, PA is highly sensitive to class imbalance. For instance, if a large proportion of pixels in a dataset belong to background class, a trivial model that labels every pixel as background can still achieve a high PA score.

Mean Pixel Accuracy (mPA)

To mitigate the class imbalance issue, mPA computes the per-class accuracy and averages these values across all classes. Formally,

$$\mathrm{PA}^c(y, \hat{y}) = \frac{TP^c}{TP^c + FN^c} = \frac{\sum_i \mathbf{1}[y_i = c]\mathbf{1}[\hat{y}_i = c]}{\sum_i \mathbf{1}[y_i = c]}, \tag{2.18}$$

$$\mathrm{mPA}(y, \hat{y}) = \frac{1}{C} \sum_{c=1}^{C} \mathrm{PA}^c(y, \hat{y}) \tag{2.19}$$

where C is the total number of classes in the dataset. Although an improvement over raw PA, it still neglects the spatial overlap between predicted and ground-truth regions and considers only pixel-wise correctness. In particular, false positives (pixels mistakenly assigned to class c) are not penalized using mPA, which can lead to an overestimation of a model's performance.

Precision

Although not commonly used for evaluating segmentation networks, some works report per-class precision as a complementary measurement of model performance. This metric is defined as

$$\mathrm{Precision}^c(y, \hat{y}) = \frac{TP^c}{TP^c + FP^c} = \frac{\sum_i \mathbf{1}[y_i = c]\mathbf{1}[\hat{y}_i = c]}{\sum_i \mathbf{1}[\hat{y}_i = c]}, \tag{2.20}$$

In practice, however, precision also ignores the geometric structure of the predicted and ground-truth regions and is therefore less sensitive to spatial misalignments.

2.5.2 Overlap-Based Metrics

Unlike pixel-level metrics, overlap-based metrics assess segmentation quality by computing the intersection and union between predictions and ground-truth masks. This makes them more robust, as they explicitly capture region-level consistency.

Intersection over Union (IoU)

IoU, also referred to as the Jaccard distance in other fields, measures the region-level agreement between two masks by evaluating the size of their intersection relative to the size of their union. Mathematically, IoU for class c is formulated as

$$\mathrm{IoU}^c(y, \hat{y}) = \frac{TP^c}{TP^c + FP^c + FN^c} = \frac{\sum_i \mathbf{1}[y_i = c]\mathbf{1}[\hat{y}_i = c]}{\sum_i \mathbf{1}[y_i = c] + \mathbf{1}[\hat{y}_i = c] - \mathbf{1}[y_i = c]\mathbf{1}[\hat{y}_i = c]}. \quad (2.21)$$

IoU is the most widely used evaluation metric for semantic segmentation. It effectively penalizes both over-segmentation and under-segmentation cases, making it highly reflective of the geometric quality and spatial accuracy of the predicted shapes.

Using the per-class IoU scores, the mean Intersection over Union (mIoU) is defined as

$$\mathrm{mIoU}(y, \hat{y}) = \frac{1}{C} \sum_{c=1}^{C} \mathrm{IoU}^c(y, \hat{y}). \quad (2.22)$$

The standard practice for computing mIoU in the literature is to aggregate all pixels in the dataset rather than computing the metric on a per-mask basis. In other words, this metric treats all ground-truth and predicted masks as one large collection of pixels when calculating the intersection and union.

Dice Coefficient (F1-Score)

The Dice coefficient, also known as the F1-score in binary classification, is another overlap-based metric that measures the similarity between the predicted and ground-truth regions. For a given class c, the Dice score is defined as:

$$\mathrm{Dice}^c(y, \hat{y}) = \frac{2TP^c}{2TP^c + FP^c + FN^c} = \frac{2\sum_i \mathbf{1}[y_i = c]\mathbf{1}[\hat{y}_i = c]}{\sum_i \mathbf{1}[y_i = c] + \sum_i \mathbf{1}[\hat{y}_i = c]}. \quad (2.23)$$

Although closely related to IoU, the Dice coefficient places proportionally more weight on the overlapping region and tends to be more forgiving when the predicted region is considerably larger or smaller than the ground truth. For this reason, it is frequently used in medical image segmentation tasks where foreground regions occupy only a small portion of the image.

2.5.3 Soft Evaluation Metrics

Traditional segmentation metrics discussed in Subsection 2.5.1 and Subsection 2.5.2 assume that ground-truth annotations are hard and one-hot, meaning that each pixel belongs exclusively to a single class. In practice, however, this assumption is increasingly challenged by modern segmentation pipelines, where soft labels naturally appear through downsampling or ambiguous boundaries.

Soft evaluation metrics extend classical hard-label metrics so that they operate directly on per-pixel probability distributions. Rather than assigning each pixel to a single definitive class, these metrics evaluate the similarity or agreement between two distributions at each pixel. Consequently, they provide a more principled assessment of prediction quality in settings where the ground truth is inherently soft or imprecise.

Among the different soft-label extensions, two of the most widely used are the soft IoU (soft Jaccard) [43] and soft Dice [49] metrics. These metrics replace binary intersection and union terms in their corresponding hard-label formulations with continuous overlap functions, allowing them to operate directly on probabilistic masks. Denoting soft ground-truth masks by q and soft predictions by p, the per-class formulations of these two metrics are

$$\text{Soft-IoU}^c_{\text{std}}(q,p) = \frac{\sum_i q_{ic}\, p_{ic}}{\sum_i q_{ic} + \sum_i p_{ic} - \sum_i q_{ic}\, p_{ic}}, \tag{2.24}$$

$$\text{Soft-Dice}^c_{\text{std}}(q,p) = \frac{2\sum_i q_{ic}\, p_{ic}}{\sum_i q_{ic} + \sum_i p_{ic}}. \tag{2.25}$$

Similar to their hard-label counterparts, soft Dice is generally more forgiving than soft IoU, and as a result, soft Dice typically yields higher scores. For this reason, we focus primarily on soft IoU for the remainder of this thesis.

However, the standard soft IoU, as proposed in [43], exhibits an important limitation. A fair evaluation metric should be maximized when the prediction matches the ground-truth distribution, i.e., when $p = q$. In contrast, the standard soft IoU defined in Equation 2.24 does not satisfy this property. To illustrate this and for simplicity, consider a single-pixel scenario: $\frac{\delta}{\delta p}\text{Soft-IoU}^c_{\text{std}}(q,p) = \frac{q^2}{(q+p-qp)^2}$ which is strictly positive for any $q \in (0,1)$. Hence, the standard version is strictly increasing in p. As a result, the function is always maximized at $p = 1$ rather than $p = q$, contradicting the desired behavior.

In this thesis, we focus on introducing refined relaxations of the soft IoU metric that address the limitations of the standard formulation. The relaxations proposed in Section 4

are specifically designed to ensure that the metric behaves consistently when evaluating probabilistic predictions. These refined metrics are then utilized to evaluate the performance of soft-label-trained segmentation networks.

Chapter 3

Proposed Downsampling Approach

To address the limitations of conventional downsampling methods, we seek a downsampling approach that produces smooth, geometry-aware soft labels by assigning spatially varying weights that reflect the contribution of each input pixel to different low-resolution output locations. Our goal is to establish a pipeline that considers the spatial structure, preserves probability distributions, and adapts naturally to arbitrary input and output resolutions.

3.1 Problem Statement

The objective of downsampling a high resolution soft mask $Y \in \mathbb{R}^{H \times W \times C}$ is to convert it into a lower-resolution mask $\widetilde{Y} \in \mathbb{R}^{H' \times W' \times C}$ while preserving both the spatial structure and probabilistic nature of the input. Here, H and W denote the input resolution, H' and W' represent the target resolution, and C is the number of semantic categories. Ideally, each output pixel should represent a geometry-aware aggregation of the input labels, where the contribution of each input pixel is determined by its spatial proximity to the output location rather than by local sampling rules or fixed rectangular regions.

Hence, the central problem is to design a downsampling operation that:

- **Aggregates information over a meaningful spatial region**
- **Assigns weights based on spatial proximity**, ensuring the smoothness of soft labels;
- **Produces valid probability distributions at every output location**; and

- **Adapts naturally to arbitrary input and output resolutions** by adjusting its spatial support according to the scaling factor.

3.2 Method Overview

Our downsampling method, Weighted Average Pooling (WAP), is designed to produce smooth and geometry-aware soft labels by computing spatially varying weights that determine how each input pixel contributes to each output location. The core idea is to precompute a positional contribution weight tensor that depends only on the input and output resolutions, and not on the input mask values. This ensures that the weighting functions capture purely the geometric relationship between resolutions, independent of the content of the mask.

As illustrated in Figure 3.1, WAP proceeds in four steps. First, we construct a positional encoding on the coarse output grid, where each slice corresponds to only a single output pixel and encodes its relative position with respect to all other locations on that grid. Next, each slice in this encoding is smoothly upsampled to the input resolution, producing an unnormalized spatial weighting kernel for every output pixel. These upsampled kernels are then normalized to form valid probability distributions, ensuring that each kernel sums to one and reflects a smooth notion of spatial proximity. Finally, the downsampled label at each output location is computed by taking a weighted average of the input mask using the corresponding normalized kernel.

The positional contribution weights are independent of the input mask and depend solely on the resolutions involved. Moreover, the use of smooth, spatially varying kernels avoids the artifacts produced by hard binning schemes and the limitations introduced by bilinear interpolation. The result is a flexible and resolution-agnostic downsampling pipeline that preserves probabilistic structure while respecting spatial geometry.

3.3 Components of the Method

Our downsampling method consists of four components: positional encoding, upsampling, normalization, and weighted aggregation. Together, these steps construct a set of spatially varying weighting kernels that depend only on the input and output resolutions and are used to produce geometry-aware soft labels.

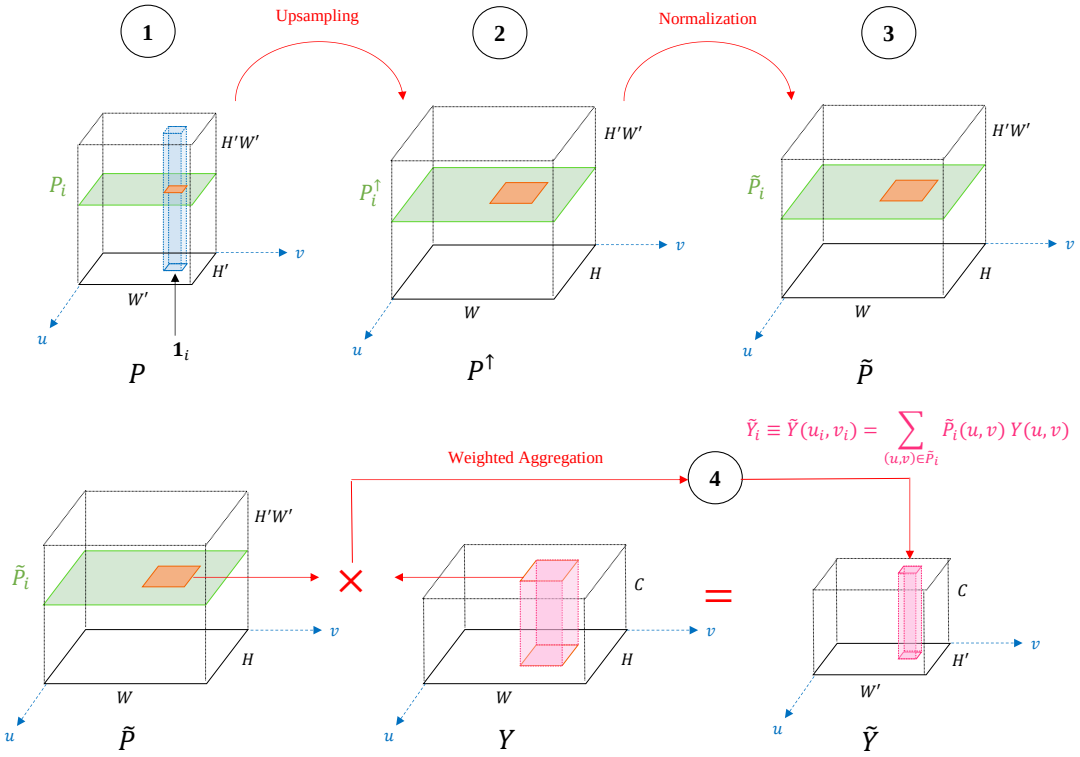


Figure 3.1: Overview of the Weighted Average Pooling (WAP) method for downsampling. (1) A positional encoding is defined on the coarse output grid, with each slice corresponding to a single output pixel. (2) Each slice is smoothly upsampled to the input resolution, producing an unnormalized weighting kernel. (3) The upsampled slices are normalized to form valid spatial probability distributions. (4) Each output soft label is obtained by performing a weighted aggregation of the input mask using the corresponding normalized kernel.

3.3.1 Positional Encoding

We construct a positional encoding tensor $P \in \{0,1\}^{H' \times W' \times (H'W')}$, where each of the $H'W'$ slices corresponds to a unique output pixel on the coarse grid. The encoding is defined in a way that each slice is a one-hot map, and each column of the tensor is also one-hot.

Formally, we index output pixels by $i \in \{1, \cdots, H'W'\}$, and let (u_i, v_i) denote the spatial coordinates of pixel i on the $H' \times W'$ grid.

The positional encoding is then defined as

$$P_i(u,v) = \begin{cases} 1, & \text{if } (u,v) = (u_i, v_i), \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

As the definition suggests, each slice P_i is a one-hot indicator that activates exactly at the spatial location of output pixel i. Moreover, each spatial location (u, v) across all slices exactly corresponds to a single index in the depth dimension, thus $P(u, v)$ is also a one-hot column. This establishes a bijection between coarse-grid pixels and the slices in the encoding.

3.3.2 Upsampling

To transfer the coarse positional patterns to the input resolution, each slice P_i is smoothly upsampled to size $H \times W$, producing $P_i^{\uparrow} \in \mathbb{R}^{H \times W}$. The collection of all upsampled slices forms $P^{\uparrow} \in \mathbb{R}^{H \times W \times H'W'}$. Concisely, $P^{\uparrow}$ is defined as

$$P^{\uparrow} = U(P), \tag{3.2}$$

where U is any standard upsampling operator. In this thesis, we adopt bilinear interpolation as the upsampling function U, as it is commonly utilized in modern segmentation networks for upsampling logits.

A visual example of this process is shown in Figure 3.2. When a one-hot slice P_i is upsampled using bilinear interpolation, the only nonzero entry at spatial location (u_i, v_i) spreads smoothly across neighboring positions on the high-resolution grid. As a result, the value 1 in the coarse grid is distributed into a small region of nonzero weights in the high-resolution grid, with the highest value located near the original position and progressively smaller values assigned to surrounding pixels. This produces a smooth spatial kernel that reflects geometric proximity and serves as the unnormalized contribution map for output pixel i.

3.3.3 Normalization

In the third step, each upsampled slice $P_i^{\uparrow}$ is normalized to obtain a valid spatial probability distribution:

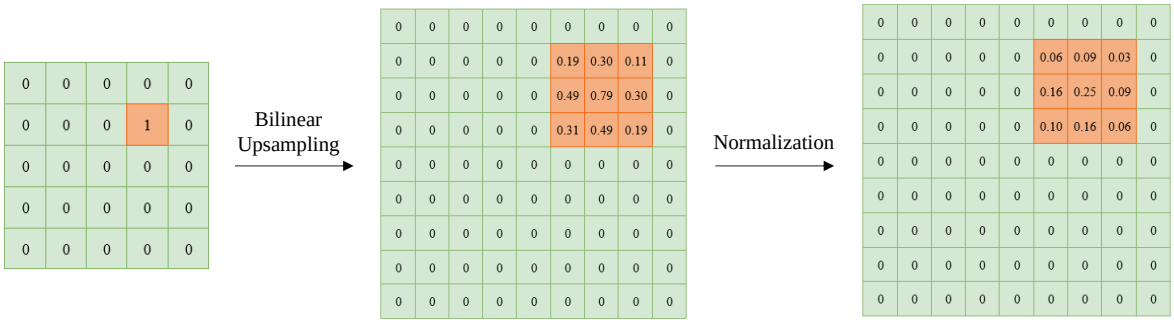


Figure 3.2: Example of bilinear upsampling and normalization of a positional encoding slice for the case $H = W = 9$ and $H' = W' = 5$. Bilinear interpolation spreads the single nonzero entry into a smooth spatial neighborhood whose values decay with distance from the original location. Normalization then converts this pattern into a valid spatial probability distribution.

$$\widetilde{P}_i = \frac{1}{\sum_{u,v} P_i^{\uparrow}(u,v)} P_i^{\uparrow} \tag{3.3}$$

Because $P_i^{\uparrow}$ is generated by smoothly upsampling a one-hot slice, the normalization preserves the spatial structure of the kernel while converting it into a proper probability map. The resulting tensor $\widetilde{P}_i$ is therefore a geometry-aware normalized contribution map that quantifies how strongly each input pixel influences the output location i.

3.3.4 Weighted Aggregation

Given the input soft label map $Y \in \mathbb{R}^{H \times W \times C}$, the downsampled output at location u_i, v_i is computed using a geometry-aware weighted average of all input labels:

$$\widetilde{Y}(u_i, v_i) = \sum_{(u,v) \in H \times W} \widetilde{P}_i(u,v)\, Y(u,v) \tag{3.4}$$

The resulting vector $\widetilde{Y}(u_i, v_i)$ remains a valid soft label, as

$$
\begin{aligned}
\sum_{c=1}^{C} \widetilde{Y}(u_i, v_i, c) &= \sum_{c=1}^{C} \sum_{(u,v)\in H\times W} \widetilde{P}_i(u,v)\, Y(u,v,c) \\
&= \sum_{(u,v)\in H\times W} \widetilde{P}_i(u,v) \underbrace{\sum_{c=1}^{C} Y(u,v,c)}_{1} \\
&= \sum_{(u,v)\in H\times W} \widetilde{P}_i(u,v) \\
&= 1,
\end{aligned}
\tag{3.5}
$$

preserving its probabilistic interpretation across the C semantic classes. Each output pixel aggregates the input labels according to a meaningful, smoothly defined spatial region rather than a fixed grid or a rectangular bin.

3.4 Advantages

WAP offers several advantages over traditional methods, such as nearest-neighbor interpolation, bilinear interpolation, area-based interpolation, and AAP. These advantages arise from the use of geometry-aware weighting kernels that depend only on the input and output resolutions.

1. **Geometry-aware Weighting**

   In contrast with nearest-neighbor or bilinear interpolation, which operate over fixed and often minimal neighborhoods, WAP constructs spatial kernels whose support naturally reflects the scale of the downsampling operation. Each output pixel aggregates information from a spatial region proportional to the change in resolution, ensuring that the downsampled labels respect the geometric structure of the input.

2. **Smooth Spatial Variation**

   The weighting kernels vary smoothly across the input mask as a result of continuous upsampling of the positional encoding. Consequently, the downsampled labels transition more naturally across space, eliminating the block artifacts of uniform pooling and the sharp discontinuities associated with nearest-neighbor approach.

3. **Resolution-agnostic Design**

   As the positional weights depend only on the input and output resolutions, WAP adapts to arbitrary scaling factors. In contrast to bilinear interpolation, which always uses a fixed neighborhood regardless of the scaling ratio, our approach automatically adjusts its effective receptive field based on the resolution change.

4. **Preservation of Probabilistic Structure**

   Because each spatial kernel is normalized, the resulting downsampled labels remain valid soft probability distributions across semantic classes. This property is essential for training segmentation models that rely on soft-label formulations.

3.5 Summary

This chapter introduced WAP, a geometry-aware downsampling method designed to generate smooth and spatially coherent soft labels. Unlike conventional downsampling methods, which rely on local sampling or uniform averaging, the proposed approach uses spatially varying weighting kernels derived exclusively from the geometric relationship between input and output resolutions. By constructing a positional encoding, upsampling it to the input resolution, normalizing the resulting kernels, and applying them to aggregate the input label map, WAP produces soft labels that reflect spatial proximity, preserve probabilistic structure, and adapt naturally to arbitrary scaling factors.

To conclude, Algorithm 1 summarizes the complete downsampling procedure presented in this chapter.

Algorithm 1 Weighted Average Pooling for Downsampling

Require: Input soft mask $Y \in \mathbb{R}^{H \times W \times C}$, target resolution (H', W')

Ensure: Downsampled soft mask $\widetilde{Y} \in \mathbb{R}^{H' \times W' \times C}$

1: **for** $i = 1$ to $H'W'$ **do**

2: Let (u_i, v_i) be the coordinates of output index i on the $H' \times W'$ grid

3: Define the one-hot positional slice $P_i \in \{0, 1\}^{H' \times W'}$ by $P_i(u, v) = 1$ if $(u, v) = (u_i, v_i)$ and 0 otherwise

4: Upsample the slice to input resolution: $P_i^{\uparrow} \leftarrow U(P_i)$

5: Normalize to obtain a spatial probability distribution:

$$\widetilde{P}_i(u, v) = \frac{P_i^{\uparrow}(u, v)}{\sum_{u', v'} P_i^{\uparrow}(u', v')}$$

6: Compute the downsampled soft label at (u_i, v_i):

$$\widetilde{Y}(u_i, v_i) = \sum_{(u,v) \in H \times W} \widetilde{P}_i(u, v)\, Y(u, v)$$

7: **end for**

8: **return** $\widetilde{Y}$

Chapter 4

Relaxations of Soft-mIoU for Evaluation

As discussed in Subsection 2.5.3, the standard soft IoU (Equation 2.24) fails to satisfy the fundamental consistency property Soft-IoU$^{\text{std}}(q, p) = 1$ when $p = q$, except in the degenerate case where both distributions are one-hot. Figure 4.1 visualizes this behavior for two classes and a single pixel case. Despite $p = q$ the surface never reaches 1 except at the binary extremes, clearly showing that the standard soft IoU underestimates similarity whenever the distributions are soft, precisely the setting needed to evaluate models trained with soft ground-truth masks.

Given this limitation, we introduce three relaxations of soft IoU that properly satisfy the consistency requirement and provide more reliable behavior for soft-label evaluation.

4.1 Symmetric Difference Relaxation

As discussed in [55], for each class c in the dataset and an arbitrary pixel i, we can define the intersection of the binary prediction mask p and binary ground-truth mask q as $v_i^c = p_i^c \cap q_i^c$, the union as $u_i^c = p_i^c \cup q_i^c$, and the symmetric difference representing mispredictions as $m_i^c = (p_i^c \setminus q_i^c) \cup (q_i^c \setminus p_i^c)$. It is well known that $|v_i^c| = p_i^c q_i^c$ and $|m_i^c| = |p_i^c - q_i^c|$. The union can therefore be expressed as

$$|u_i^c| = |v_i^c| + |m_i^c| = p_i^c q_i^c + |p_i^c - q_i^c|. \tag{4.1}$$

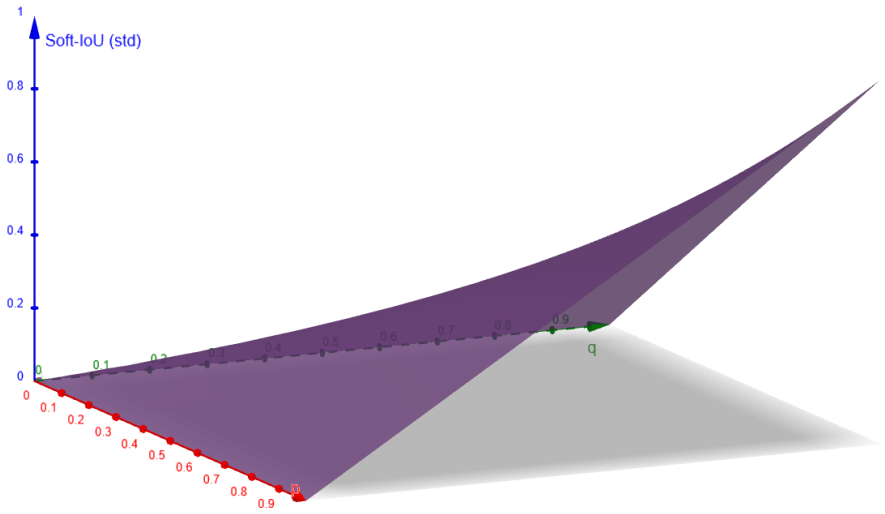


Figure 4.1: Illustration of the standard soft IoU formulation for a single pixel with two classes. Even along the diagonal $p = q$ the metric does not output the value 1 except at the one-hot corner (p = q = 1). Thus, even perfect agreement between prediction and ground truth does not yield maximal IoU unless both are one-hot.

For soft labels $p_i^c, q_i^c \in [0, 1]$, the set-theoretic identity no longer holds exactly. Nevertheless, we extend the same decomposition by defining relaxed overlap and error terms

$$|\tilde{v}_i^c| = p_i^c q_i^c, \qquad |\tilde{m}_i^c| = |p_i^c - q_i^c|, \qquad |\tilde{u}_i^c| = |\tilde{v}_i^c| + |\tilde{m}_i^c| = p_i^c q_i^c + |p_i^c - q_i^c| \tag{4.2}$$

Thus, we define a relaxation of the intersection-over-union for soft labels as

$$\text{Soft-IoU}_{\text{sd}}^c(p, q) = \frac{\sum_i |\tilde{v}_i^c|}{\sum_i |\tilde{u}_i^c|} = \frac{\sum_i p_i^c q_i^c}{\sum_i p_i^c q_i^c + \sum_i |p_i^c - q_i^c|}. \tag{4.3}$$

This relaxation effectively resolves the main issue with the standard formulation. When $p = q$, we have $|p_i^c - q_i^c| = 0$, and therefore $\text{Soft-IoU}_{\text{sd}}^c(p, q) = 1$. Figure 4.2 illustrates the behavior of the symmetric-difference–based soft IoU relaxation for the two-class, single-pixel case.

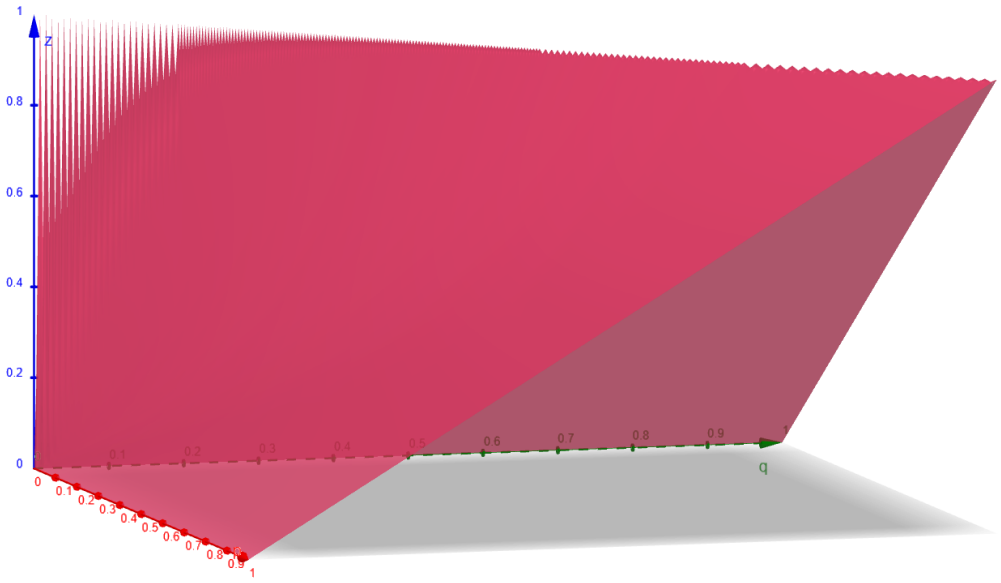


Figure 4.2: Visualization of the symmetric-difference soft IoU for a two-class, single-pixel setting. The surface shows the value of the metric as a function of p and q.

4.2 Square-Root Relaxation

The square-root relaxation replaces the multiplicative intersection of two distributions with their geometric mean. The resulting formulation is defined as

$$\text{Soft-IoU}^c_{\surd}(p, q) = \frac{\sum_i \sqrt{p_i^c q_i^c}}{\sum_i p_i^c + \sum_i q_i^c - \sum_i \sqrt{p_i^c q_i^c}}. \tag{4.4}$$

This modification preserves the standard (hard) IoU when p and q are one-hot masks, since $\sqrt{p_i^c q_i^c} = p_i^c q_i^c$ in the binary case. In contrast to the standard soft IoU, the square-root relaxation satisfies the consistency property. When $p = q$,

$$\text{Soft-IoU}^c_{\surd}(p, q) = \frac{\sum_i \sqrt{p_i^c p_i^c}}{\sum_i p_i^c + \sum_i p_i^c - \sum_i \sqrt{p_i^c p_i^c}} = \frac{\sum_i p_i^c}{\sum_i p_i^c} = 1,$$

and consequently, perfect agreement between prediction and ground-truth distributions results in the metric evaluating to one even when soft labels are used. It is also guaranteed

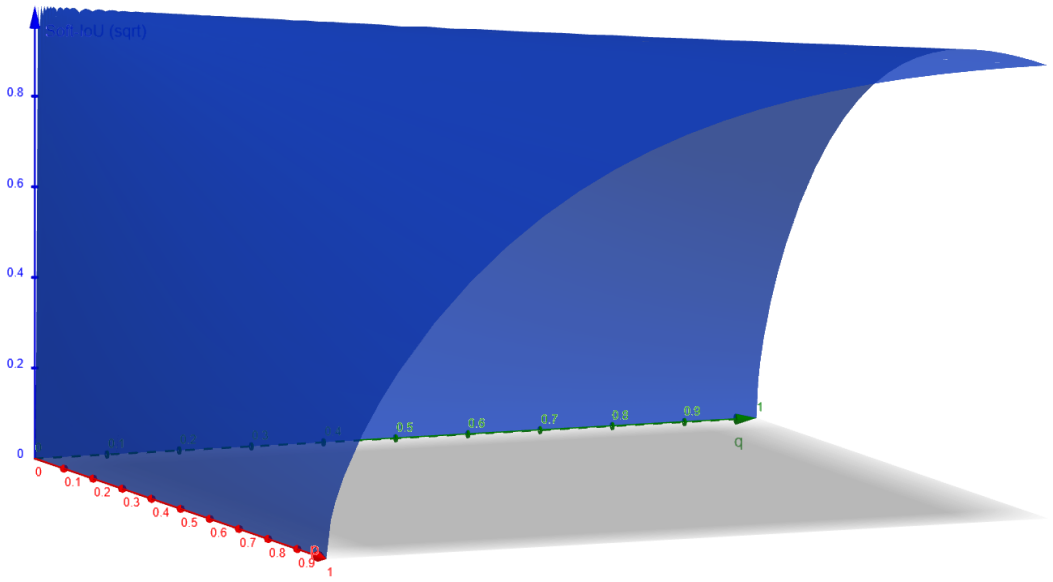


Figure 4.3: Visualization of the square-root soft IoU for a two-class, single-pixel setting. The surface shows the value of the metric as a function of p and q.

that $\text{Soft-IoU}^c_{\surd}(p, q) \leq 1$, since

$$\sqrt{p_i^c q_i^c} \leq p_i^c + q_i^c - \sqrt{p_i^c q_i^c}$$

holds for any pixel i.

Figure 4.3 illustrates the behavior of the square-root relaxation for the two-class, single-pixel setup.

4.3 Second-Order (Squared) Relaxation

The second-order relaxation modifies the standard soft IoU by incorporating squared magnitudes in the union term, leading to a sharper notion of overlap between distributions. The resulting formulation is defined as

$$\text{Soft-IoU}^c_{\text{sq}}(p, q) = \frac{\sum_i p_i^c q_i^c}{\sum_i (p_i^c)^2 + \sum_i (q_i^c)^2 - \sum_i p_i^c q_i^c}. \tag{4.5}$$

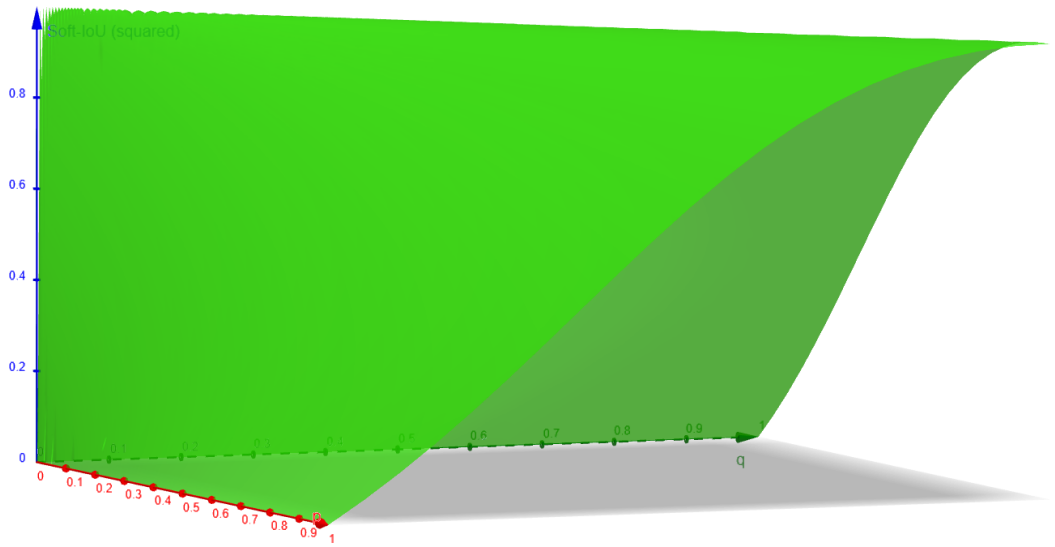


Figure 4.4: Visualization of the second-order soft IoU for a two-class, single-pixel setting. The surface shows the value of the metric as a function of p and q.

This formulation preserves the standard (hard) IoU when p and q are one-hot masks, since $(p_i^c)^2 = p_i^c$ and $(q_i^c)^2 = q_i^c$ in the binary case. In contrast to the standard soft IoU, the second-order relaxation satisfies the consistency property. When $p = q$,

$$\text{Soft-IoU}_{\text{sq}}^c(p, q) = \frac{\sum_i (p_i^c)^2}{\sum_i (p_i^c)^2 + \sum_i (p_i^c)^2 - \sum_i (p_i^c)^2} = \frac{\sum_i (p_i^c)^2}{\sum_i (p_i^c)^2} = 1,$$

and consequently, perfect agreement between prediction and ground-truth distributions results in the metric evaluating to one even when soft labels are used. Furthermore, it is guaranteed that $\text{Soft-IoU}_{\text{sq}}^c(p, q) \leq 1$, since

$$p_i^c q_i^c \leq (p_i^c)^2 + (q_i^c)^2 - p_i^c q_i^c$$

holds for any pixel i.

Figure 4.4 illustrates the behavior of the second-order relaxation for the two-class, single-pixel setup.

4.4 Summary

This chapter examined the limitations of existing IoU-based evaluation metrics when applied to soft predictions and soft ground-truth annotations. Although soft extensions of IoU are commonly used in practice, we showed that the intrinsic inconsistency of the standard soft IoU formulation undermines the interpretability and reliability of soft-label evaluation, particularly in settings where uncertainty is an inherent part of the supervision.

To address this issue, we introduced a family of principled relaxations of this metric designed specifically for soft-label semantic segmentation. These relaxations modify the intersection and union operators in a way that preserves probabilistic validity while guaranteeing consistency with the hard-label IoU in the discrete limit. Crucially, all proposed variants achieve their maximum value under perfect agreement between prediction and ground truth, thereby restoring the expected behavior of an evaluation metric.

Overall, this chapter highlights that evaluation metrics must be designed in accordance with the nature of the supervision. The proposed relaxations offer a principled and consistent foundation for evaluating soft-label semantic segmentation.

Chapter 5

Void Pixels as Uncertain Supervision

The PASCAL VOC dataset includes a designated void label to mark pixels for which annotators were unable to assign a confident or unique semantic class. These pixels typically occur near object boundaries, in thin or low-contrast regions, or in regions where multiple semantic interpretations are plausible, as illustrated in Figure 5.1. Void pixels are usually treated as missing labels by standard semantic segmentation networks and are therefore excluded from both training and evaluation.

While convenient, this practice implicitly assumes that void regions carry no beneficial information. In contrast, we argue that void pixels represent a meaningful source of annotation ambiguity and therefore naturally lend themselves to a soft-label interpretation. Rather than discarding these pixels, we propose replacing void labels with probabilistic distributions over semantic classes, enabling their participation in training within the proposed soft-label segmentation framework.

Formally, let $y_i \in \{1, 2, \ldots, C\} \cup \{\text{void}\}$ denote the hard ground-truth label at pixel i. Our goal is to construct a soft label $q_i \in \Delta^{C-1}$. We investigate two complementary strategies for constructing such soft labels.

5.1 Distance-Based Void Replacement

The first approach constructs soft labels for void pixels based on their proximity to surrounding confidently labeled pixels. The underlying intuition is that void pixels often occur near one or more semantic classes, for instance along object boundaries, and that their ambiguity can be approximated by computing distances to nearby labeled regions.



Figure 5.1: Example from the PASCAL VOC dataset illustrating void pixels in the annotation mask. Void regions (white) appear both as thin boundary pixels around object contours and as large contiguous areas corresponding to missing or unlabeled objects. These regions reflect different sources of annotation ambiguity.

For each class $c \in \{1, 2, \ldots, C\}$, let Ω_c denote the set of pixels labeled as class c. For each void pixel i, we compute the minimum Euclidean distance to each class region as

$$d_c(i) = \min_{i' \in \Omega_c} \|i - i'\|_2 \,. \tag{5.1}$$

For classes that are not present in the input image, the corresponding set Ω_c is empty, and we define $d_c(i) = \infty$. This convention ensures that absent classes receive zero probability mass in the subsequent weighting.

The computed distances are then converted to class probabilities using inverse-distance weighting. Hence, the soft label assigned to void pixel i for class c is defined as

$$q_i^c = \frac{d_c(i)^{-1}}{\sum_{k=1}^{C} d_k(i)^{-1}} \,. \tag{5.2}$$

This normalization creates a valid probability distribution over classes, assigning higher

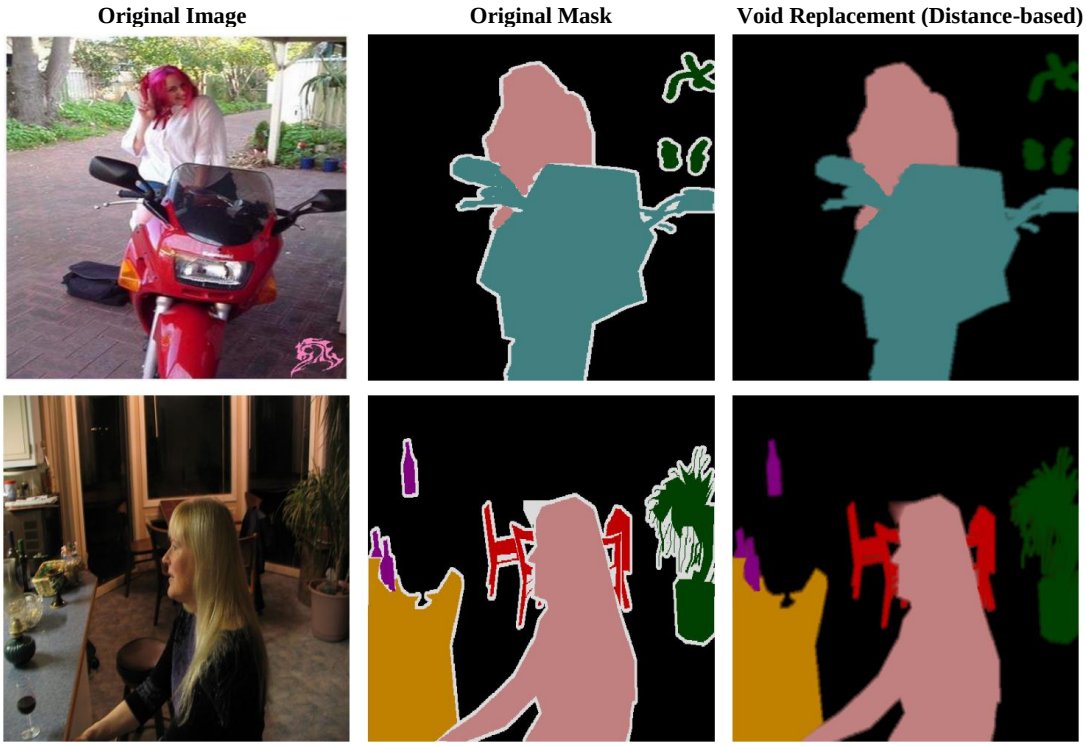


Figure 5.2: Illustration of the proposed distance-based void replacement strategy on PASCAL VOC examples. From left to right: the original input image, the corresponding segmentation mask containing void pixels, and the resulting soft-label visualization after applying distance-based void replacement. Void regions near object boundaries are replaced with soft distributions that reflect proximity to surrounding semantic regions, while absent classes are suppressed.

probability to classes that are spatially closer to the void pixel while suppressing distant or non-present classes, as illustrated in Figure 5.2.

To better illustrate the behavior of the proposed method near object boundaries, Figure 5.3 provides magnified views of several regions around class transitions. As shown, void pixels located along thin boundaries are replaced with smoothly varying soft distributions rather than abrupt hard assignments.

While simple, interpretable, and well suited for replacing void pixels that arise along object boundaries or thin structures, this approach has a clear limitation: void pixels may also correspond to larger ambiguous regions in an image. In such cases, the uncertainty may

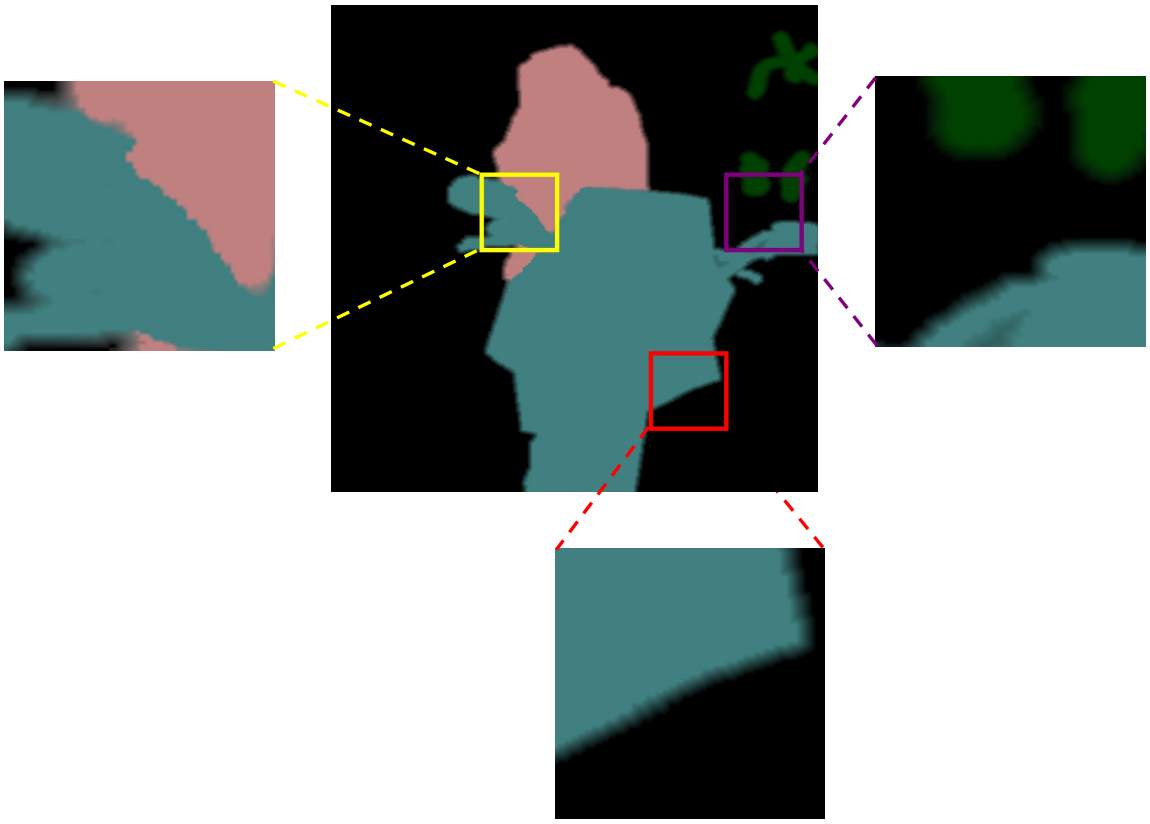


Figure 5.3: Magnified visualization of boundary behavior under distance-based void replacement. The central panel shows the soft-label map with highlighted boundary regions. The surrounding zoomed-in regions reveal a gradual transition in probability mass across class boundaries, highlighting how void pixels near object edges are assigned smooth distributions rather than discrete assignments.

involve semantic classes that are not present elsewhere in the image. Since the proposed method constructs soft labels only over classes present in the input image, it is unable to assign probability mass to absent but semantically feasible classes.

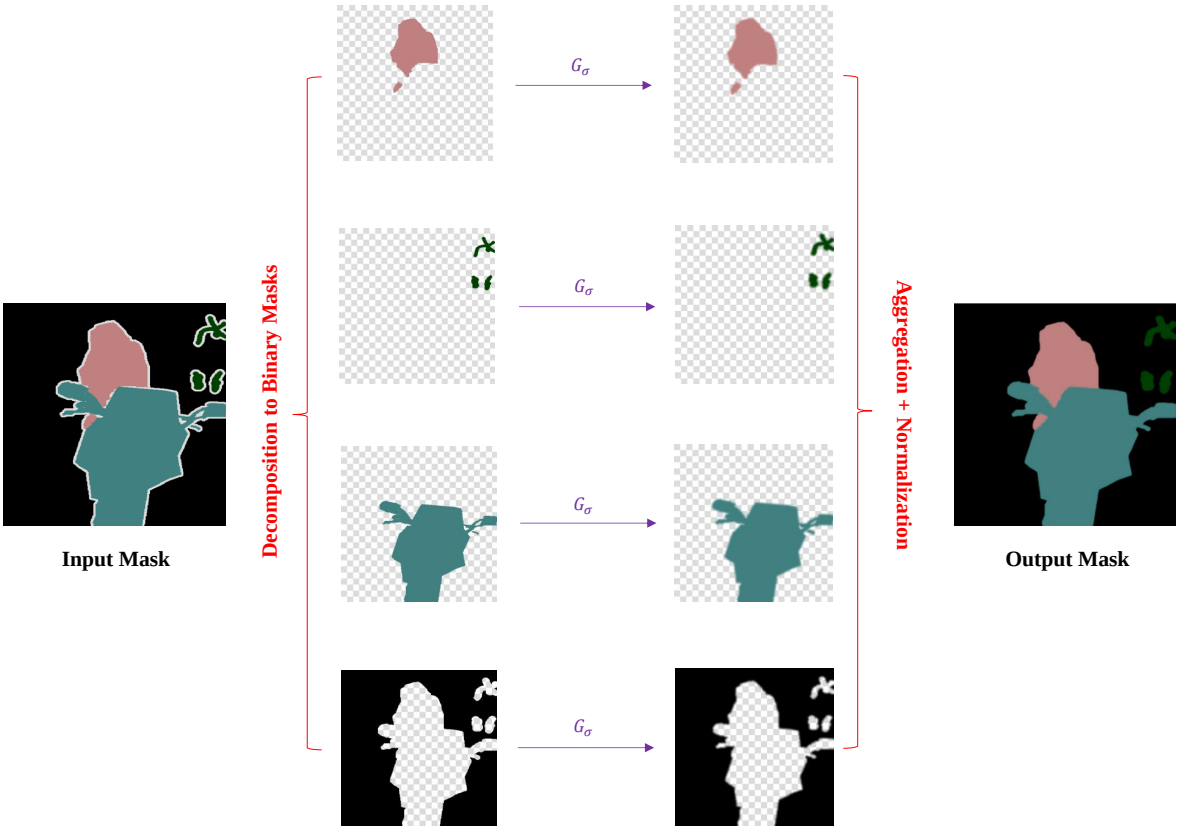


Figure 5.4: Overview of the Gaussian void replacement pipeline. The input segmentation mask is first decomposed into class-wise binary maps. Gaussian smoothing is applied independently to each binary map, allowing class evidence to spread spatially. The smoothed responses are then aggregated and normalized across classes to produce the final soft-label segmentation mask.

5.2 Gaussian Void Replacement

The second proposed approach applies a diffusion-based formulation of annotation uncertainty. Instead of explicitly computing distances to class regions, semantic evidence is propagated spatially by applying Gaussian smoothing to binary class maps (see Figure 5.4), which are then integrated and normalized to obtain a final soft class distribution.

Formally, for each present class c, we construct a binary indicator function $\mathbf{1}_c(i)$, which equals one if pixel i is labeled as class c, and zero otherwise. Let G_σ represent a Gaussian

kernel with standard deviation σ. We apply convolution to produce class confidence scores:

$$s_c(i) = (G_\sigma * \mathbf{1}_c)(i). \tag{5.3}$$

Finally, at each pixel location, these smoothed confidence maps are normalized across classes to form the probability distribution

$$q_i^c = \frac{s_c(i)}{\sum_{k=1}^{C} s_k(i)}\,. \tag{5.4}$$

Figure 5.5 provides examples of the resulting soft labels on PASCAL VOC, illustrating how Gaussian smoothing produces gradual transitions near boundaries and ambiguous regions. A notable advantage of this diffusion-based approach is that it not only assigns soft labels to void pixels, but also also relaxes originally hard annotations by introducing a controlled degree of softness in those regions. Through Gaussian smoothing, semantic evidence could propagate locally beyond class boundaries, leading to smooth transitions between neighboring classes. This behavior is particularly beneficial in regions where hard annotations may be noisy or misaligned with true object boundaries, as it provides a natural mechanism for mitigating annotation errors.

To further highlight the spatial behavior of the Gaussian formulation, Figure 5.6 presents magnified views of selected boundary regions. Compared to the distance-based approach, the Gaussian method produces visibly smoother transitions across different classes, and the boundary itself appears less abrupt. This effect is a result of the spatial propagation of class evidence through convolution with the Gaussian kernel.

The standard deviation σ of the Gaussian kernel controls the spatial extent of the smoothing applied to class evidence. Smaller values of σ confine this effect to a local neighborhood, resulting in soft labels primarily near object boundaries while largely preserving the original hard annotations elsewhere. In contrast, larger values of σ result in broader smoothing, producing smoother class distributions and greater tolerance to annotation noise at the expense of reduced boundary sharpness. Consequently, σ defines an explicit trade-off between locality and robustness, allowing the degree of softness in the supervision to be adjusted according to the desired sensitivity to annotation uncertainty.

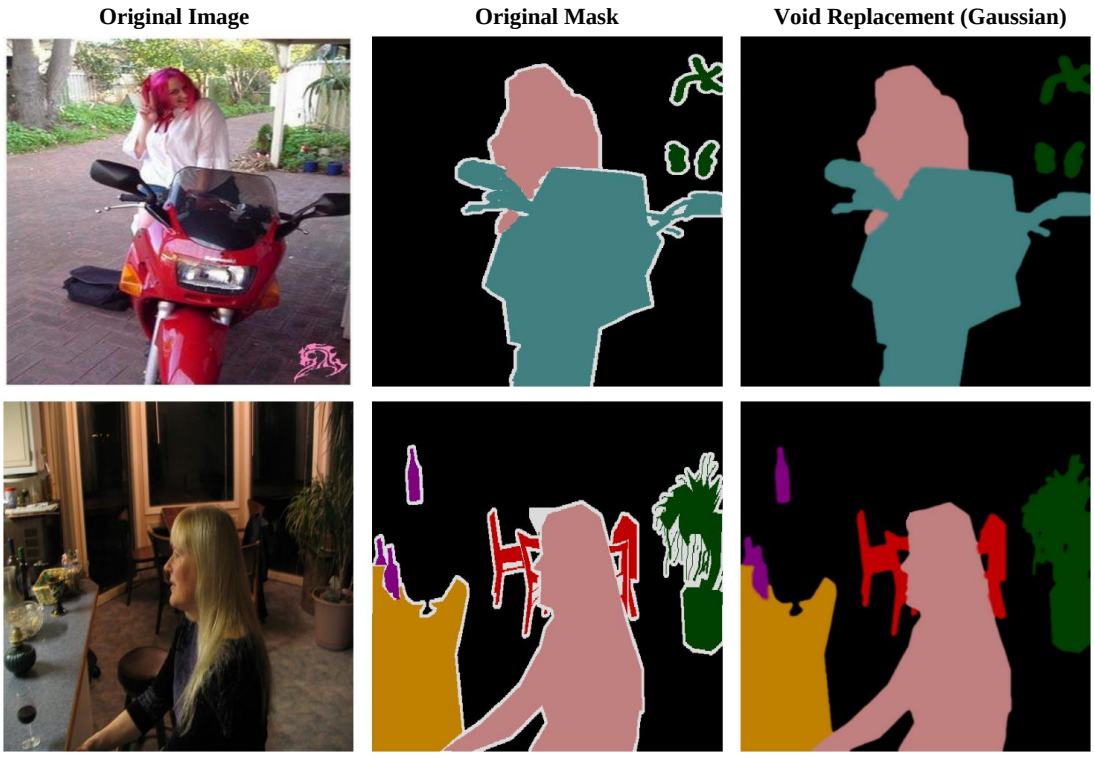


Figure 5.5: Illustration of the proposed Gaussian void replacement strategy on PASCAL VOC examples. From left to right: the original input image, the corresponding segmentation mask containing void pixels, and the resulting soft-label visualization after applying Gaussian smoothing. Compared to distance-based replacement, this approach introduces gradual transitions not only in void regions but also near originally hard boundaries, while still suppressing absent classes.

5.3 Summary

In this chapter, two principled strategies for replacing void pixels with soft semantic labels were introduced: a distance-based formulation and a Gaussian smoothing formulation. The distance-based approach assigns class probabilities according to geometric proximity to surrounding labeled regions, providing an interpretable and localized mechanism for modeling boundary uncertainty. In contrast, the Gaussian formulation propagates semantic evidence through convolutional smoothing, producing spatially continuous class distributions that relax both void and originally hard annotations. While the former emphasizes proximity-driven weighting centered around ambiguous pixels, the latter induces

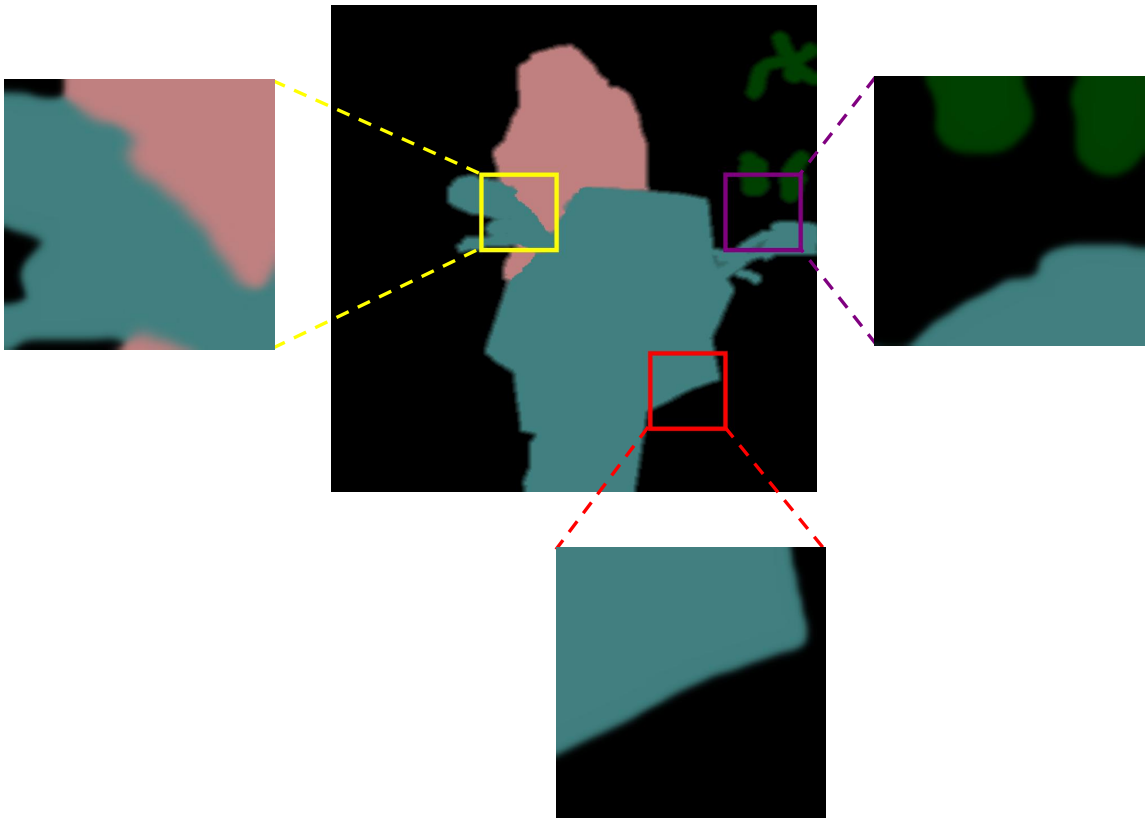


Figure 5.6: Magnified visualization of boundary behavior under Gaussian void replacement. The highlighted regions illustrate how Gaussian smoothing induces gradual transitions across object edges. Compared to distance-based void replacement, the boundary appears smoother due to diffusion of class evidence through convolution with the Gaussian kernel.

a broader regularization effect that yields smoother transitions and increased robustness to annotation noise. Together, these two approaches provide complementary ways to model annotation uncertainty, offering an alternative to traditional hard-label segmentation.

Chapter 6

Experiments

6.1 Dataset

All experiments were conducted on the PASCAL VOC 2012 [14] semantic segmentation dataset, which contains 21 semantic classes (20 foreground categories plus background). Following standard practice, training was performed using the VOC 2012 training split together with the augmented annotations from [15]. This results in approximately 10,582 training images, compared to the 1,464 images in the original training set. The dataset also includes void pixels (regions marked as ambiguous or unlabeled), which are encoded with value 255. During training, these pixels were ignored in the loss function and did not contribute to gradient updates. For evaluation, we used the official VOC 2012 validation split, consisting of 1,449 images.

6.2 Implementation Details

In this thesis, we use pretrained DINOv2-Base weights from [40] as a frozen backbone for feature extraction. On top of these features, we train lightweight semantic segmentation heads. Specifically, we consider (i) a linear head, and (ii) a three-layer convolutional head consisting of two 3×3 convolutional layers with padding 1, followed by a final 1×1 convolutional layer that maps features to class logits. All segmentation heads are trained for 40 epochs in the reported experiments.

During training, all input images and ground-truth segmentation masks are resized to a resolution of 518×518, following the setup recommended for DINOv2-based dense pre-

diction tasks. Given the default patch size of 14 used by DINOv2, this results in backbone feature maps of spatial resolution 37×37. In experiments involving soft-label supervision (excluding void replacement experiments), the ground-truth masks are therefore downsampled to this coarse resolution to align with the feature map. The Adam optimizer is used to minimize the corresponding loss functions, with a learning rate of 10^{-3}.

At validation stage, each input image and its corresponding ground-truth mask are rescaled to the closest multiple of 14 in each spatial dimension, ensuring compatibility with the patch-based backbone without introducing padding artifacts. For fine-level evaluation, the predictions are upsampled back to this resolution before metric computation. For coarse-level evaluation, ground-truth masks are downscaled to a resolution of $(H/14, W/14)$ using the proposed geometry-aware downsampling method, and evaluation is performed directly at the coarse-level resolution.

6.3 Handling of Void Pixels at Coarse Resolution

The PASCAL VOC dataset contains void pixels, corresponding to ambiguous or boundary regions, which are represented by a dedicated ignore value. For training and evaluation with hard masks, void pixels are simply ignored. They do not contribute to the loss function and are excluded from all evaluation metrics, following standard practice in semantic segmentation.

When working with soft labels and soft evaluation metrics, however, a dedicated strategy for handling void regions becomes necessary. Downsampling a ground-truth mask that contains void pixels may lead to coarse locations that mix semantic classes with void regions, resulting in ambiguous supervision. To address this, let the dataset contain C semantic classes, and denote the soft ground-truth label at pixel i by

$$\bar{q}_i = (\bar{q}_i^1, \cdots, \bar{q}_i^C, \bar{q}_i^{\text{void}}) \in \Delta^C, \tag{6.1}$$

where $\bar{q}_i^{\text{void}}$ represents the probability of pixel i being void. We define the transparency of pixel i as the probability of it being non-void,

$$\alpha_i = 1 - \bar{q}_i^{\text{void}}. \tag{6.2}$$

The refined soft label $q_i \in \Delta^{C-1}$ over the semantic classes is then implicitly defined via

$$\alpha_i q_i = (\bar{q}_i^1, \cdots, \bar{q}_i^C). \tag{6.3}$$

The transparency term α_i also plays a crucial role during optimization. In particular, $\alpha_i = 0$ corresponds to pixels that are certainly void, for which no learning updates should be applied. We incorporate this behavior by treating α_i as a per-pixel weight when computing all loss functions introduced in Section 2.4, ensuring that void regions do not contribute to parameter updates.

6.4 Quantitative Results

6.4.1 Effect of Downsampling Strategy

We compare nearest-neighbor, bilinear, area-based, and the proposed geometry-aware downsampling approach under both hard and soft supervision. Results for AAP are not reported, as its performance matches that of area-based downsampling when the rescaling factor is an integer, which is the case in our experimental setup. Fine-level performance is reported using standard mIoU. Nearest-neighbor downsampling yields inherently hard labels and is therefore evaluated only under hard supervision, whereas the remaining downsampling methods are evaluated under both hard and soft supervision, where hard labels are obtained by applying an argmax operation to the corresponding soft labels produced by each downsampling method. For all experiments involving soft supervision, CCE loss is used, and soft ground-truth masks are obtained using the corresponding downsampling strategy.

Table 6.1 reports results on a linear head and an MLP head. Across both heads, nearest-neighbor downsampling under hard supervision consistently results in lower performance than all other strategies, indicating significant information loss when discrete labels are propagated to the coarse resolution without any form of spatial aggregation.

In contrast, smoother downsampling strategies, including bilinear, area-based, and the proposed WAP method, achieve consistently higher mIoU scores. For the linear head, these methods cluster tightly around similar performance levels. Notably, WAP method achieves the highest performance under hard supervision and remains competitive under soft supervision.

Almost similar trends are observed for the MLP head, which consistently outperforms the linear head. Nearest-neighbor downsampling remains suboptimal, while differences

Table 6.1: Comparison of different downsampling approaches under hard and soft supervision at coarse resolution, evaluated using a linear and an MLP segmentation head. Fine-level performance is reported using standard mIoU. Nearest-neighbor downsampling is evaluated only under hard supervision, while all other downsampling methods are evaluated under both hard and soft supervision. Models trained with soft supervision use CCE loss, whereas hard-supervised counterparts use standard CE loss.

Linear Head		**MLP Head**	
GT Downsampling	mIoU (std)	GT Downsampling	mIoU (std)
Nearest-Neighbor	79.58	Nearest-Neighbor	81.02
Bilinear (hard)	81.47	Bilinear (hard)	84.21
Bilinear (soft)	81.38	Bilinear (soft)	84.33
Area (hard)	81.41	Area (hard)	84.38
Area (soft)	81.18	Area (soft)	84.15
WAP (hard)	81.49	WAP (hard)	84.49
WAP (soft)	81.33	WAP (soft)	84.37

among smoother methods are relatively small. Among these, WAP again achieves the strongest performance under both hard and soft supervision, though the performance gains over other smooth strategies are small.

Overall, these results highlight the importance of using distribution-aware downsampling strategies when training with coarse supervision, as naive discretization via nearest-neighbor interpolation leads to avoidable performance degradation.

6.4.2 Effect of the Loss Function under Soft Supervision

Table 6.2 isolates the effect of the training loss function when learning from coarse-level soft ground-truth masks. All methods share the same backbone, proposed downsampling strategy, and optimization setup; only the loss function and the segmentation head are different in the experiments.

Overall, the standard mIoU shows limited sensitivity to the choice of loss under soft supervision for both head architectures. For the linear head, all losses result in comparable performance, with differences limited to a narrow range below 0.5%. A similar trend is observed for the MLP head, where performance variations remain small despite the

Table 6.2: Effect of the training loss function under coarse-level soft supervision. WAP is used as the downsampling strategy in all experiments, and results are reported for both linear and MLP segmentation heads.

Linear Head		**MLP Head**	
Loss Function	mIoU (std)	Loss Function	mIoU (std)
CCE	81.33	CCE	84.37
CE	80.87	CE	84.23
RKL	81.22	RKL	82.69
CD	80.98	CD	83.95

increased capacity of the head. While only minor fluctuations are observed, CCE tends to achieve slightly higher mIoU across both linear and MLP settings.

This observation highlights a key limitation of hard evaluation metrics: standard mIoU alone cannot reliably distinguish between different probabilistic training objectives. Although these losses impose fundamentally different constraints on the predicted distributions, such distinctions are largely lost once probabilistic outputs are reduced to hard predictions during evaluation. This motivates the use of soft evaluation metrics in soft-label segmentation settings.

6.4.3 Fine-Level Hard Supervision vs Coarse-Level Soft Supervision

Table 6.3 compares segmentation performance under three different training settings, including fine-level hard supervision, coarse-level hard supervision, and coarse-level soft supervision. CCE is used for soft supervision, as it was shown in the previous subsection to be sufficient among the reported soft-training losses when evaluated with standard mIoU. Standard CE was used to train the hard-label counterparts.

Although fine-level hard supervision achieves the best performance across both segmentation heads, the performance gap relative to coarse-level supervision remains small. Both hard and soft supervision at the coarse-level achieve mIoU values close to those obtained with fine-level labels, with differences of less than 1% in all settings. This indicates that despite reduced spatial resolution, coarse-level training retains most of the discriminative information required for accurate segmentation.

Table 6.3: Comparison of training with fine-level hard labels, coarse-level hard labels, and coarse-level soft labels. Fine-level supervision uses hard ground-truth labels at the original resolution and is trained with standard CE. Coarse-level supervision is obtained by downsampling the ground-truth labels using WAP, and is trained with CE loss for hard labels and CCE loss for soft labels.

Training Settings			**Linear Head**	**MLP Head**
GT Resolution	GT Type	Loss Function	mIoU (std)	mIoU (std)
Fine-level	Hard	CE	81.75	84.89
Coarse-level	Hard	CE	81.49	84.49
Coarse-level	Soft	CCE	81.33	84.37

6.4.4 Comparison of Soft IoU Relaxations

We now compare different relaxations of the soft IoU metric for evaluating probabilistic segmentation outputs. In this work, we consider four variants: the standard soft IoU, a symmetric-difference-based relaxation, and the proposed square-root and squared relaxations.

At the fine resolution, ground-truth annotations are hard, whereas soft IoU metrics operate directly on probabilistic predictions. Since the objective of this subsection is to compare IoU relaxations in a genuinely soft-label setting, we focus on reporting coarse-level soft-mIoU scores, where the ground truth is also probabilistic, although such metrics could similarly be computed at the fine resolution.

For reference, we additionally report standard mIoU at the fine resolution, computed from hard predictions. This provides an interpretable baseline that allows us to assess how closely different soft IoU relaxations align with the behavior of the standard IoU metric. In particular, this comparison highlights which relaxations exhibit stricter or looser notions of overlap relative to standard mIoU and whether they preserve the relative ranking of models observed under hard evaluation.

Table 6.4 compares different IoU relaxations using both fine-level and coarse-level training. For all coarse-level evaluations, including the fine-level training experiment, soft ground-truth masks are obtained using the proposed geometry-aware downsampling approach. In all experiments, a linear head is used to produce per-class predictions.

Table 6.4: Comparison of IoU relaxations for evaluating probabilistic segmentation outputs.

Training Settings			**Evaluation**				
GT Resolution	GT Type	Loss Function	Fine-level mIoU	Coarse-level soft mIoU			
			std	std	sd	sqrt	squared
Fine-level	Hard	CE	81.75	69.30	70.92	79.80	83.19
Coarse-level	Soft	CCE	81.33	71.45	72.75	81.33	83.39
Coarse-level	Soft	CE	80.87	67.64	69.59	79.21	83.25
Coarse-level	Soft	RKL	81.22	76.56	77.13	83.55	82.30
Coarse-level	Soft	CD	80.98	49.23	50.66	60.90	79.95

Considering the coarse-level soft mIoU scores, the square-root and squared relaxations consistently yield higher values than the standard soft IoU, indicating a less punitive treatment of uncertainty in probabilistic predictions. In particular, squared relaxation achieves the largest scores across most training settings, while the standard soft IoU produces the lowest scores, reflecting its stricter interpretation of mismatch between distributions.

Examining the relative ordering and separation induced by different relaxations reveals further differences in behavior. The squared relaxation maintains the strongest agreement with the ranking offered by standard fine-level mIoU, consistently identifying weaker training objectives, such as CD, as inferior while grouping stronger losses in a manner comparable to hard IoU evaluation. Moreover, it produces relatively small gaps among top-performing configurations and a clear separation from weaker ones, which reflects the narrow performance differences observed under standard mIoU.

The square-root relaxation also captures the general performance trend but yields larger score gaps between training configurations and introduces minor rank inversions among top-performing losses, such as favoring RKL over CCE. In contrast, stricter relaxations, including the standard soft IoU and the symmetric-difference-based variant, amplify performance differences, leading to large score gaps between training schemes and reduced alignment with the fine-level mIoU pattern.

Overall, these results suggest that while looser relaxations naturally produce higher absolute scores, the squared and square-root relaxations most faithfully preserve the relative ordering induced by standard mIoU, whereas stricter relaxations may distort this ordering by over-penalizing probabilistic mismatch.

Table 6.5: Quantitative comparison of void replacement approaches on the full PASCAL VOC training set using a linear segmentation head. Hard-label baselines are trained with the CE loss function, while CCE is used in soft-label settings.

Training-Set Partition	GT Void Replacement	mIoU (std)
Original	No VR	82.73
	Distance-Based VR	82.25
	Gaussian VR	82.68
Augmented	No VR	81.75
	Distance-Based VR	81.72
	Gaussian VR	81.61

6.4.5 Comparison of Void Replacement Strategies

We present a quantitative comparison of the proposed void replacement approaches on the PASCAL VOC dataset. Rather than aiming to demonstrate an overall performance improvement, these experiments are designed to evaluate whether replacing void labels with soft supervision leads to stable and interpretable behavior across different data conditions, particularly in regions affected by annotation ambiguity.

It is important to note that void annotations are present only in the original PASCAL VOC training set, which consists of 1,464 images. In contrast, the augmented training set contains a total of 10,582 images, of which the remaining 9,118 images do not include any void pixels. As a result, void replacement strategies can only impact the ground-truth for the original subset, while having no direct effect on the additional augmented images. To account for this distinction, we report results separately on the original and augmented partitions, allowing us to assess the behavior of void replacement both in isolation and in the presence of additional hard-labeled data. For each dataset partition, we compare three strategies: a standard hard-label baseline without void replacement (No VR), distance-based void replacement (Distance-Based VR), and Gaussian void replacement (Gaussian VR).

All experiments in this section are conducted using a linear segmentation head. Hard-label baselines are trained with the CE loss function, while all soft-label variants, including Distance-Based and Gaussian VR, are trained using CCE. For the Gaussian VR approach, the kernel size is fixed to 15 and the standard deviation is set to $\sigma = 2$ in all experiments.

Table 6.5 summarizes the quantitative comparison of different void replacement strate-

gies on the original and augmented PASCAL VOC training partitions. On the original partition, where void annotations are present, both proposed methods yield performance that is comparable to the hard-label baseline. Although hard supervision achieves the highest mIoU, the Gaussian VR approach closely matches this performance, indicating that introducing soft supervision in void regions does not substantially degrade segmentation quality. The Distance-Based variant exhibits a slightly larger drop in performance on the original partition, but still maintains competitive results relative to the hard-label baseline.

On the augmented partition, where the majority of images do not contain void pixels, all three strategies exhibit quite similar performance. This behavior is expected, as void replacement has no effect on the additional augmented samples. Consequently, most of the observed differences arise from indirect training effects, because only a small subset of images is directly modified by void replacement.

As discussed in earlier sections, void replacement strategies are primarily intended to address uncertainty near object boundaries and thin structures, rather than large void regions corresponding to missing or unlabeled objects. To isolate this effect, we introduce an additional setting in which masks with large void regions are excluded from training. We refer to this setting as the *Boundary-Focused* subset.

To construct the boundary-focused subset, we exclude masks whose void pixels form large regions that are far from any present semantic class. For each void pixel in a training mask, we compute its Euclidean distance to each semantic class present in the annotation, and obtain a per-pixel nearest-distance map by taking the minimum across classes. If the maximum of these distances exceeds a threshold T (set to 5 pixels in our experiments), the mask is classified as containing a bulk void region and is excluded. Applying this process removes 450 masks from the PASCAL VOC training set in both original and augmented partitions.

Table 6.6 reports results on the boundary-focused subset. In this setting, both void replacement strategies perform comparably to the hard-label baseline, suggesting that introducing soft supervision in void regions does not degrade segmentation quality and, in some cases, results in marginally higher mIoU.

Table 6.6: Quantitative comparison of void replacement approaches on the boundary-focused subset of PASCAL VOC, where large void regions are excluded. This setting primarily targets ambiguous object boundaries and thin structures. All experiments use a linear segmentation head. Hard-label baselines are trained with CE loss, while soft-label methods are trained using CCE.

Training-Set Partition	**GT Void Replacement**	**mIoU (std)**
Original (Boundary-Focused)	No VR	81.56
	Distance-Based VR	81.65
	Gaussian VR	81.62
Augmented (Boundary-Focused)	No VR	81.71
	Distance-Based VR	81.72
	Gaussian VR	81.53

6.5 Qualitative Results

6.5.1 Qualitative Analysis of Downsampling Approaches

We first provide a qualitative comparison of downsampling baselines on real semantic segmentation annotations from the PASCAL VOC dataset. In particular, we consider the following downsampling methods: nearest-neighbor, bilinear interpolation, area-based, AAP, and the proposed geometry-aware downsampling method.

Figure 6.1 shows representative PASCAL VOC ground-truth masks together with their downsampled counterparts at a lower spatial resolution. Nearest-neighbor downsampling exhibits clear artifacts, including jagged boundaries and the frequent disappearance of thin or small structures, as each coarse pixel is determined by a single sampled location. Bilinear interpolation alleviates some of these artifacts by incorporating local averaging; however, it often produces inconsistent boundary thickness and blurred class transitions.

In contrast, area-based downsampling, AAP, and the proposed method all aggregate information from multiple input pixels and therefore produce visually smoother and more stable coarse representations. On real PASCAL VOC annotations, which consist of large, contiguous regions, the outputs of these aggregation-based methods look highly similar. This similarity is expected, as uniform averaging already preserves the dominant semantic content in such cases. Subtle differences emerge near object boundaries and in regions containing thin or closely spaced structures.

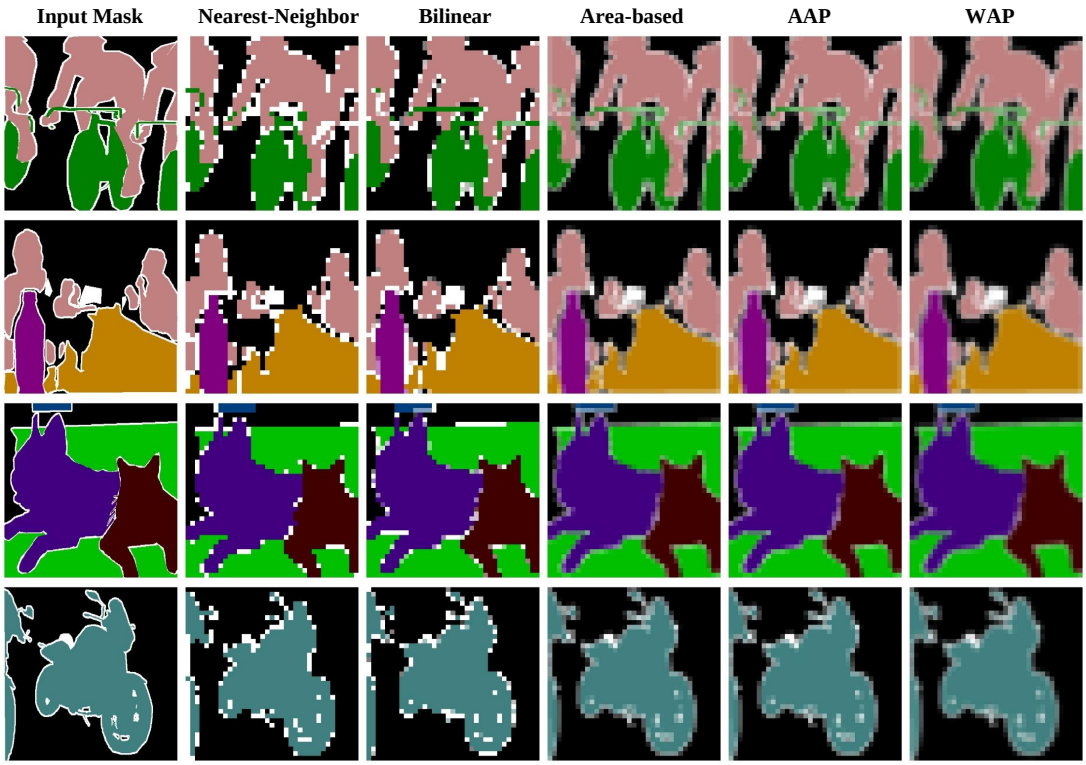


Figure 6.1: Qualitative comparison of downsampling approaches on PASCAL VOC ground-truth masks. All ground-truth masks are downscaled by a factor of 14 in each spatial dimension. Nearest-neighbor downsampling introduces jagged boundaries and often removes thin structures, while bilinear interpolation reduces some artifacts but remains sensitive to sampling. Area-based, AAP, and the proposed geometry-aware method produce smoother and more stable coarse representations on real-world annotations dominated by large contiguous regions, resulting in visually similar outputs. Subtle differences appear primarily near object boundaries and narrow objects.

While these differences may be subtle or difficult to perceive visually in natural segmentation masks, they become more apparent in controlled synthetic examples. In particular, Figures 6.2 and 6.3 illustrate two carefully designed scenarios in which the proposed downsampling strategy exhibits a qualitatively distinct behavior from area-based downsampling and AAP.

Figure 6.2 considers a synthetic example consisting of two adjacent thin vertical stripes belonging to different semantic classes, positioned near a coarse-cell boundary. Area-based

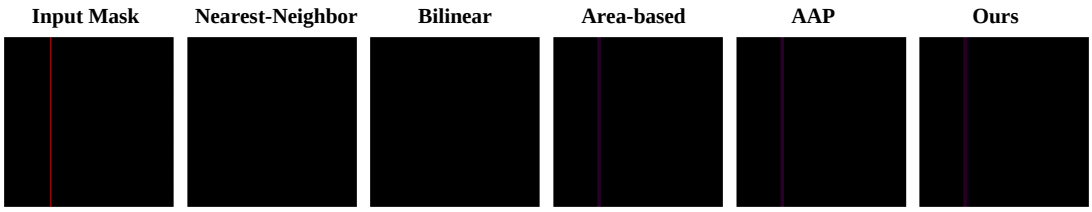


Figure 6.2: Qualitative comparison of downsampling approaches on a synthetic two-stripe pattern. The input mask contains two adjacent thin vertical stripes belonging to different classes (shown in red and blue), and is downsampled by a factor of 14 in each spatial dimension. Due to severe aliasing, nearest-neighbor and bilinear downsampling completely remove the stripes at the coarse resolution. Area-based downsampling and AAP collapse the two distinct stripes into a single coarse stripe by averaging labels within each bin. In contrast, the proposed geometry-aware downsampling preserves the separation between the two stripes, producing distinct soft labels that reflect their underlying spatial structure.

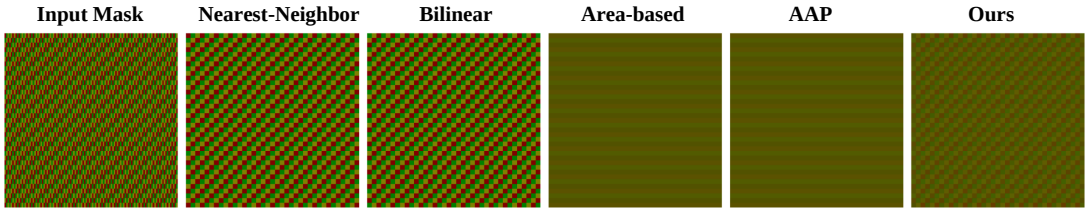


Figure 6.3: Qualitative comparison of downsampling approaches on a synthetic barcode pattern. The input mask consists of a grid of downsampling bins, each subdivided into asymmetric vertical regions belonging to three classes (shown in red, yellow and green), forming a barcode-like structure. Nearest-neighbor and bilinear downsampling yield overly sharp outputs, as they are dominated by local sampling. Area-based downsampling and AAP result in identical outputs by averaging class frequencies within each bin. This averaging removes the barcode structure entirely, producing vertically uniform soft labels and repeated horizontal bands. In contrast, the proposed geometry-aware downsampling responds to the internal spatial configuration of input mask, yielding spatially varying soft assignments that reflect geometric differences within and across bins.

downsampling and AAP aggregate labels uniformly within each coarse pixel, effectively collapsing the two distinct fine-level stripes into a single coarse stripe when both fall inside the same bin. As a result, the identity of the individual stripes is lost at the coarse resolution. In contrast, the proposed geometry-aware method distributes probability mass across neighboring coarse pixels according to continuous spatial weights, allowing the two stripes

to remain distinguishable in the downsampled representation. This example illustrates how the proposed approach preserves fine-scale geometric structure that is inherently discarded by either uniform averaging methods or local sampling approaches.

Figure 6.3 presents a synthetic barcode-like pattern, in which each coarse downsampling bin is subdivided into asymmetric vertical regions belonging to different classes. Area-based downsampling and AAP produce vertically identical outputs across bins, as these methods depend solely on aggregate class proportions and are invariant to the spatial ordering of labels within each bin. In contrast, the proposed geometry-aware approach responds to the internal spatial structure, resulting in spatially varying soft assignments and smooth transitions across neighboring bins.

6.5.2 Qualitative Comparison Across Training Strategies

Training with Different Downsampling Methods

Figure 6.4 presents a qualitative comparison of models trained using a linear head and different downsampling strategies for supervision, including nearest-neighbor, bilinear, area-based downsampling, and the proposed WAP approach. For each example, we show the original image, the fine-resolution ground-truth mask, and the corresponding predictions obtained by models trained under each strategy using soft supervision.

Consistent with the quantitative results reported in Table 6.1, nearest-neighbor downsampling leads to visibly weaker predictions, particularly in terms of object completeness. This behavior can be attributed to its reliance on local sampling, where each coarse location stores the label of a single fine-resolution pixel, rather than aggregating information from a neighborhood.

In contrast, bilinear downsampling, area-based downsampling, and WAP produce visually similar predictions across the examples. While minor differences can be observed in boundary smoothness and small structures, no clear qualitative advantage is evident among these methods, which aligns with their comparable mIoU scores.

Soft Supervision with Different Loss Functions

Figure 6.5 shows a qualitative comparison of predictions obtained using different loss functions for soft supervision at the training phase, all trained with an MLP segmentation head and using WAP for downsampling. While noticeable differences can be observed on individual images, the overall qualitative behavior across loss functions is almost similar. This

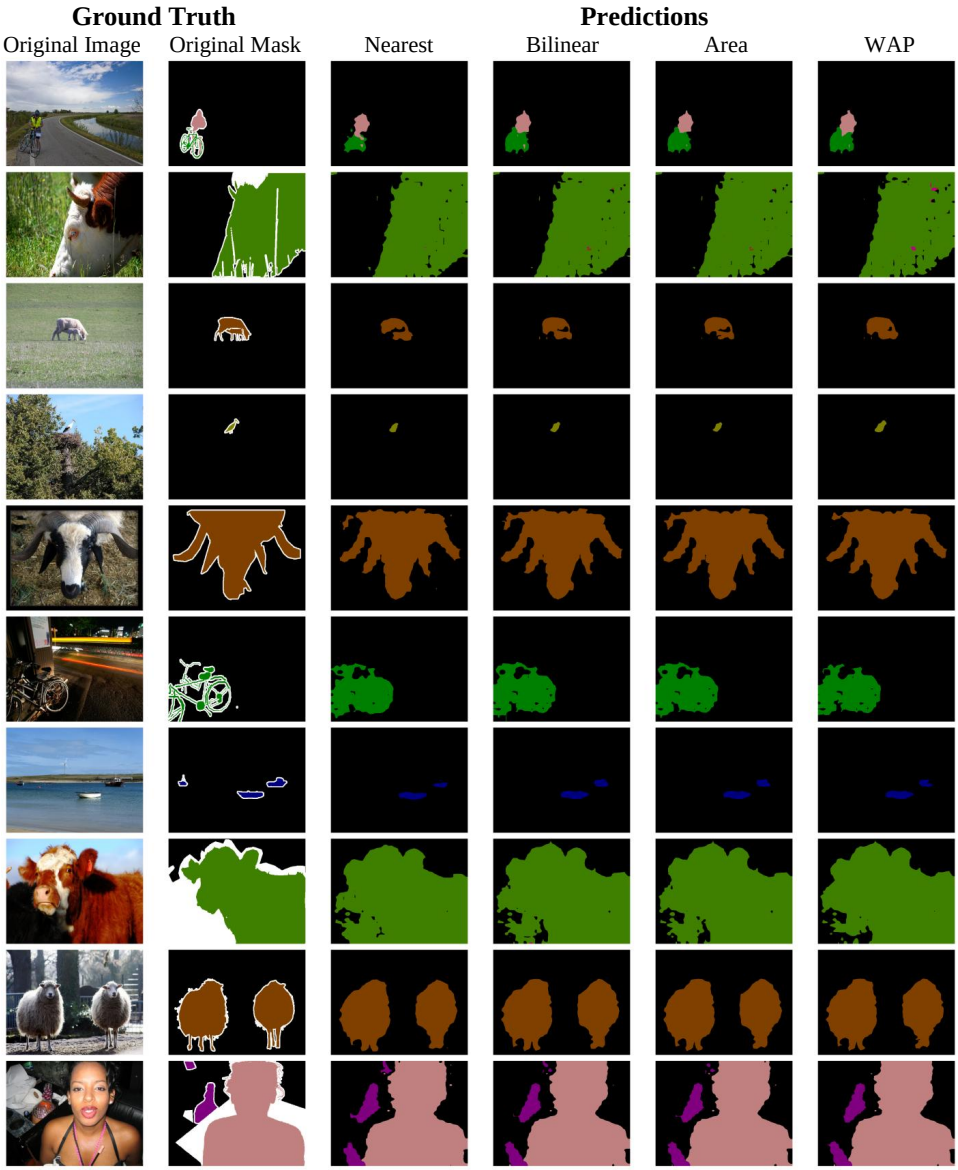


Figure 6.4: Qualitative comparison of approaches obtained by models trained with different downsampling strategies for soft-label supervision. From left to right: original image, ground-truth mask, and predictions for nearest-neighbor, bilinear, area-based downsampling, and the proposed WAP method.

is consistent with the quantitative results, which indicate comparable mIoU performance among most losses. In particular, CE and CCE yield slightly more stable and complete predictions, aligning with their stronger mIoU scores reported in Table 6.2, while RKL results in a weaker qualitative behavior in several cases.

Training at Different Levels of Resolution

Figure 6.6 presents a qualitative comparison of predictions obtained by training at different levels of resolution using linear and MLP segmentation heads. For each head, results are shown for fine-level supervision and coarse-level soft supervision (obtained via WAP downsampling).

Overall, the MLP head produces smoother and more spatially coherent predictions than the linear head across both resolution settings. This effect is particularly noticeable along object boundaries, where the MLP head tends to generate smoother contours. At the same time, the linear head can preserve sharper local details in certain cases and avoids excessive smoothing, resulting in competitive predictions for some examples despite its simpler architecture. Quantitatively, the MLP head achieves a higher mIoU than the linear head across resolution levels, in agreement with the observations in Section 6.4.3.

Regarding supervision resolution, fine-level training generally results in higher performance than coarse-level soft supervision. However, this trend is not consistent across all examples, and in some cases coarse-level soft supervision yields predictions that are comparable to those obtained using fine-level supervision.



Figure 6.5: Qualitative comparison of predictions obtained using different loss functions for soft supervision with an MLP segmentation head. From left to right: original image, ground-truth mask, and predictions for cross-entropy (CE), reverse KL divergence (RKL), collision divergence (CD), and collision cross-entropy (CCE). WAP downsampling was used in all experiments.

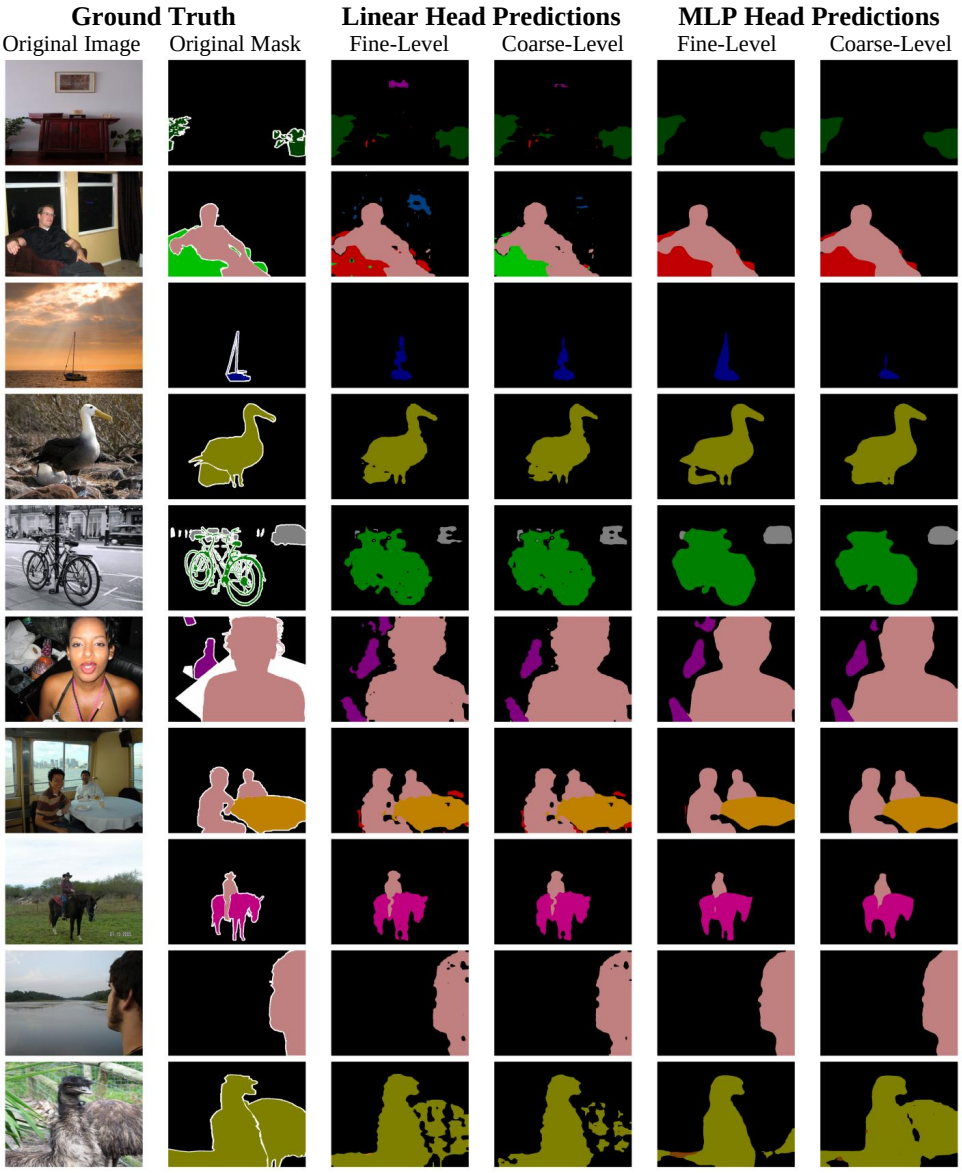


Figure 6.6: Qualitative comparison of predictions obtained by training at different resolution levels using linear and MLP segmentation heads. For each head, results are shown for fine-level supervision and coarse-level soft supervision obtained via WAP downsampling.

Chapter 7

Conclusion and Future Work

This thesis investigated the role of soft labels in semantic segmentation from a unified, end-to-end perspective, addressing label generation, supervision, and evaluation. To this end, we proposed WAP, a geometry-aware downsampling strategy for semantic segmentation masks that produces smooth, resolution-agnostic soft labels while preserving their probabilistic nature. In contrast to conventional approaches, the proposed WAP approach constructs spatially varying weights that explicitly model how fine-resolution pixels contribute to coarse-resolution supervision. This results in soft labels that more accurately capture the underlying spatial uncertainty, particularly around object boundaries and thin structures.

In parallel, we introduced consistent relaxations of the IoU metric for soft predictions and soft ground truth. These relaxations recover the standard IoU in the hard-label setting and achieve their maximum value under perfect agreement between prediction and ground truth. As a result, they provide a principled and interpretable evaluation protocol for probabilistic segmentation outputs.

In addition to soft-label generation via downsampling, this thesis also investigated strategies for handling annotation uncertainty encoded by void labels. Instead of discarding void pixels during training, we proposed void replacement methods that assign soft class distributions based on spatial context.

Through extensive experiments on the PASCAL VOC 2012 dataset, we systematically compared soft-label supervision with conventional hard-label pipelines across multiple training configurations. Although hard-label supervision achieves higher performance on average, the proposed soft-label framework, including both soft-label generation through downsampling and void replacement strategies, remains competitive and demonstrates

stable and consistent behavior across settings, confirming its conceptual soundness and practical feasibility.

A limitation of our work is the computational overhead introduced by the proposed WAP procedure. Compared to simple nearest-neighbor or bilinear downsampling, the construction of spatially varying aggregation weights requires additional computation. However, in the experimental setup considered in this thesis, the downsampling is applied as a pre-processing step to generate soft coarse targets and therefore does not affect training or inference time of the segmentation model. Nevertheless, this overhead may become more noticeable in experiments where soft label generation is performed on-the-fly or integrated directly into the training loop.

While this thesis explored multiple loss functions for training with soft labels and employed soft IoU relaxations for evaluation, an interesting direction for future work is to incorporate soft IoU directly into the training objective. Optimizing a differentiable loss based on consistent relaxations of soft IoU could better align training with the evaluation criterion. Additionally, evaluating the proposed framework in fully end-to-end fine-tuning and on larger-scale datasets would help assess its generality and practical impact. Since this thesis focuses on frozen backbone representations, understanding how soft supervision affects learned representations when the model is trained end-to-end remains an important open question. Finally, void replacement strategies could be extended beyond local spatial context, for example by introducing semantic priors to better model large ambiguous regions.

References

[1] 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.

[2] 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.

[3] 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.

[4] Serge Belongie, Chad Carson, Hayit Greenspan, and Jitendra Malik. Color-and texture-based image segmentation using em and its application to content-based image retrieval. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 675–682. IEEE, 1998.

[5] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23:1222–1239, 2002.

[6] Garrick Brazil, Xi Yin, and Xiaoming Liu. Illuminating pedestrians via simultaneous detection & segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 4950–4959, 2017.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[8] LC Chen, G Papandreou, I Kokkinos, K Murphy, and AL Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:834–848, 2017.

[9] 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.

[10] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[11] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1635–1643, 2015.

[12] 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.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[14] 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(2):303–338, 2010.

[15] 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.

[16] Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI brainlesion workshop*, pages 272–284. Springer, 2021.

[17] 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.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[19] Markus Hofmarcher, Thomas Unterthiner, José Arjona-Medina, Günter Klambauer, Sepp Hochreiter, and Bernhard Nessler. Visual scene understanding for autonomous driving using semantic segmentation. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 285–296. Springer, 2019.

[20] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[21] Talha Ilyas, Zubaer Ibna Mannan, Abbas Khan, Sami Azam, Hyongsuk Kim, and Friso De Boer. Tsfd-net: Tissue specific feature distillation network for nuclei segmentation and classification. *Neural Networks*, 151:1–15, 2022.

[22] Ying Jin, Jiaqi Wang, and Dahua Lin. Semi-supervised semantic segmentation via gentle teaching assistant. *Advances in Neural Information Processing Systems*, 35:2803–2816, 2022.

[23] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.

[24] 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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[25] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European conference on computer vision*, pages 695–711. Springer, 2016.

[26] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

[27] Xiaolong Li, Yuyin Li, Jinquan Ai, Zhaohan Shu, Jing Xia, and Yuanping Xia. Semantic segmentation of uav remote sensing images based on edge feature fusing and multi-level upsampling integrated with deeplabv3. *PloS one*, 18(1):e0279097, 2023.

[28] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[29] 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.

[30] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[31] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15305–15314, 2023.

[32] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 34:16331–16345, 2021.

[33] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33:978–994, 2010.

[34] Jie Liu, Yixiao Zhang, Jie-Neng Chen, Junfei Xiao, Yongyi Lu, Bennett A Landman, Yixuan Yuan, Alan Yuille, Yucheng Tang, and Zongwei Zhou. Clip-driven universal model for organ segmentation and tumor detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 21152–21164, 2023.

[35] 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.

[36] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7086–7096, 2022.

[37] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017.

[38] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8364–8375, 2022.

[39] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[40] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[41] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[43] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pages 234–244. Springer, 2016.

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[45] Mausoom Sarkar, Mayur Hemani, Rishabh Jain, Balaji Krishnamurthy, et al. Parameter efficient local implicit image function network for face segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20970–20980, 2023.

[46] Wei Shen, Zelin Peng, Xuehui Wang, Huayu Wang, Jiazhong Cen, Dongsheng Jiang, Lingxi Xie, Xiaokang Yang, and Qi Tian. A survey on label-efficient deep image segmentation: Bridging the gap between weak supervision and dense prediction. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):9284–9305, 2023.

[47] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.

[48] 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.

[49] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *International Workshop on Deep Learning in Medical Image Analysis*, pages 240–248. Springer, 2017.

[50] Boyuan Sun, Yuqi Yang, Le Zhang, Ming-Ming Cheng, and Qibin Hou. Corrmatch: Label propagation via correlation matching for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3097–3107, 2024.

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[52] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[53] Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse attend and segment: Unsupervised zero-shot segmentation using stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3554–3563, 2024.

[54] 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.

[55] Zifu Wang, Xuefei Ning, and Matthew Blaschko. Jaccard metric losses: Optimizing the jaccard index with soft labels. *Advances in Neural Information Processing Systems*, 36:75259–75285, 2023.

[56] Junde Wu, Ziyue Wang, Mingxuan Hong, Wei Ji, Huazhu Fu, Yanwu Xu, Min Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation. *Medical image analysis*, 102:103547, 2025.

[57] Xiongwei Wu, Doyen Sahoo, and Steven CH Hoi. Recent advances in deep learning for object detection. *Neurocomputing*, 396:39–64, 2020.

[58] Yongyang Xu, Liang Wu, Zhong Xie, and Zhanlong Chen. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. *Remote Sensing*, 10(1):144, 2018.

[59] Hongjia Zhai, Gan Huang, Qirui Hu, Guanglin Li, Hujun Bao, and Guofeng Zhang. Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[60] Zhongwen Zhang and Yuri Boykov. Collision cross-entropy for soft class labels and deep clustering. *arXiv preprint arXiv:2303.07321*, 2023.

[61] Zhongwen Zhang and Yuri Boykov. Soft self-labeling and potts relaxations for weakly-supervised segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20244–20253, 2025.

[62] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[63] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[64] 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.

[65] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[66] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European conference on computer vision*, pages 696–712. Springer, 2022.

[67] Helong Zhou and Liangchen Song. Rethinking soft labels for knowledge distillation: A bias–variance tradeoff perspective. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[68] Kunlin Zou, Xin Chen, Yonglin Wang, Chunlong Zhang, and Fan Zhang. A modified u-net with a specific data argumentation method for semantic segmentation of weed images in the field. *Computers and Electronics in Agriculture*, 187:106242, 2021.

[69] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*, 2020.