KU LEUVEN

FACULTY OF
ENGINEERING SCIENCE

Think before you Learn: Image Segmentation with Weak Supervision

Andrei-Bogdan Florea

Thesis submitted for the degree of
Master of Science in Engineering:
Computer Science, option Artificial
Intelligence

Supervisor:
Prof. Dr. Luc de Raedt

Assessor:
Ir. Adem Kikaj
Dr. Jose Manuel Alvarez

Assistant-supervisor:
Ir. Jaron Maene

Academic year 2024 – 2025

Preface

This thesis has been a valuable opportunity to immerse myself more deeply in academic research. I have devoted considerable time and effort to this work, learning to appreciate not only the outcomes but also the process itself. As one of the most extensive technical projects that I have worked on, this thesis challenged me to integrate my prior knowledge of image segmentation with new concepts from the emerging field of neurosymbolic AI.

I would like to express my gratitude to my promoter, Prof. Dr. Luc de Raedt, for his insightful feedback provided during the presentations. I hope that this thesis reflects the expectations set during our discussions. I am also deeply thankful to my supervisor, Jaron Maene, who has consistently offered his guidance and support throughout the academic year. His advice helped me overcome challenges and define the direction of this research.

Andrei-Bogdan Florea

Contents

Abstract

Image segmentation is a core task of computer vision, enabling a detailed understanding of the scene. Pixel-level annotations are essential for training image segmentation models, but producing such detailed labels is both time-consuming and expensive. This thesis investigates an alternative path, weakly supervised semantic segmentation (WSSS), which involves training a segmentation model using indirect forms of supervision that are significantly cheaper to obtain.

Rather than designing separate solutions for each type of supervision, the central objective of this thesis is to create a unified framework that enables learning from various forms of weak supervision, including classic weak labels (bounding boxes, scribbles, points, and image-level tags) and background knowledge. To achieve this, the proposed method uses a fuzzy logic-based approach to reason in a structured and scalable manner.

More concretely, the framework fine-tunes a foundational vision model, Segment Anything (SAM), to maximally satisfy a set of logical constraints derived from the available weak forms of supervision. Despite its segmentation accuracy, SAM is not suitable for inference on new images because it relies on prompts. To address this, the framework includes a subsequent stage where it trains a segmentation network in a fully supervised manner using pseudo-labels produced by SAM. This network is the desired outcome of WSSS, achieving state-of-the-art results on two datasets: Pascal VOC 2012 and LGG segmentation, with mIoU scores of 87.9% and 73.6%, respectively.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations

Abbreviations

WSSS	Weakly supervised semantic segmentation
SAM	Segment Anything model [41]
CAM	Class activation map [100]
mIoU	Mean Intersection over Union
MIL	Multiple instance learning
WMC	Weighted model counting
WFOMC	Weighted first-order model counting
NLP	Natural language processing
CNN	Convolutional neural network
ViT	Vision Transformer [23]
CRF	Conditional Random Field
CLIP	Conditional Language-Image Pre-Training model [68]
FOL	First-order Logic
LTN	Logic Tensor Network [2]
LGG	Low-grade glioma

Chapter 1

Introduction

This first chapter sets the foundation for the development of this thesis. It begins by establishing the context surrounding this work. It then specifies the problem addressed and, thus, the motivation behind this research. It briefly introduces the related, pre-existing approaches and then describes the proposed solution, emphasizing the main contributions of this work. This chapter concludes by outlining the structure of the subsequent chapters.

1.1 Context

Over the last decade, there have been tremendous advances and research efforts in the field of machine learning, particularly in deep learning [73]. These have extended into computer vision, which involves image analysis, where the availability and scale of data have powered the development of various technologies [56, 41]. More specifically, large volumes of high-quality data enable training accurate models. With full supervision, training requires access to target labels, which annotators typically create manually or with the help of semi-automated tools that still require human input [6]. However, this is a time-consuming procedure, as it often involves annotating fine-grained details or recognizing variations between classes that are hard to distinguish [98].

Additionally, the success of a firm specializing in machine learning technologies significantly depends on the quality and availability of data resources. When publicly available data does not fit the business requirements, the firm may opt to collect proprietary data [47]. For a wide range of tasks in supervised learning, this process not only consists of acquiring raw data (e.g., images) but also labeling it. This labeling phase accounts for a significant portion of the costs, which can be either time-related (requiring a person to perform the annotation manually) or hiring costs (requiring external expertise) [96]. Reducing time-related costs may increase efficiency and, therefore, decrease the time-to-market of the solution.

Figure 1.1: Segmentation map of an image from the Cityscapes dataset [17].

1.2 Problem

Image segmentation involves assigning a class label to each pixel of an image. For instance, Figure 1.1 shows the expected result of segmenting an image from the Cityscapes dataset [17]. As mentioned in the previous section, a significant downside of the fully supervised approach to semantic segmentation is the high annotation cost: pixel-wise labels require extensive (manual) effort, with the annotation time extending to several minutes per image [98]. There are tasks such as 3D segmentation, where labeling a single image can take several hours [99].

In contrast, in a weakly supervised setting, annotators provide *weak labels*, which require less annotation effort than pixel-wise labels and only take seconds to perform [1]. In weakly supervised semantic segmentation (WSSS), the task tackled by this thesis, examples of such higher-level labels include image-level tags ("the image contains a car"), bounding boxes ("there is a person in this bounding box"), scribbles ("these pixels belong to an airplane"), or points ("this point belongs to a dog").

Naturally, weak supervision aims to extract as much information as possible from weak labels, with the hope of approaching the performance of full supervision. Thus, a weakly supervised paradigm requires careful considering which labels to learn from and how to design the learning process to minimize this performance gap. These questions make the problem intrinsically hard.

1.3 Related work

Many works on weak supervision and WSSS, in particular, have been proposed over the years, demonstrating that this topic is an active area of research.

Researchers often focus on exploiting a single type of weak label, typically one of the ones previously mentioned. The approaches in this direction are diverse, ranging from using class activation maps (CAMs) [100, 34] when only having access to image-level class labels to a paradigm of propagating information to unknown regions in the case of scribbles [66]. All of these approaches fall into a category

of specialized, ad-hoc solutions, as they narrow their focus on one type of weak supervision, therefore being inflexible.

A few other works learn from multiple types of weak labels during the training procedure. Xu et al. [90] employ a max-margin clustering framework to determine the optimal assignment of class labels, with constraints formulated based on the typical weak labels. Moreover, Ke et al. [37] utilize the same weak labels and propose a method for encoding pixels into a latent space, where a contrastive learning approach guides the pixel representation. While these methods can learn from the most significant types of weak labels, it is not clear how to extend them beyond classic weak labels.

As another weak supervision setting, prompting a foundational segmentation model such as the Segment Anything Model (SAM) [41] has become an important area of research recently. SAM is capable of producing unlabeled segmentation masks for various tasks by utilizing points or bounding boxes as additional inputs. Starting from a text prompt with all classes of interest, Sun et al. [77] managed to outperform previous weakly supervised approaches by passing the text prompt through Grounded DINO [53] to obtain bounding boxes, and later feeding the bounding boxes as prompts to SAM. Other approaches [94, 49] guide the segmentation model through learnable prompts rather than relying on handcrafted text prompts. While these approaches yield impressive results, relying solely on the pre-trained segmentation model implies that performance is reduced in scenarios or domains that are underrepresented in the dataset on which the model was trained. For example, applying SAM on X-ray images [19] yields strong results only after fine-tuning the base model. In addition, a classic, weakly supervised pipeline may be preferred over the use of large foundational models in applications constrained by available computing resources or time [77].

1.4 Approach

This thesis follows a conventional two-stage paradigm commonly used in the WSSS literature. The process begins by using a weakly supervised method to produce pseudo-segmentation labels for the training images. These serve as (fully annotated) target labels for training a segmentation model in the second stage. The network obtained through this process constitutes the desired result of WSSS.

Where this method distinguishes itself from other approaches is in the first stage, which consists of a weakly supervised fine-tuning of the Segment Anything model. Instead of exploiting a single type of weak label for producing pseudo-labels, this thesis aims to effectively learn from multiple types of weak labels within a single, unified framework. It achieves this by expressing logical constraints based on the annotated weak labels and other prior knowledge. The framework converts these expressions into differentiable learning signals (loss functions) used for training SAM. This probabilistic view is not novel for image classification, where Shukla et al. [74] introduce a dynamic programming-based algorithm for computing the probabilities required for training a classifier under various forms of weak supervision. However, the segmentation task has received less attention, likely due to the daunting challenge

of working in such a high-dimensional space.

Apart from providing a way to learn from the classic weak labels, the framework developed in this thesis offers more flexibility in the types of signals that the neural network can learn from. This method enables learning from prior knowledge expressed as logical constraints, serving as a form of higher-level signals (e.g., "if there is a TV in the image, then there is also a table"). Such flexibility is also of great use outside academia, as it provides a way to iteratively correct situations where a neural network performs poorly.

1.5 Results

Evaluation proceeds on two datasets: Pascal VOC 2012 [24] and LGG segmentation [64]. The former is a standard benchmark for WSSS. At the same time, the latter demonstrates the potential for applying the framework in the medical domain, specifically for brain tumor segmentation. For a fair comparison to other works, the second stage of the process involves training multiple segmentation networks: Mask2Former [15], DeepLabV2 [11], and ConvNeXt-UPerNet [55, 87].

By training with qualitative pseudo-labels produced by SAM in the first phase, the networks achieve state-of-the-art performance in terms of the mean intersection over union (mIoU). On the Pascal VOC *test* set, Mask2Former reaches 87.9% mIoU, while DeepLabV2 with a ResNet101 backbone produces a result of 78.2% mIoU, improved to 79.6% with the Conditional Random Field post-processing available for DeepLabV2. Similarly, the second-stage training of a ConvNeXt-UPerNet network achieves 73.6% IoU for the tumor class on the LGG dataset, outperforming other WSSS methods.

The key factors that contribute to such results are (1) leveraging SAM as a strong baseline for segmentation, thus outperforming pre-SAM methods, and (2) benefiting from the most amount of information during training. While previous works learn from a single weak label type (image tags, points, bounding boxes, or scribbles), this method benefits from both bounding box and scribble annotations.

1.6 Use of generative AI

Generative AI tools contributed to the clarity and correctness of the text in this thesis during the writing process. This thesis used ChatGPT to assist with rephrasing the text for improved readability and to draft LaTeX-formatted equations. Additionally, it employed Grammarly to help correct grammatical errors and refine the overall clarity of the text. These tools supported the writing process while keeping the originality of the content.

1.7 Thesis structure

The following chapter introduces the key concepts used in this work, thus providing the essential background information for understanding this topic.

Chapter 3 presents the literature review, which explores the two primary research directions, semantic segmentation and weak supervision, and examines their integration. Additionally, this chapter examines works that effectively learn from logic through constraints.

Chapter 4 further expands upon the problem statement and establishes the objectives for this thesis. Chapter 5 presents an in-depth exploration of the proposed solution for the problem above. Chapter 6 evaluates the solution and examines the results in comparison with other works. It does so from various perspectives, building an argument for the relevance of this work. Finally, chapter 7 summarizes the findings and provides a critical assessment of whether the goals of this work were reached. It also mentions the limitations of the current approach and how further research can address these challenges.

Chapter 2

Background

This chapter provides definitions and explanations for the concepts used in this study. These include notions related to image segmentation, types of supervision, types of weak labels, logic, and the probabilistic and fuzzy extensions of logic. This part assumes that the reader has a basic understanding of deep learning, neural networks, and first-order logic.

2.1 Image segmentation

Image segmentation is a core challenge of computer vision. As mentioned previously, it consists of classifying images on a per-pixel basis. Despite having such a concise definition, multiple forms of image segmentation exist.

Semantic segmentation. Semantic segmentation assigns each pixel in an image a label corresponding to a *class* category [18], such as *pedestrian*, *car*, or *road*. It does not distinguish between different objects of the same class.

Instance segmentation. Instance segmentation aims to identify and distinguish individual *objects* [60] by labeling each pixel with the object it is a part of. However, the objects themselves do not receive a class label.

Panoptic segmentation. Panoptic segmentation [40] generalizes the previous tasks by assigning semantic class labels to pixels while also distinguishing individual instances.

Figure 2.1 displays an overview of the three tasks of image segmentation, as presented by Kirillov et al. [40]. Although this thesis focuses on semantic segmentation, the method discussed is not limited to this specific task. The framework allows extensions for tasks such as instance or panoptic segmentation.

2.2 Types of supervision

Over the years, machine learning researchers have proposed various methods for learning from data, most of which aim to alleviate the heavy annotation burden

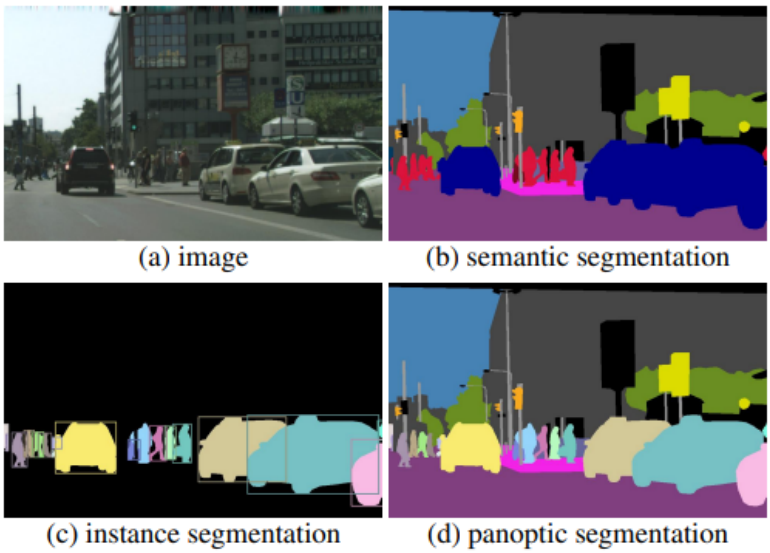


Figure 2.1: Overview of the Image Segmentation tasks, as presented by Kirillov et al. [40].

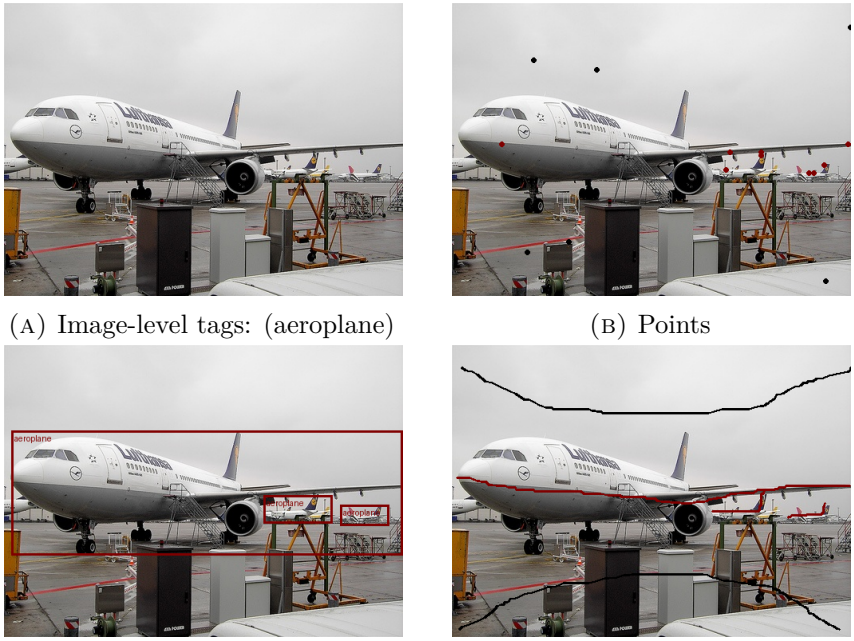
associated with fully supervised approaches. The part below outlines the learning paradigms relevant to this work.

Fully supervised learning corresponds to training a model on labeled datasets containing both raw inputs and corresponding target outputs. In the context of semantic segmentation, this means each data point consists of an image along with its per-pixel class labels.

Weakly supervised learning is a paradigm in which models learn from indirect forms of supervision. Unlike full supervision, where the training data corresponds to the exact output the model is expected to produce, weakly supervised methods rely on less informative but easier-to-obtain labels [66]. It is also important to distinguish weakly supervised approaches from *semi-supervised learning* [67], which involves learning from a small labeled dataset and a (larger) set of unlabeled examples. This setting is not in the scope of this thesis. An example of weak supervision applied in the context of binary classification is Multiple Instance Learning (MIL), where labels are available at a bag level rather than an instance level and indicate whether the bag contains the positive class [74].

Weakly supervised semantic segmentation (WSSS) is an application of the weakly supervised methodology in the context of segmentation. These methods exploit information from one (or more) indirect labels and aim to produce accurate segmentation maps of the input images [66]. The following section provides a detailed discussion of the most commonly used types of weak labels and also introduces other forms of indirect supervision.

Sparsely annotated semantic segmentation [48] refers to approaches that



(A) Image-level tags: (aeroplane)

(B) Points

(C) Bounding boxes

(D) Scribbles

FIGURE 2.2: Main types of weak labels in WSSS.

rely solely on sparse (or partial) labels, such as points or scribbles. It is, thus, a subcategory within the broader field of WSSS. The work here is not limited to these types of weak labels, but many works in this subfield relate to the problem discussed.

2.3 Types of weak labels

Figure 2.2 displays the four main types of weak labels in WSSS.

Image-level tags indicate the classes present in the image. One such example is Figure 2.2a, where "aeroplane" is the only present tag. By default, the background is part of the present classes.

Points offer the class label of one or more pixels in the image. As visualized in Figure 2.2b, they are easy to annotate and provide localization on top of the image-level tag annotation.

Scribbles are collections of points easily drawn by annotators. They are a form of partial labels, as shown in Figure 2.2d, offering a trade-off between the sparsity of point-level supervision and the high labeling costs associated with fully annotated images. A scribble also contains an annotation for the class that the set of points corresponds to.

Bounding boxes represent rectangles drawn around objects and also provide the class labels of the objects localized within their boundaries. In this thesis, the

convention used to define a bounding box is based on its upper-left and bottom-right corners, denoted as (y_{min}, x_{min}) and (y_{max}, x_{max}), respectively.

Less common indirect forms of supervision may include volume annotations ("at least half of the image contains dogs") [38], shape priors [25], or other domain-specific knowledge in the form of logical expressions ("if the sun is present in an image, it is above all other objects").

2.4 Logic

2.4.1 Classical Logic

As this thesis employs a predicate (or first-order) fuzzy logic for expressing and learning from constraints, it is essential to first establish the terminology of *classical (predicate) logic*. It consists of terms and formulas. *Terms* are either constants (e.g. *Airplane*), variables (e.g. y_2) or functions $f(t_1, ..., t_n)$ where $t_1, ..., t_n$ are terms. A *formula* can be either an atom predicate (e.g., $Box(x_{min}, y_{min}, x_{max}, y_{max}, c)$) when it expresses a relation between terms or can be arbitrarily nested by joining other formulas with *logical connectives*. These, in turn, are the negation $\neg$, conjunction $\wedge$, disjunction $\vee$, implication $\Rightarrow$, equivalence $\Leftrightarrow$, as well as quantifiers $\exists$ and $\forall$.

2.4.2 Probabilistic logic

At the basis of probabilistic reasoning stands the weighted model counting (WMC) problem [9], useful to compute the success probability of a query formula. WMC typically operates in a propositional logic setting over a set Lit of literals which represent variables or their negation. An *interpretation* or a *structure* $\mathfrak{A}_{Lit}$ denotes the truth assignment to those literals. A *model* of a formula ψ is an interpretation $\mathfrak{A}_{Lit}$ where the truth assignment of variables leads to the satisfaction of ψ, denoted as $\mathfrak{A}_{Lit} \models \psi$. $Mod(\psi)$ is the set of models of the formula ψ. Given a propositional formula ψ and a weight function w, WMC computes:

$$WMC(\psi, w) = \sum_{\mathfrak{A}_{Lit} \in Mod(\psi)} \prod_{l \in \mathfrak{A}_{Lit}} w(l), \tag{2.1}$$

where w is a function $w\colon Lit \to \mathbb{R}$ that maps every literal to a real number representing its weight. In probabilistic logic, the weight function follows the probability distribution of the variables. Thus, every variable v has an associated probability of success p and therefore, $w(v) = p$ and $w(\neg v) = 1 - p$.

Extending this computation to first-order logic formulas with finite domains typically involves a *grounding* operation that instantiates a formula with all combinations of domain elements. Quantified formulas $\forall x\ \psi$ and $\exists x\ \psi$ in particular are respectively grounded to a conjunction and a disjunction between the instantiations of ψ. Grounding produces a propositional formula to which WMC applies directly.

Alternatively, computing the success probability of a first-order formula ψ reduces to the weighted first-order model counting problem (WFOMC) [44]. Interpretations

$\mathfrak{A}_A$ here denote truth assignments to the set of atoms and their negation, while the function w maps a predicate or its negation to a real number:

$$WMC(\psi, w) = \sum_{\mathfrak{A}_A \in Mod(\psi)} \prod_{P \in \mathfrak{A}_A} w(P) \prod_{\neg P \in \mathfrak{A}_A} w(\neg P). \tag{2.2}$$

2.4.3 Fuzzy logic

Fuzzy logic is an approximate reasoning paradigm where the truth values of variables correspond to real numbers in the interval $[0, 1]$ [79], varying from false to true. In first-order fuzzy logic, predicates receive truth values instead [79].

Fuzzy logic replaces the boolean operators for logical connectives with continuous functions: negation corresponds to an operator N; a fuzzy t-norm T generalizes conjunction, a fuzzy t-conorm S generalizes disjunction, and the material implication corresponds to a fuzzy implication I [79]. As an example, a fuzzy negation is $N(x) = 1 - x$, a fuzzy t-norm is the product t-norm $T(x, y) = x \cdot y$ [27]. Fuzzy t-conorms typically use De Morgan's law to define $S(x, y) = N(T(N(x), N(y)))$ [79]. First-order fuzzy logic also defines fuzzy equivalents for the universal and existential quantifiers [2]. These are aggregation operators A^U and A^E, respectively, which are non-decreasing functions having all possible groundings of the subformula as arguments.

Inference in a fuzzy logic context implies obtaining the truth degree (or satisfaction) of a query formula ψ by recursively applying the fuzzy operators that construct ψ. Then, considering a function w that assigns truth values to predicates, the truth degree V of a formula ψ as presented by Krieken et al. [79] follows this inductive definition on the structure of ψ:

$$\begin{aligned}
V(P(t_1, \ldots, t_n)) &= w(P(t_1, \ldots, t_n)) \\
V(\neg\psi) &= N(V(\psi)) \\
V(\psi \wedge \phi) &= T(V(\psi), V(\phi)) \\
V(\psi \vee \phi) &= S(V(\psi), V(\phi)) \\
V(\psi \rightarrow \phi) &= I(V(\psi), V(\phi)) \\
V(\forall x\ \psi) &= A^U(V(\psi[x = x_1]), \ldots, V(\psi[x = x_n])) \\
V(\exists x\ \psi) &= A^E(V(\psi[x = x_1]), \ldots, V(\psi[x = x_n])),
\end{aligned} \tag{2.3}$$

where $x_1, \ldots, x_n$ denote all possible groundings of x, and the notation $\psi[x = x_k]$ refers to the formula ψ with the bounded variable x substituted by the ground term x_k.

The time complexity of evaluating a formula in fuzzy logic scales linearly in the number of predicates and logical connectives used in the formula, but exponentially with the number of nested quantifiers [79] because the aggregation operators iteratively scan the domain of objects for each quantifier encountered.

Chapter 3

Related work

This chapter explores key areas of research related to this study. It begins by highlighting recent advances in semantic segmentation. Next, it connects these approaches to techniques that apply weak supervision in the problem of semantic segmentation. Finally, it introduces methods that train neural networks using signals derived from logic. The analysis provides a detailed examination of each technique, highlighting its advantages and potential limitations.

3.1 Semantic segmentation

Over time, semantic segmentation has been a significant topic of interest in the field of computer vision [18, 60]. Its application domains are vast, receiving attention in industries such as healthcare [81, 85], agriculture [57], while also enabling self-driving vehicles [7]. The advances in deep learning, in particular, have translated into more accurate segmentation methods.

Among the earlier deep learning-based methods that pioneered the use of an encoder-decoder convolutional neural network (CNN) [56] are SegNet [4] and U-Net [72]. The encoder specializes in extracting image features, while the decoder sequentially upscales the intermediary results to the desired output resolution of the segmentation mask, receiving additional signals from the encoder at every upscaling layer.

The DeepLab family of models [11, 12, 13] have set important benchmarks in the problem of semantic segmentation. Built using the ResNet [30] backbone as an encoder, they enable the use of a much deeper network. Additionally, other improvements made to the architecture, including the use of atrous spatial pyramid pooling and the addition of an edge refinement module, resulted in increased segmentation accuracy.

With their success in natural language processing (NLP) [80], Transformer models have recently been introduced in computer vision pipelines [15, 101]. The primary issue with adopting Transformers is that computing self-attention scales quadratically with the sequence length [80]. While this is less of a problem in NLP tasks, computing self-attention in the case of images scales quadratically in

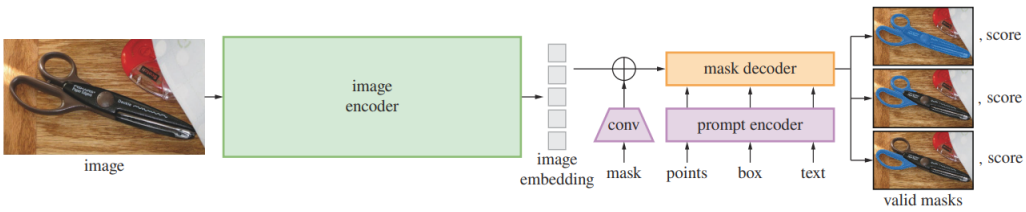


Figure 3.1: The architecture of SAM, proposed by Kirillov et al. [41].

the number of pixels [23], which is infeasible in practice. The Vision Transformer (ViT) [23] and the Swin Transformer [54] address this issue by attending to image patches rather than individual image pixels. While these two works do not tackle segmentation in particular, methods such as Mask2Former [15] use transformers to achieve state-of-the-art performance in all segmentation tasks (semantic, instance, and panoptic) on the COCO dataset [51] at its publishing time.

The techniques above aim to train a promptless, closed-set segmentation network (i.e., a network that receives a single image and produces a segmentation map with class labels among a known set of classes). Recently, however, there has been a shift towards promptable segmentation [101]. The latter represents a collection of methods based on Foundational Models, which are networks of immense scale trained on massive amounts of data, exhibiting powerful generalization potential. In segmentation, zero-shot methods such as Segment Anything (SAM) [41] demonstrate this. SAM can produce a meaningful segmentation given an interactive prompt, such as a bounding box or a point. As shown in the model architecture proposed by Kirillov et al. [41] and displayed in Figure 3.1, it achieves this by encoding both the image and the prompts using Transformer models and combining them in a Transformer-based decoder used for generating the segmentation masks. Although the quality of the results is high for such an open-set method, SAM does not directly solve WSSS. In contrast, this task expects to produce a trained *promptless* segmentation network that performs inference on unseen images without additional prompts.

3.2 Weakly supervised semantic segmentation

Although the previous section focused on fully supervised semantic segmentation, this thesis mainly contributes to WSSS. This section is an in-depth analysis of the evolution of WSSS along its research paths. More specifically, this section follows related methods and groups them by the type of weak annotations they exploit.

Before exploring the related works individually, it is essential to note that almost all methods employ a two-stage procedure. The first step involves extracting pseudo-segmentation labels for training images before using them as ground-truth data in a second stage training.

3.2.1 Bounding boxes

Among the earlier methods that use bounding boxes as weak annotations is DeepCut [69], which employs a two-step iterative procedure similar to the Expectation-Maximization (EM) algorithm [62]: an E step which labels pixels based on the current model and an M step which updates the model. The model itself consists of a CNN and a Conditional Random Field (CRF) [10], where the CNN assigns per-pixel labels and the CRF imposes appearance and smoothness constraints on pixel neighborhoods. While the method does not benefit from the segmentation accuracy of recent models, considering pixel neighborhoods may still be relevant in the context of WSSS at present.

Further along, Kervadec et al. [39] solve a constrained optimization problem that aims to satisfy the bounding box *tightness constraint* [46] while optimizing for a *global emptiness constraint*. The tightness constraint assumes that the target object is perfectly framed within the bounding box, and thus, every row and column has a pixel of the given class. On the other hand, the global emptiness constraint models the outer region as background, enforcing that there are no pixels of the target class outside of the bounding box.

Ji et al. [31] utilize bounding box annotations to isolate objects of interest and create one dataset per class. Each such dataset contains a single class, so the method employs a WSSS technique based on image-level tags to segment that class. After producing pseudo-labels for individual objects, the method stitches them back into the initial images at places corresponding to the bounding box annotations, exploring multiple heuristics to handle potential overlaps.

The method proposed by Wang et al. [82] improves on the work of Kervadec et al. [39] by tackling its main limitation, namely the requirement of tight bounding box annotations. Instead, it learns from loose bounding box annotations by enforcing the same Multiple Instance Learning (MIL) on polar lines in the bounding box rather than axis-parallel lines.

A recent direction of research has been the use of large foundational models [101] in the context of WSSS. In this sense, Jiang et al. [32] explore simply prompting SAM [41] with the bounding box annotations of the Pascal VOC 2012 dataset [24] to obtain pseudo-labels. Then, the method performs fully supervised training in the second stage using the DeepLabV2 segmentation network. While this approach serves as a baseline and demonstrates the zero-shot capabilities of SAM, it remains unclear whether fine-tuning SAM in a weakly supervised manner can improve its performance.

3.2.2 Scribbles

Early approaches focused on ways of propagating the scribble-based sparse annotations to the rest of the image. For instance, ScribbleSup [50] divides the images into superpixels and transforms the problem of pixel-wise segmentation into a classification of superpixels. It optimizes an objective function composed of a unary term that models the class of each super-pixel and a binary term that models the

similarity of two adjacent superpixels. A scribble contributes to the class assignment of superpixels, while unlabeled superpixels receive information from neighboring superpixels.

Other methods aim to propagate the information contained in the scribbles iteratively. AGMM [86] proposes a network with a segmentation head and a Gaussian mixture model (GMM) head. Besides the (partial) cross-entropy loss provided by the scribbles, the predicted segmentation mask receives self-supervision from the GMM predictions. TEL [48] is another coarse-to-fine method that represents the image as a minimum spanning tree and trains a segmentation network via an additional branch that models the interaction of color and semantic information of pixels across the MST.

In Scribble hides class [97], the authors perform a joint task of classification and segmentation. The classifier extracts class activation maps (CAM) [100], interpreted as the most discriminative features of the classified objects. The segmentation network receives localization hints from such CAMs through the localization rectification module proposed.

Chan et al. [8] discuss using features of the classified pixels in training to expand scribbles to unknown regions. In this sense, the method extracts local and global prototypes from intermediate feature maps. These are part of an augmentation procedure, where a loss term enforces consistency between the original predictions and the predictions obtained from a feature map augmented with prototypes.

Besides the use of bounding boxes for prompting, the work of Jiang et al. [32] also benchmarks SAM when prompted with scribbles. More precisely, they sample 20% of the points in each scribble and use them as prompts for SAM. The result is similar to the one obtained by prompting SAM with bounding boxes.

3.2.3 Points

Point-based methods are a more active area of research in interactive segmentation [76, 92]. Despite this, several methods have still been proposed for the problem of WSSS.

Bearman et al. [5] tackle the sparsity of the annotations by jointly optimizing both the cross-entropy loss for the labeled points and a loss based on an objectness prior. The latter involves the probability that any point is part of an object and encourages pixels to be classified as background when that probability is low or encourages pixels to be one of the 20 object classes of Pascal VOC otherwise.

PCAM [59] primarily uses class activation maps. While initially intended for image-level supervision, PCAM exploits the localization aspect of points to improve predictions. The technique adds a point-supervised term of the loss function applied to both the classification branch and the CAM-generating branch.

Again, Jiang et al. [32] test the effectiveness of SAM in a scenario where it receives a single point per object instance as prompt. They confirm the assumption that less informative labels lead to a worse performance by empirical evidence, as the point-based prompting for SAM accounts for a 69.1% mIoU of the produced

pseudo-labels, compared to the respective 89.7% and 91.5% for the scribble and bounding box-based prompting methods mentioned in the previous subsections.

3.2.4 Image-level tags

Training based on image-level tags has received the most attention in WSSS [66]. Because such labels offer no localization hints, earlier works relied on emerging properties of neural networks to identify discriminative features of the classes and then propagate them. For instance, SEC [43] reformulates the problem as an optimization of three losses: one based on localization cues extracted from a classifier CNN; another that prevents too small predictions, and the last one which targets the spatial and color properties of the pixels. This paradigm is similar to methods that use class activation maps [100], such as PuzzleCAM [34]. There, the authors propose extracting CAMs for tiles or patches of the image, followed by a merging procedure. The CAM produced in this approach helps train the network using a reconstruction loss.

The survey by Chen et al. [14] highlights the focus placed on image-level label methods and the transition made from traditional CNN-based approaches to methods that rely on the generalization and zero-shot capabilities of large vision models. The study examines the primary applications of two foundational vision models: Contrastive Language-Image Pre-Training (CLIP) [68] and Segment Anything (SAM) [41].

CLIP encodes images and text into the same space, where similar inputs are close in the embedding space, therefore enabling a large degree of interaction between images and text. Weakly supervised methods leverage this property by turning image-level tags into language prompts. CLIMS [88] extracts the CAM of an image and maximizes the similarity between the highlighted region and a piece of text corresponding to the given class label, minimizing the similarity between the non-highlighted region and the same piece of text, suppressing hand-crafted backgrounds co-occurring with the given image tag. CLIP-ES [52] is a training-free method that relies again on hand-crafted prompts and refinement procedure to the initial CAM. Context Prototype-Aware Learning (CPAL) [78] introduces a self-supervised learning element to standard CAM-based methods by comparing the initially generated CAMs with those produced by a prototype-aware module, thereby iteratively refining them. Ultimately, the technique builds upon CLIP-ES and demonstrates improvements over the original results. Xu et al. [93] notice a modality gap between text and image in CLIP and propose a method of building vision prototypes instead of text prototypes. Their method builds upon CPAL and effectively enhances its segmentation accuracy. SemPLeS [49] corrects initially proposed masks by using trainable prompts for CLIP. These learnable prompts aim to suppress co-occurring background while still maximizing the similarity between objects proposed in the initial mask and the class-aware textual prompts.

Although CLIP has received more attention in WSSS, many works aim to extract pseudo-labels from SAM. The same study by Jiang et al. [32], referenced in the previous subsections, investigated the use of SAM when only image-level tags are

available. More precisely, they report the findings of two approaches: feeding points sampled from CAMs as prompts to SAM or using SAM in its "segment-everything" mode and adopting another method for classifying each proposed mask. Sun et al. [77] show how a pipeline composed of GroundedDINO [53] and SAM can sequentially turn image-level tags into bounding boxes and, subsequently, segmentation maps. Lastly, Yang et al. [94] jointly use CLIP and SAM. There, learnable CLIP prompts help propose an initial pseudo-segmentation mask. This, in turn, becomes a mask-based dense prompt for SAM, capable of refining the coarse prediction into a final pseudo-segmentation label used in the second stage fully supervised training. SAM also has applications in medical image segmentation with SimTxtSeg [89] evaluating a hybrid image-level tag and language supervision for extracting pseudo-labels from colonic polyp images and MRI brain tumor images.

3.2.5 Universal methods

Although less common, some approaches have identified that weakly supervised methods could exploit multiple forms of annotations during training. The motivation behind such methods is that incorporating diverse weak labels can enhance segmentation accuracy.

Xu et al. [90] propose to solve a max-margin clustering problem aimed at classifying super-pixels. The approach minimizes margin violation while satisfying a set of constraints derived from the available weak labels, supporting image-level tags, bounding boxes, and scribbles. For instance, the image-level tags constraints dictate that if a tag is not present in an image, no superpixels should be assigned that tag, and if a tag is present, then at least one superpixel takes it as its class label. Additionally, the method treats bounding boxes as smaller-sized images, enforcing the same tag-based constraints on them.

The approach of Ke et al. [37] involves encoding each pixel into an embedding space generated through a contrastive learning method. Various forms of contrastive relationships between pixels guide training. These include visual or textural similarity or being part of the same superpixel or not. Unfortunately, despite being able to simultaneously learn from all of the classic weak labels, the experiments include only three separate scenarios, where only tags, scribbles, or bounding boxes are used during training.

3.3 Differentiable Logics

Shukla et al. [74] tackle the problem of weakly supervised image classification by defining a differentiable count-based loss function for each of the three settings. These losses follow from a dynamic programming method that counts the probability that exactly s out of k variables are true, given their success probability. While the dynamic programming algorithm works well on the problem of image classification, it is intractable for image segmentation. This is because of its $\mathcal{O}(k^2)$ runtime complexity in the size of the items to be classified, which correspond to pixels in an image in the case of segmentation.

A probabilistic approach for training neural networks is DeepProbLog [58], which adds a neural predicate to ProbLog. The probability of success of such a neural predicate depends on the output probabilities extracted from a neural network. Learning becomes possible by grounding the logical program, compiling the knowledge into a structure that optimizes WMC, and computing a loss based on the probability of success of a query.

DL2 [26] specifies prior knowledge using logic and converts it to differentiable signals for training neural networks. The differentiable loss functions created through this method have the property that they are 0 when the constraint is satisfied. The logical language consists of boolean operators applied to comparison terms. The authors evaluate the method in various settings, including unsupervised, semi-supervised, and fully supervised scenarios, and demonstrate that learning from constraints is beneficial in all cases, with increased accuracy in unsupervised and semi-supervised settings and a high constraint satisfaction ratio when utilizing full supervision.

Some studies analyze the use of fuzzy operators for training neural networks. Van Krien et al. [79] model a set of constraints for two problems using the MNIST dataset [20]. With semi-supervision, the training process utilizes labeled samples to directly supervise a CNN while applying logical constraints to guide learning from unlabeled data. The study conducts a comparative performance analysis of common fuzzy operators, including not only T-norms and S-norms for conjunction and disjunction but also aggregators for universal and existential quantifiers. The conclusions indicate that the product t-norm typically achieves the best results. Similarly, Flinkow et al. [27] quantify both the prediction accuracy and the constraint satisfaction accuracy when using various constraints. They benchmark DL2 [26] and fuzzy approaches, concluding that the choice of differentiable logic depends on the problem.

Other techniques involve constructing a semantic loss function and fall under the scope of semantic-based regularization [21]. It involves converting first-order logic formulas into real values, which are used as a regularization term in the loss function. This approach also extends to training neural networks [22, 91], with the main applications being in a semi-supervised setting.

Logical Tensor Networks (LTN) [2] are yet another approach for integrating deep learning with logical reasoning. As opposed to the previously mentioned techniques that learn to shape predictions through a logic-based loss, LTNs focus on learning tensor representations for the predicates within the architecture. More specifically, the authors introduce a real logic formalism over first-order logic to accommodate tensors as the concrete interpretations of logical objects (i.e., terms, variables, constants). In this sense, as mentioned by van Krien et al. [79], LTNs combine neural computation with logic on a low level, while previously mentioned techniques add logic as a form of high-level learning. In LTNs, connectives and quantifiers correspond to fuzzy operators, facilitating learning. LTNs utilize the product t-norm T_P to replace conjunction, which generally performs better than other fuzzy logic operators in the context of training neural networks. Subsequent work improves this baseline, with logLTN [3] replacing the semantics of LTNs with semantics in the logarithm space for numerical stability.

3.4 Conclusion

Based on the information presented in this chapter, the motivation for this thesis becomes apparent: it stems from the current limitations of weakly supervised segmentation. Recent works have not addressed the possibility of learning from multiple types of weak labels, as most of the focus has been directed at learning from the simplest form of weak annotation, namely image-level tags. Earlier approaches designed to overcome this limitation, on the other hand, do not benefit from recent advances in semantic segmentation. While each of the approaches mentioned in this chapter has its strengths, there is a question of whether a method capable of learning from multiple such sources could achieve better results.

The differentiable logic systems discussed lead to several key conclusions. While a universal weakly supervised method already exists for image classification [74], it has not been extended to image segmentation. Probabilistic logic systems such as DeepProbLog rely on weighted model counting, which becomes computationally intractable in the high-dimensional space of image segmentation. In contrast, approaches based on fuzzy logic remain promising due to their efficient inference. As several studies have shown, their effectiveness depends on selecting fuzzy operators that align with the problem at hand, namely image segmentation. Logic Tensor Networks, particularly logLTN, are well-suited for segmentation, where scalability to high-dimensional data is essential.

Chapter 4

Problem statement

This chapter defines the research problem addressed in this thesis: training a segmentation model in a weakly supervised manner using various forms of indirect supervision. The thesis aims to balance two perspectives by investigating the theoretical application of a differentiable logic to address this problem while also demonstrating strong segmentation quality in practical settings.

4.1 Problem definition

The thesis focuses on weakly supervised *semantic* segmentation, the most studied form of image segmentation in the literature on weak supervision. As introduced in earlier chapters, semantic segmentation is the problem of assigning a class label to every pixel of an image [18]. Deep learning, through the training of neural networks, achieves state-of-the-art results in semantic segmentation [15]. In this setting, before formally defining WSSS as addressed in this thesis, this section defines semantic segmentation in the context of deep learning.

Definition 4.1.1 (Semantic segmentation with neural networks). Let $\mathcal{X}$ be the input space of images, where $x \in \mathcal{X}$ is an image of shape $H \times W \times C$, with H and W the height and width of the image, and C the number of channels (3 for RGB images). Let $\mathcal{Y}$ be the output space, where $y \in \mathcal{Y}$ is an image of shape $H \times W$ called the segmentation mask, with values among a finite set of possible classes $\mathcal{C}$, with $|\mathcal{C}| = C$.

Then, semantic segmentation using neural networks aims to learn a parameterized function

$$f_\theta \colon \mathcal{X} \to [0,1]^{H \times W \times C}, \tag{4.1}$$

that maps an image x to per-pixel class probabilities $P(y_{i,j} = c|x)$ with $c \in \mathcal{C}$. Thus, the segmentation mask is computed by assigning the class with maximal probability to each pixel:

$$\hat{y}_{i,j} = \arg\max_{c \in \mathcal{C}} f_\theta(x)_{i,j,c}. \tag{4.2}$$

Further, this thesis defines WSSS in a way that explicitly uses a set F of first-order logic formulas expressing (soft) constraints and prior knowledge that guide training. Although other methods may rely solely on weak labels, this formulation highlights the three components necessary for training in the context of this thesis: the input images, the weak annotations, and the logical constraints.

Definition 4.1.2 (Weakly supervised semantic segmentation).
Let $\mathcal{X}$ be the input space of images defined earlier, $\mathcal{W}$ the space of annotated weak labels, and F a set of first-order logic formulas derived from $\mathcal{W}$ or introduced as prior knowledge. Let $\mathcal{D}$ be a dataset consisting of N pairs of images and their weak labels (x_i, W_i), with $x \in \mathcal{X}$ and $W_i \subset \mathcal{W}$.

The objective of weakly supervised semantic segmentation is to learn the same function f_θ as in equation 4.1 by optimizing a loss function

$$\mathcal{L}(f_\theta|\mathcal{D}, F) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f_\theta(x_i), W_i, F). \tag{4.3}$$

4.2 Objectives

4.2.1 Learning from classic weak labels

Unlike prior work that designs ad hoc solutions for every type of supervision, WSSS is tackled here in a principled and unifying manner. The goal is to support a variety of weak labels for training by deriving learning signals from at least the four main types of weak labels: image-level tags, points, scribbles, and bounding boxes.

Goal 1 Derive and incorporate learning signals from the four main types of weak labels (image-level tags, points, scribbles, and bounding boxes) into the training of the segmentation network.

4.2.2 Incorporating prior knowledge

Beyond the standard weak labels, segmentation networks can benefit from other forms of indirect supervision during training. Examples include spatial priors (e.g., "*object 1* is to the right of *object 2*", "if there is a tree, then the ground is below the tree"), low-level structural constraints (e.g., "if a pixel is labeled with class c, at least one of its neighbors must be labeled with class c), and high-level domain-specific knowledge (e.g., "if there is a boat, the image must contain water"). Therefore, this thesis aims to enhance the learning process with signals derived from first-order logic.

Goal 2 Derive and incorporate learning signals from first-order logic, including quantifiers over finite domains, into the training of the segmentation network.

Note that restricting quantifiers to finite domains does not limit the expressiveness in this setting since logical objects are network outputs, and thus, their set is finite.

4.2.3 Outperforming other WSSS methods

Because other works exploit a single type of weak label, a reasonable goal is that the proposed system obtains more accurate segmentation results when multiple annotation types are available. Intuitively, spending more effort on annotation should result in higher-quality predictions.

Goal 3 Consider a dataset of images $\mathcal{I}$, split into training and testing sets $\mathcal{I}_{train}$ and $\mathcal{I}_{test}$, along with bounding-box annotations b_n^k and scribble annotations s_n^k for every object of interest k present in each training image $\mathcal{I}_n \in \mathcal{I}_{train}$. Let W_{train} be the set of all weak annotations available for the training set, where $W_{train}^n = \{b_n^i | i \in [1..k]\} \cup \{s_n^i | i \in [1..k]\}$ are thus the weak labels for the nth training example. Let F be the set of logic formulas that the model aims to maximally satisfy during training. Lastly, consider the ground-truth segmentation masks $\mathcal{M}_{test}$ for every image of the test set.

Then, after training with $(\mathcal{I}_{train}, W_{train}, F)$, the segmentation network should outperform methods that exploit a single type of weak label on the test set $\mathcal{I}_{test}$, as measured by the mIoU with respect to $\mathcal{M}_{test}$.

The focus is therefore limited to datasets annotated with bounding-box and scribble labels, as these achieve a balance between (1) the ease of annotation and (2) the amount of information provided. Notice that with these annotations, the method receives more localization and spatial information than methods that exploit only image-level tags, only points, or only one of the bounding box and scribble annotations, and hence should outperform those methods.

As the majority of works in the literature measure the quality of predictions through the mean intersection over union (mIoU), this thesis also adopts this metric in the evaluation. This approach then additionally assumes access to ground-truth segmentation labels for the test set to compute the mIoU. This is also a fair assumption in a practical setting, as a (smaller) set of fully annotated segmentation masks serves to quantify the quality of the segmentations produced.

Chapter 5

Proposed method

This chapter introduces and details the proposed system. It begins with an overview of the method, highlighting the novelties and justifying the choices made. Then, it describes the proposed method in depth in the following two sections, organized along the two main stages of the proposed framework: a weakly supervised training of SAM, followed by a fully supervised training of a segmentation network on pseudo-labels produced by SAM. As the majority of the contributions of this thesis are in the first stage, the discussion focuses more extensively on that part.

5.1 Overview

To reiterate the context, WSSS aims to train a segmentation model based solely on weak annotations, finally obtaining a network that can operate independently at inference time on unseen images. At the same time, this thesis aims to facilitate learning from various types of weak labels as well as other prior knowledge.

To achieve these goals, the proposed framework addresses WSSS through a two-staged pipeline designed to accommodate weak supervision while also ultimately producing a prompt-free segmentation network. Figure 5.1 provides a visual representation of the approach. This two-staged workflow is standard, as illustrated in the related work chapter. At a high level, it follows these steps:

1. **Weakly supervised fine-tuning of the Segment Anything Model (SAM)** through indirect supervision signals derived from logical constraints
2. **Fully supervised training of a segmentation network** with the pseudo-labels as ground truths

In the first stage, the method uses bounding-box prompts as input for SAM to produce masks for each object instance. It then merges these masks to create a pseudo-segmentation label for the entire image - that is, a segmentation mask generated without manual annotation. Although SAM provides high-quality predictions in various zero-shot contexts, the proposed system further improves segmentation results by training SAM to *maximally satisfy* the set of given formulas $\mathcal{F}$. These

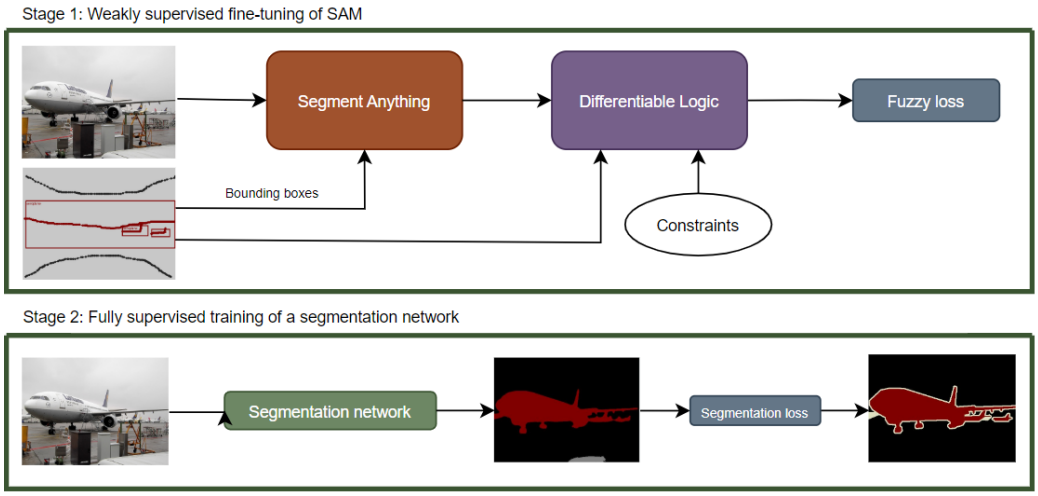


Figure 5.1: Overview of the proposed method, as a two-stage pipeline.

formulas capture both high-level properties present in the training images and low-level structural properties that the predicted masks should satisfy. Grounding these formulas combines network outputs with logical fuzzy connectives, producing differentiable learning signals. Most WSSS techniques treat foundational models as static pseudo-label generators, focusing solely on designing better prompts. In contrast, this work fine-tunes SAM with the indirect forms of supervision previously obtained before using it to generate pseudo-labels for the training samples. In this latter phase, the trained SAM directly produces masks when prompted with weak prompts, specifically bounding boxes.

The last step acts as a standalone, separate part of the process. It involves training a segmentation model using the previously obtained pseudo-labels. This training treats the pseudo-labels as ground truth data in place of the usual manual pixel-wise annotations. This training is easily reproducible and follows standard training pipelines set in previous studies.

5.1.1 Novelty

The proposed method contributes to WSSS from two perspectives. First, from a capability standpoint, the system supports learning from various forms of indirect supervision. Although some earlier works discussed in a previous chapter have explored learning from all of the classic weak labels [37, 90] in the problem of image segmentation (bounding boxes, scribbles, points, and image-level tags), they do not leverage of recent advances in semantic segmentation. Furthermore, they overlook other prior knowledge that may be available for a specific task. This thesis argues that predicate logic provides a flexible framework for incorporating prior knowledge into the learning process. This universality is especially valuable in practice, where multiple learning signals can compensate for the coarse level of annotation associated with weak supervision.

Second, from a methodological perspective, this thesis demonstrates the use of a neuro-symbolic framework for semantic segmentation, a largely unexplored research direction. The proposed framework blends logical reasoning and neural networks, sitting at the intersection between Logic Tensor Networks [2] and semantic-based loss construction [22]. The fuzzy operators employed are particularly suitable for expressing constraints over segmentation outputs, enabling an efficient and differentiable computation of semantic loss terms. The proposed set of constraints facilitates training, achieving state-of-the-art results on two different benchmarks, Pascal VOC 2012 [24] and LGG segmentation [64].

5.1.2 Reasoning

Three choices shape the approach. Firstly, splitting the training process into two stages adds flexibility and, more importantly, ensures alignment with the desired outcome of the WSSS problem, namely obtaining a prompt-less segmentation network. Secondly, the choice of Segment Anything as a pseudo-labeler provides reliable baseline performance for pseudo-label generation. Lastly and most notably, a tailored set of fuzzy operators enables expressing and learning from complex constraints in a weakly supervised setting, forming the core contribution of this thesis.

Staged training Most methods tackle WSSS by splitting the learning process into the two stages previously mentioned. The primary motivation stems from the outcome required by the problem of weakly supervised segmentation, as defined in Section 4.1. More specifically, even though the training employs weak labels, the final segmentation network should perform inference on new images, *without* the use of additional labels or hints as input. The second stage of the procedure achieves this with a fully supervised training of a segmentation model.

Additionally, many studies demonstrate that the second-stage network achieves higher segmentation accuracy than the one trained in the first stage [49, 94]. While not a formal justification, especially since the performance is measured on two different splits of the data, this result aligns with the findings of other studies [71], which demonstrate that neural networks are robust even when learning on noisy labels. In this case, the segmentation network trained in the second stage learns to identify the common, correct patterns that appear in input images and their pseudo-labeling while simultaneously learning to ignore the less common, incorrect parts of the segmentation seeds.

An alternative worth considering is the adoption of a single-stage training pipeline in place of the proposed method. Despite the benefit of simplifying the training process, such a method sacrifices flexibility. For instance, the two-stage strategy is particularly suitable in resource-constrained environments. Accurate pseudo-labels obtained from SAM are likely to lead to improved segmentation performance for a second-stage training of a lightweight segmentation network, compared to the alternative of directly training it with coarse signals from weak supervision. Section 7.3 further emphasizes this as a direction for future work.

Segment Anything for pseudo-labels Recent works in the WSSS literature utilize large, foundational models to generate accurate pseudo-labels. While using CLIP in this context has its merits, this thesis focuses on analyzing and improving masks produced by SAM. Two key factors motivate this choice, namely (1) SAM directly performs segmentation, and (2) SAM exhibits a strong baseline performance when prompted with bounding boxes [32], a type of weak label that the proposed system also learns from.

The related works chapter showed that most techniques focus on designing or learning more effective prompts for the foundational models. However, given the interconnectivity of this thesis with logic and the differentiable signals derived from it, a weakly supervised fine-tuning of SAM with these signals as part of the loss function represents a more logical direction.

Differentiable logic On the one hand, training a neural network with (soft) logical constraints implies being able to express these constraints on the network outputs, facilitating backpropagation. On the other hand, segmentation is, in part, defined by the large dimensionality of the predicted mask, as opposed to a task such as classification. Therefore, these two perspectives show that scalability is crucial when incorporating logic over the outputs of a segmentation network.

To show the necessity of operators with fuzzy semantics, this discussion first examines probabilistic logic as a potential choice for a differentiable framework. In this context, the network outputs directly represent probabilities by using a final Softmax layer. Given the success probability p of a formula ψ to learn from, using the learning signal $-log(p)$ corresponding to the negative log-likelihood loss maximizes the success probability of ψ. The primary challenge of probabilistic logic is the efficient computation of the success probability for an arbitrary formula, which is closely tied to the weighted model counting problem. Consider the bounding box tightness prior [46] expressed only on rows, i.e., "every row of the predicted mask contains at least one pixel with the class of the bounding box". Given a predicate $Class(i, j, c)$ that is true when pixel on row i and column j has class c and a bounding box with class c represented by its upper-left and bottom-right coordinates (y_{min}, x_{min}) and (y_{max}, x_{max}), the first-order logic formula for this constraint (assuming an extended syntax for bounded quantifiers) is

$$\forall i \in [y_{min}, y_{max}] \ (\exists c \in [x_{min}, x_{max}] \ Class(i, j, c)) \tag{5.1}$$

The success probability of the formula, given the output probabilities $\hat{p}$ of the segmentation network, has the expression:

$$\prod_{i \in [y_{min}, y_{max}]} (1 - \prod_{j \in [x_{min}, x_{max}]} (1 - \hat{p}_{i,j,c}))). \tag{5.2}$$

Computing this formula is linear in the size of the bounding box. For a slightly more complex constraint such as "every row *and every column* of the predicted mask contains at least one pixel with the class of the bounding box", there is no simple formula. WMC (WFOMC) remains the only viable method for exact computation;

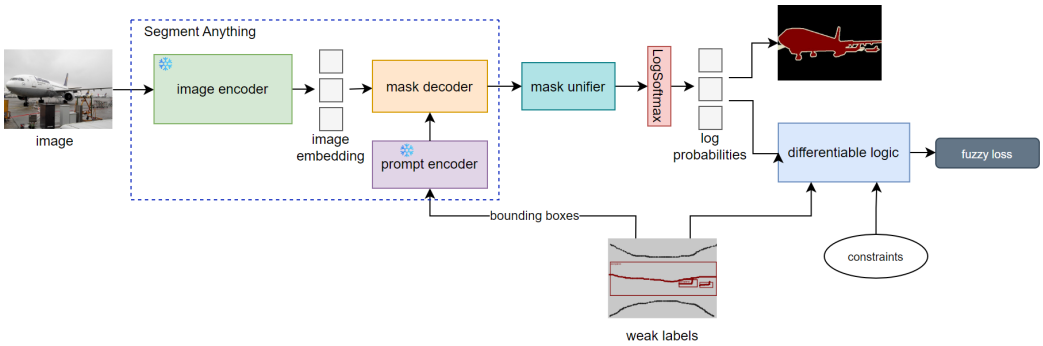


FIGURE 5.2: First stage: weakly supervised fine-tuning of SAM

however, it is intractable for constraints involving this many variables (predicates), as the global dependencies involved prevent a factorization such as the one in Equation 5.2.

In contrast, predicate fuzzy logic is a more scalable and practical approach. Instead of dealing with uncertainty in terms of probabilities, fuzzy logic models *vagueness* through degrees of truth. It is particularly suitable for training neural networks, as most fuzzy operators are differentiable by design. This allows the optimization of fuzzy logic constraints as part of the loss function. Although probabilistic inference also supports differentiation [58], fuzzy approaches offer a more efficient inference process, an aspect of utmost importance in the task of image segmentation.

5.2 First stage: Weakly supervised fine-tuning of SAM

This section performs an analysis of the first stage in the proposed workflow. Figure 5.2 illustrates the architecture along with the key components involved. The discussion proceeds by examining each component individually, following a logical order.

5.2.1 Inputs

The first stage benefits from a training set consisting of images $\mathcal{I}_{train}$, as well as weak annotations W_{train}. As this thesis aims to create a general framework for learning with any weak label types, depending on their availability, the method does not *require* the presence of a particular label type. However, it does require *at least one* weak label to be present for each object instance in an image, such that SAM can produce a mask for that object. Without losing the universality of this method, the sections that follow assume that bounding boxes for every object act as prompts for SAM.

Additionally, the method relies on a set of logic formulas F. As noted in Section 5.2.5, expressing prior knowledge as first-order logic formulas is an essential part of this framework, requiring careful and thoughtful design. These formulas serve as the

foundation of learning, as the framework interprets them through fuzzy semantics and incorporates them in the construction of loss functions.

5.2.2 Segment Anything

While most prior techniques design or learn better prompts for SAM, this approach instead improves segmentation accuracy by *fine-tuning* the model. Given the large number of parameters, it is impractical to train all of them. The image encoder of the largest model (ViT-H), in particular, contains 636 million parameters.

Freezing both encoders while keeping the lightweight mask decoder trainable addresses this issue. This choice is simple yet effective, striking a balance between segmentation performance and resource efficiency.

From a semantic and quality standpoint, given its immense scale and pretraining, the image encoder already produces rich embeddings that possess all the properties required for downstream tasks. Similarly, fine-tuning already specializes in converting prompts into a meaningful representation for the model. Thus, the mask decoder offers the most significant potential for performance improvement, having the capability of (1) extracting the relevant semantics from the image embeddings and (2) using the localization information as given by the prompt encoder.

Notably, training only the mask decoder reduces the resources required and the training time. Since image embeddings are static during training, pre-computing and storing them removes the need for forward (and backward) passes through the image encoder during training. This observation holds when training does not include image augmentations.

These training settings enable the use of SAM-huge, the largest of the Segment Anything models, with significant performance gains over smaller versions. Although the mask decoder does not support a batched input of image embeddings, it does allow batched inference with multiple prompts per image. As stated previously, without loss of generality, bounding boxes serve as inputs to the prompt encoder. This choice relies on the results obtained by Jiang et al. [32], which show that prompting SAM with bounding boxes produces pseudo-labels of higher quality than prompting with other types of labels. GPU memory usage can vary because each image may contain a variable number of objects, i.e., a variable number of bounding boxes. For the experiments in this thesis, however, this first stage uses less than 10 GB of VRAM.

5.2.3 Mask unifier

For each prompt, SAM produces three binary segmentation masks, each associated with a confidence score. The multi-mask output intends to solve ambiguities that arise from prompts. For instance, when a point is placed on a car wheel and serves as prompt, it is unclear whether the desired target is the wheel or the entire car. In this sense, for each bounding box prompt, the mask unifier selects the binary mask with the highest score. More sophisticated selection schemes exist, especially in a multiclass setting. However, they are redundant, considering that fine-tuning of the

mask decoder also affects the scoring. Therefore, taking the mask with the highest score delegates the resolution of ambiguities to the fine-tuning process.

More importantly, two challenges emerge due to the class-agnostic nature of SAM.

Firstly, when provided with a prompt, SAM produces a binary mask that distinguishes foreground and background, with no information of classes. The first step in constructing a coherent, multiclass segmentation mask is to assign the foreground in each binary mask the label of the bounding box used for prompting.

A second challenge emerges when these binary masks overlap, introducing ambiguity in the resulting multi-class segmentation map. To address this, the mask-merging procedure described in detail in Algorithm 1 combines the binary masks by selecting the most probable outcome. More specifically, given the set of binary masks and the corresponding class labels derived from the bounding box prompts, the algorithm constructs a $H \times W \times C$ tensor of log probabilities. Each channel in the tensor corresponds to one class and contains per-pixel confidence scores for that class. For a given class, the mask is obtained by taking the element-wise maximum over all binary masks associated with that class. This algorithm treats the background separately, as the absence of any foreground prediction. Finally, a log-softmax operation applied across channels transforms the confidence scores into log probabilities. This operation is numerically stable.

Algorithm 1 Mask-merging procedure for constructing the multi-class log-probability map

Input: List of masks $\{m_i\}_{i=1}^{N}$, class tags $\{t_i\}_{i=1}^{N}$, number of classes C
Returns: Log-probability tensor $Z \in [-\infty, 0]^{C \times H \times W}$
Initialize all elements of Z with $-\infty$
for $i = 1$ to N **do**
    $Z_{t_i} \leftarrow \max(Z_{t_i}, m_i)$ # aggregate scores for each class
end for
$Z_0 \leftarrow -\max(Z_{1:C}, \text{dim} = 0)$ # background = negative of max foreground
$Z \leftarrow \log \text{Softmax}(Z, \text{dim} = 0)$ # normalize to log-probabilities
return Z

This procedure clearly shows that backpropagation does not occur through *all* network outputs. Instead, only the *most confident* predictions which form the tensor of log probabilities of shape $H \times W \times C$ are part of the computation. This form enables reasoning using fuzzy semantics, as detailed in the following sections. Besides the simplification, focusing on the most confident predictions is beneficial from a learning perspective. A wrong, high-confidence prediction receives a stronger penalty and implicitly affects the discarded predictions.

5.2.4 Differentiable logic in the logarithm space

WSSS relies on indirect supervision signals originating from sparse and weak annotations such as scribbles, bounding boxes, or other background knowledge. These do

not directly constrain what the network predicts. Instead, the network learns from the structured, higher-level relationships expressed over regions of the image. To connect the high-level constraints with the low-level predictions of neural networks, this differentiable logic framework operates directly on the low-level outputs of the neural network. This approach employs first-order fuzzy logic with the product t-norm, defined in logarithmic space. Operating in log-space ensures numerical stability and compatibility with neural network outputs, particularly when handling high-dimensional segmentation maps.

Syntactically, formulas in this logic are constructed from terms, predicates, logical connectives, and quantifiers. Terms represent objects in the domain, such as pixel coordinates and class labels. Predicates express properties or relations over these terms. *Interpreted* predicates such as $Class(i, j, c)$, which expresses that pixel (i, j) has class c, differ from predicates such as $Scribble(P, c)$ that *symbolically* encode weak forms of supervision. The standard logical connectives - conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), implication ($\rightarrow$) - combine predicates and other subformulas. Universal ($\forall$) and existential ($\exists$) quantifiers range over finite domains and contribute to the expressivity of this logic.

In terms of **semantics**, in fuzzy logics, truth values correspond to the interval [0, 1], where zero means completely false and one corresponds to entirely true. Instead, a grounding function G grounds the formulas in this logic to tensors with values in the interval $[-\infty, 0]$. Additionally, G grounds logical connectives and quantifiers to fuzzy operators that operate on tensors. This definition of grounding is similar to the one in Logic Tensor Networks [2], departing from the common meaning of the term that refers to substituting variables with constants or terms. Also notably, the negation, conjunction, disjunction, and implication operate *element-wise* on the tensor inputs. At the same time, the quantifiers work in a *vectorized* manner across a dimension or over the entire input tensor.

Connection with the neuro-symbolic literature The method described resembles Logic Tensor Networks and logLTN [3] in particular, as both rely on fuzzy semantics in the logarithmic space and operate on tensor data. LTNs are generic frameworks suitable for various applications, making the logic explicit by modeling predicates through neural networks, with the raw inputs serving as terms of these predicates. In contrast, the tailored implementation in this thesis addresses segmentation by directly operating on the network output, a tensor $\hat{z}$ with shape $H \times W \times C$, in line with the literature on semantic-based regularization. Predicates such as $Class(i, j, c)$ serve only to illustrate the logical formulas involved and the theoretical foundations for this thesis. Notably, although both operate in the logarithm space, this implementation departs from the operators for negation, disjunction, and aggregation defined in logLTN.

Thus, the proposed system aligns with the LTN framework, belonging to the same family of differentiable fuzzy logics that operate on tensors in a scalable manner. Still, the focus is on creating loss terms from network outputs, similar to semantic-based regularization techniques; however, these terms serve as the only supervision signals

here. Borrowing from these frameworks enables the proposed system to successfully tackle segmentation, a direction the literature has yet to explore.

Negation The grounding of the negation connective $\neg$ is a fuzzy negation operator. More specifically, it is the log-space equivalent of the strong negation $N_P(x) = 1 - x$, which corresponds to $N_{logP}(x) = \log(1 - e^x)$, also referred to as the *log1mexp* operation. Therefore,

$$G(\neg\psi) = N_{logP}(G(\psi)) = \log(1 - e^{G(\psi)}) \tag{5.3}$$

In practice, this operation is numerically unstable if performed naively. To solve this issue, *log1mexp* is computed with the method proposed by Mächler [63], which handles two cases, depending on whether x is small or large, using numerically stable implementations of the $log1p(x) = \log(1 + x)$ and $expm1(x) = e^x - 1$ operations. More specifically:

$$\log(1 - e^x) = log1mexp(x) = \begin{cases} \log(-expm1(x)) & 0 < x \leq \log(2) \\ log1p(-e^x)) & x > log(2) \end{cases} \tag{5.4}$$

LogLTN, in contrast, does not use a negation operator, opting to express (or convert) formulas in negative normal form (NNF). While this does not limit the expressivity of the language, the process may alter the truth values of formulas. Concretely, a limitation of logLTN that the authors also identify is that it does not preserve logical equivalences. Not defining a negation operator also implies that the fuzzy operator configurations defined are not symmetric, i.e., the disjunction is not the dual co-norm of the conjunction.

Conjunction In this work, the grounding of the conjunction $\wedge$ originates from the product t-norm $T_P(x, y) = x \cdot y$. Again, the log-space equivalent T_{logP} replaces the original formulation:

$$G(\psi \wedge \phi) = T_{logP}(G(\psi), G(\phi)) = G(\psi) + G(\phi) \tag{5.5}$$

This choice mainly stems from the properties of the gradient of T_{logP}. More specifically, the gradient with regard to either one of its inputs is 1, ensuring that there are no vanishing or exploding gradients during backpropagation.

Other fuzzy t-norms lack these properties and are therefore unsuitable for training neural networks. For instance, the original product t-norm T_P has non-zero gradients for both of its inputs. Still, the gradient itself can have a significant variation of scale, depending on the values of x and y, possibly leading to exploding or vanishing gradients when chaining multiple conjunction operations. The Gödel t-norm $T_G(x, y) = min(a, b)$ is unsuitable for learning because it is *single-passing* [79], meaning that during backpropagation, only one of the inputs has a non-zero derivative. Other t-norms such as the Łukasiewicz t-norm $T_{LK}(x, y) = max(x + y - 1, 0)$ stop the gradient propagation entirely when the sum of x and y is smaller than 1 [79].

Disjunction The implementation of the disjunction $\vee$ operator combines the conjunction and negation operators defined previously, using de Morgan's law:

$$G(\psi \vee \phi) = G(\neg(\neg\psi \wedge \neg\phi)) \tag{5.6}$$

This formulation corresponds to the dual co-norm of T_{logP} and ensures that disjunction inherits the numerical stability and other properties of the conjunction and negation operators. Like the log-space conjunction, this implementation for disjunction supports differentiable reasoning without the risk of vanishing or single-passing gradients.

An operator such as Log-Sum-Exp may require less operations and may be more numerically stable. Still, it risks exceeding the upper bound of the $[-\infty, 0]$ domain of log probabilities because it corresponds to a sum of probabilities that ignores double-counting. Thus, further chaining Log-Sum-Exp with the negation operator is not possible, affecting the expressivity of the logic. LogLTN uses a Log-Mean-Exp operator instead, which is bounded by the maximum value of the tensor, thereby resolving the issue with Log-Sum-Exp. However, this configuration again loses interpretability from a probabilistic perspective, as the disjunction replaced by Log-Mean-Exp receives a truth value lower than that of one of the terms within the disjunction.

Implication For similar reasons to the choice of the fuzzy disjunction operator, The Reichenbach S-implication replaces the implication connective:

$$G(\psi \rightarrow \phi) = G(\neg\psi \vee \phi). \tag{5.7}$$

Quantifiers The previous connectives apply an element-wise operation on the elements of their input tensors. For instance, the conjunction $\wedge$ returns a tensor with the same shape as the two input tensors, where each element is the sum of the corresponding two elements of the input tensor. Universal and existential quantifiers, however, use aggregation to *reduce* the dimensionality of the input tensor.

Crucially, for the problem at hand, this aggregation is expressive enough to support two use cases: (1) aggregation across a single dimension of a multi-dimensional input tensor and (2) aggregation across the entire input tensor (which can be single or multi-dimensional). The implementation allows, thus, these two modes of operation. In a style similar to the *dim* argument of PyTorch tensor operations, the two modes differ by the presence or absence of this additional argument that specifies the dimension used for aggregation.

The grounding of the universal quantification $\forall$ corresponds to the A^U_{logP} aggregator, which is the conjunction of all elements of tensor $x \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_n}$. Equivalently, it applies the log-space product t-norm defined earlier to the elements involved in the aggregation. The equations below show this more formally:

$$
\begin{aligned}
G(\forall x\ \psi) = A^U_{\log P}(G(\psi)) &= \sum_{i_1=1}^{d_1}\sum_{i_2=1}^{d_2}\cdots\sum_{i_n=1}^{d_n} G(\psi)_{i_1 i_2 \ldots i_n} \\
G(\forall_k x\ \psi) = A^U_{\log P}(G(\psi), k) &= y \in \mathbb{R}^{d_1\times\cdots\times d_{k-1}\times d_{k+1}\times\cdots\times d_n}, \\
\text{with} \quad y_{i_1,\ldots,i_{k-1},i_{k+1},\ldots,i_n} &= \sum_{i_k=1}^{d_k} G(\psi)_{i_1,\ldots,i_{k-1},i_k,i_{k+1},\ldots,i_n}
\end{aligned}
\tag{5.8}
$$

where $\forall_k$ refers to the quantification along the kth dimension of tensor x.

Similar to de Morgan's law for conjunction and disjunction, the duality between the quantifiers $\exists x P(x) \leftrightarrow \neg\forall\neg P(x)$ helps ground the existential aggregator A^E_{logP}:

$$
\begin{aligned}
G(\exists x\ \psi) &= G(\neg\forall x\ (\neg\psi)) \\
G(\exists_k x\ \psi) &= G(\neg\forall_k x\ (\neg\psi)),
\end{aligned}
\tag{5.9}
$$

where $\exists_k$ refers to the existential quantification along the kth dimension of tensor x.

LogLTN uses a slightly different universal quantifier A^U_{mean}. Instead of a sum, it performs the mean of truth values along the quantified dimension. The authors justify this change by examining the size of the gradients. They find that the gradient depends on the type of quantifier: the universally quantified formulas receive a gradient of -m, where m is the number of objects quantified over, while existentially quantified formulas receive a gradient of -1. Additionally, a universally quantified formula with more domain elements receives a stronger gradient than one with fewer.

The first limitation is a byproduct of the Log-Mean-Exp-based aggregation operator A^E_{LME} used by logLTN in place of the existential quantifier. In this framework, $A^E_{\log P}$ inherits the properties of the universal aggregator that it internally uses, even though the negation operator also affects the gradients.

Regarding the second point, at least for the types of constraints relevant to WSSS, there is no reason to force the gradients of formulas to be the same. What matters more is that pixels receive similar supervision signals. Indeed, taking scribbles as an example, a pixel that is part of a longer scribble should count equally towards the loss as a pixel that is the only point in a scribble. The mean aggregator A^U_{mean} used in logLTN instead distributes the gradient depending on the length of the scribble, which means that the single pixel of the latter case weighs more on gradient updates. The universal aggregation operator $A^U_{\log P}$ defined earlier accounts for this observation.

5.2.5 Constraints

The three-dimensional tensor of log probabilities $\hat{z} \in (-\infty, 0]^{H\times W\times C}$, which constitutes the unified mask as predicted by SAM and merged by the mask unifier, serves as the foundation for constructing logical constraints. The aim is to use network outputs and express high-level first-order logic formulas derived from the available weak labels or prior background knowledge.

Before describing the proposed constraints one by one, this section first introduces more notation. Firstly, the predicate $Class(i, j, c)$ has the informal semantics that

pixel on row i and column j has class c. Consequently, the grounding of this predicate corresponds to values of the network outputs $\hat{z}$, with $G(Class(i,j,c)) = \hat{z}_{i,j,c}$. Additionally, a syntactic sugar notation replaces formulas that restrict quantifiers to domains that satisfy a certain condition, such as $\forall x \ (P(x) \rightarrow \psi)$, with expressions like $\forall x \in P \ \psi$. This shorthand notation aligns with the grounding performed in the system, which applies the aggregator operator over finite domains by enumerating all possible cases.

This work defines six general-purpose constraints, with proven benefits for training in the context of WSSS. This section introduces and analyzes them in order by expressing them for an image x_i and its weak annotations W_i. Weak annotations correspond to symbolic predicates in these constraints. The predicate $Box(ymin, xmin, ymax, xmax, c)$ expresses a bounding box by its upper-left and bottom-right corners, as well as the labeled class c. $Scribble(P, c)$ denotes a scribble with P being the scribble coordinates and c being their associated class.

$\psi_{\text{background}}$ Informally, the background constraint $\psi_{background}$ expresses that pixels that are not within a bounding box belong to the background. The constraint has the following expression for image x_i with weak annotations W_i:

$$\begin{aligned}\psi_{\text{background}} = \forall i \ \forall j \ \Big(\neg\exists \ \text{Box}(y_{\min}, x_{\min}, y_{\max}, x_{\max}, c) \in W_i \\ (y_{\min} \leq i \leq y_{\max} \ \wedge \ x_{\min} \leq j \leq x_{\max})\Big) \rightarrow \text{Class}(i,j,0)\end{aligned} \tag{5.10}$$

To improve clarity and to show the form specified in the implementation, the following shorthand notation uses the set Outside(W_i) defined as the set of pixel coordinates not covered by any bounding box in W_i. Using this definition, the compact expression of the background constraint is:

$$\begin{aligned}\text{Outside}(W_i) = \left\{ (i,j) \;\middle|\; (i,j) \notin \bigcup_{\text{Box}(y_{\min},x_{\min},y_{\max},x_{\max},c)\in W_i} [y_{\min}, y_{\max}] \times [x_{\min}, x_{\max}] \right\} \\ \psi_{\text{background}} = \forall (i,j) \in \text{Outside}(W_i) \quad \text{Class}(i,j,0)\end{aligned} \tag{5.11}$$

This constraint provides a powerful learning signal for the background class, as it is equivalent to the learning signals used in a fully supervised setting for those pixels. This constraint is thus, in general, essential for learning the background regions in images. However, under this specific setup with the Segment Anything model, its impact is limited by the powerful prompt encoder and mask unification, which favor predicting object masks only within the bounding boxes used as prompts.

$\psi_{\text{bbox\_tightness}}$ This constraint expresses the bounding box tightness prior [46]. In natural language, it expresses that every row and every column of a bounding box contains at least one pixel with the class of the bounding box. For a single bounding

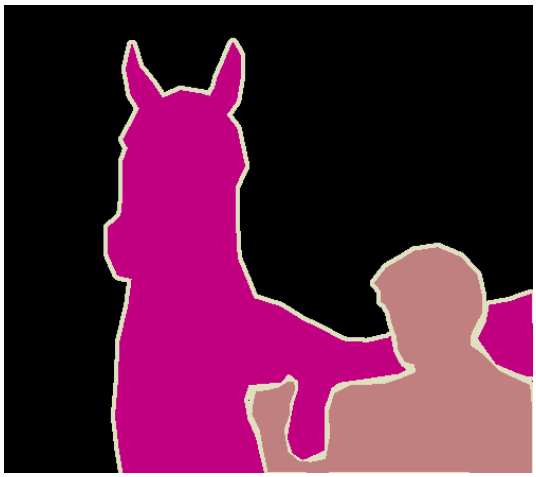


FIGURE 5.3: Ground truth mask that shows the original formulation of $\psi_{bbox\_tightness}$ is incorrect in a multi-class settings.

box b with (y_{min}, x_{min}) and (y_{max}, x_{max}) as as the upper-left and bottom-right corners and class c, the following first-order logic formula expresses this constraint:

$$\begin{aligned} \psi_b &= \psi_{b_{rows}} \wedge \psi_{b_{columns}} \\ \psi_{b_{rows}} &= \forall i \in [y_{min}, y_{max}] \; \exists_1 j \in [x_{min}, x_{max}] \; Class(i, j, c) \\ \psi_{b_{columns}} &= \forall j \in [x_{min}, x_{max}] \; \exists_0 i \in [y_{min}, y_{max}] \; Class(i, j, c) \end{aligned} \tag{5.12}$$

Thus, over all bounding boxes of image x_i with weak annotations W_i, the formula is:

$$\psi_{bbox\_tightness} = \forall Box(y_{min}, x_{min}, y_{max}, x_{max}, c) \in W_i \quad \psi_b \tag{5.13}$$

In a multi-class setting, an issue arises when an object with a class c_1 completely occludes an object with class $c_2 \neq c1$ across a row or column of the second object's bounding box. Thus, the constraints specified above are no longer true for that row or column. This is evident in Figure 5.3, where the foreground object, namely the person, occludes several columns corresponding to the bounding box of the horse. To solve this issue, $\psi_{b_{rows}}$ accounts for the possibility that each row i of bounding box b contains at least one pixel with the class of b *or with the class of any other bounding boxes that intersects row i of b*. Similarly, $\psi_{b_{columns}}$ imposes that each column expects at least one pixel with the class of the current bounding box b or the class of any other bounding box that intersects with b at that column.

$\psi_{\text{scribbles}}$ The scribble-based constraint is equivalent to partial supervision, i.e., it contributes with a loss term, which is the negative log-likelihood loss for the pixels of the scribble. Concretely, this corresponds to the first-order logic formula:

$$\psi_{\text{scribbles}} = \forall \text{Scribble}(P, c) \in W_i \; \forall (i, j) \in P \; \text{Class}(i, j, c). \tag{5.14}$$

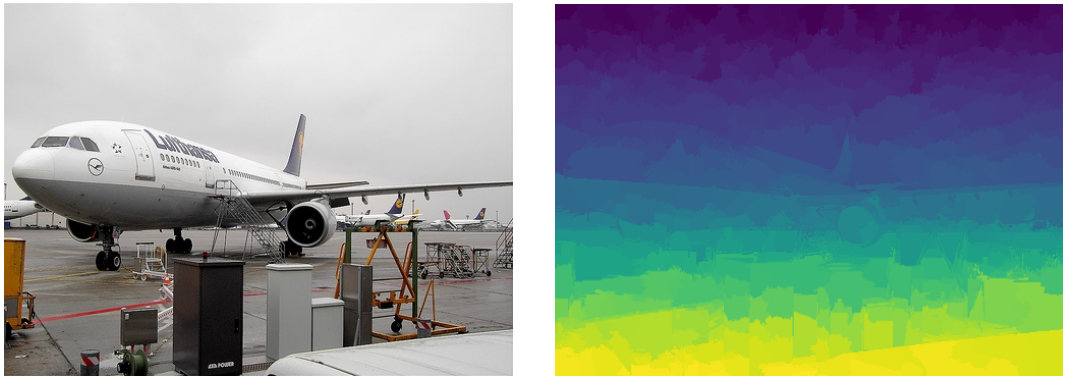


Figure 5.4: An image and its oversegmentation into superpixels.

$\psi_{\text{neighborhood}}$ and ψ_{fill} Two additional constraints ensure a low-level structural consistency of network predictions, namely $\psi_{neighborhood}$ and ψ_{fill}. These have the purpose of propagating information across neighboring pixels and do not require any form of weak annotations. Instead, they represent background prior knowledge.

Concretely, $\psi_{neighborhood}$ specifies that if a pixel has class c, then at least one of its neighbors should have the same class c. The first-order logic formula in this case is:

$$\begin{aligned}\psi_{\text{neighborhood}} = \quad & \forall(i,j,c) \in [1,H] \times [1,W] \times [1,C] \\ & \text{Class}(i,j,c) \rightarrow (\exists(i',j') \in \mathcal{N}(i,j)\ \text{Class}(i',j',c)),\end{aligned} \tag{5.15}$$

where $\mathcal{N}(i,j) = \{(i',j') \in [1,H] \times [1,W] \mid |i'-i| + |j'-j| = 1\}$.

Similarly, ψ_{fill} expresses that "if all neighbors of a pixel have the same class c, then the pixel in the middle should have class c". The first-order logic expression is, thus:

$$\begin{aligned}\psi_{\text{fill}} = \quad & \forall(i,j,c) \in [1,H] \times [1,W] \times [1,C], \\ & (\forall(i',j') \in \mathcal{N}(i,j)\ \text{Class}(i',j',c)) \rightarrow \text{Class}(i,j,c).\end{aligned} \tag{5.16}$$

ψ_{border} When fine-tuning SAM with the five constraints defined above, the network corrects significant mistakes but loses fine details and precision around edges. Therefore, an additional constraint named ψ_{border} attempts to recover the fine structure of the predictions.

More specifically, this constraint leverages an oversegmentation procedure, which separates the image into *superpixels* [70], a grouping of adjacent pixels that are close in terms of color and texture. Figure 5.4 depicts an example. Several algorithms exist for producing such superpixels. This thesis uses the *quickshift* algorithm, with a kernel size of 3 for the Gaussian kernel used for smoothing, a maximum distance of 6 between clusters, and a color-space proximity to image-space proximity ratio of 0.5.

Informally, the constraint imposed with the help of superpixels is that "if two adjacent pixels have different classes, they belong to different superpixels". Optimizing to maximally satisfy this constraint is equivalent to maximally satisfying the constraint "two adjacent pixels belonging to the same superpixel should have the same class" due to the symmetric operations of the proposed logic. Indeed, the FOL formulation below shows this more precisely:

$$\begin{aligned}
\psi_{different_up}(i,j) &= \neg\exists_2 c(Class(i,j,c) \wedge Class(i-1,j,c)) \\
\psi_{different_left}(i,j) &= \neg\exists_2 c(Class(i,j,c) \wedge Class(i,j-1,c)) \\
\phi_{boundary_up}(i,j) &= Superpixel(i,j,s_1) \wedge Superpixel(i-1,j,s_2) \wedge s_1 \neq s_2 \\
\phi_{boundary_left}(i,j) &= Superpixel(i,j,s_1) \wedge Superpixel(i,j-1,s_2) \wedge s_1 \neq s_2 \\
\psi_{border_up} &= \forall (i,j) \in [1,H] \times [1,W] \\
&\quad \psi_{different_up}(i,j) \rightarrow \phi_{boundary_up}(i,j) \\
\psi_{border_left} &= \forall (i,j) \in [1,H] \times [1,W] \\
&\quad \psi_{different_left}(i,j) \rightarrow \phi_{boundary_left}(i,j) \\
\psi_{border} &= \psi_{border_up} \wedge \psi_{border_left}
\end{aligned} \tag{5.17}$$

5.2.6 Learning

As established by the problem definition in Section 4.1, the goal of WSSS is to learn a function f_θ by minimizing a loss function on a dataset $\mathcal{D}$ consisting of images and weak labels (x_i, W_i) using a set of formulas F. This definition links training of neural networks with the concept of fuzzy maximum satisfiability [79], by aiming is to find the parameters:

$$\theta^* = \arg\min_\theta \mathcal{L}(f_\theta | D, F) = \arg\min_\theta \sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), W_i, F). \tag{5.18}$$

An important aspect is, therefore, expressing the loss function for a particular training example (x_i, W_i). Since the logic used operates over log probabilities, the aim is to optimize the negative log-likelihood of each constraint:

$$\mathcal{L}(f_\theta(x_i), W_i, F) = -\sum_{\psi \in F} G(\psi), \tag{5.19}$$

where $G(\psi)$ denotes the log-probability-based truth degree of the formula ψ. Optimizing here refers, naturally, to gradient descent methods available in the literature on neural networks. The fully differentiable logic framework enables end-to-end training.

5.2.7 Outputs

The first stage aims to obtain pseudo-segmentation labels for the training images. Pseudo-labels are a concept introduced in the literature of semi-supervised learning

[45], intended to provide a proxy for the true labels in cases of unlabeled data. In the WSSS literature, they serve a similar purpose - in this case, they represent pixel-wise annotations produced through automatic methods rather than through manual procedures.

In this work, the fine-tuned Segment Anything model generates pseudo-labels via the same mask-merging procedure. This time, instead of stopping at the level of per-channel log probabilities, the mask $\hat{y}$ corresponds to the channel-wise highest (log) probability at each pixel from the $H \times W \times C$ shaped tensor of log probabilities $\hat{z}$:

$$\hat{y}_{i,j} = \arg\max_{c \in \mathcal{C}} \hat{z}_{i,j,c}. \tag{5.20}$$

Some approaches further refine pseudo-labels using DenseCRF [10]. While this may lead to improvements on some datasets, it requires tuning and hides the true quality of the pseudo-labels produced. To maintain a clear assessment of the first stage performance, the network outputs remain unchanged.

5.2.8 Other considerations

The training process uses PyTorch to maximally satisfy the imposed constraints. It runs for 30 epochs using the Adam optimizer with a learning rate of 0.0001 and no weight decay. Since SAM's mask decoder does not support batches of images, a gradient accumulation procedure simulates a batch size of 64 which stabilizes the training in early epochs. Configuration files, such as the one listed in Appendix A, Section A.1, specify the training settings and constraints used. Implementing a constraint requires directly operating on the tensor of log probabilities and possibly weak labels present in the image, as illustrated in Appendix A, Section A.2.

5.3 Second stage: Fully supervised training

This thesis contributes mainly to the first stage of the training process, aiming to produce qualitative pseudo-segmentation labels for the images in the training set. Consequently, their effect becomes visible in the second stage, where a segmentation model receives full supervision from them. Creating accurate pseudo-labels is essential in WSSS, especially in such a two-stage workflow. This is because noisy labels severely degrade the generalization performance of neural networks [75]. This problem is even more prominent for automatically generated labels, which suffer from the implicit erroneous results associated with neural network predictions.

Despite this drawback, the fully supervised training with pseudo-labels remains necessary for reasons enumerated in Section 5.1.2. This thesis adheres to established practices from the literature in selecting both the segmentation model and training pipeline, thereby promoting a fair comparison between methods. Since most WSSS works adopt the same two-stage workflow, they are directly comparable when evaluating the same segmentation network on the same dataset. In this spirit, this thesis aims to train three segmentation networks: **Mask2Former** [15], **DeepLabV2**

[11], and **ConvNeXt-UPerNet** [55, 87]. The choice among them depends on the guidelines established by previous works. DeepLabV2 and ConvNeXt-UPerNet enable a direct comparison with existing WSSS methods, whereas Mask2Former pushes the performance even more.

The MMSegmentation toolbox [61] assists in training the Mask2Former segmentation model. This open-source tool facilitates the development and training of segmentation models, offering out-of-the-box support for a wide range of models and datasets. More concretely, this work uses the same training configuration as other works that adopted Mask2Former [94, 49]. From a model configuration perspective, this training uses the Swin-L [54] backbone. Learning occurs over 80,000 steps, with a batch size of 4 and a learning rate of 1e-4, while all other training settings remain consistent with the Mask2Former configuration in MMSegmentation.

Training a DeepLabV2 model demonstrates the effectiveness of learning from high-quality pseudo-labels while positioning this work more clearly among other approaches in the literature in terms of segmentation accuracy. Instead of utilizing the MMSegmentation toolbox, this training uses a PyTorch re-implementation of DeepLabV2 [36], with its default settings.

To align with previous work in the medical domain, this thesis also trains a ConvNeXt-UPerNet architecture. The network utilizes the ConvNeXt-tiny backbone, which has an input image size of 384x384, and is pre-trained on ImageNet-1k. Other training settings follow the instructions of SimTxtSeg [89], starting from the configuration file of ConvNeXt-UPerNet from the MMSegmentation toolbox.

Chapter 6

Evaluation

This chapter showcases the effectiveness of the proposed method by evaluating it both quantitatively and qualitatively on two different datasets: Pascal VOC 2012 [24] and LGG segmentation [64]. The evaluation follows each level of interest: after the first stage, it sets expectations for the performance of the second-stage training, and after the second-stage training, it produces the segmentation network of interest in the WSSS problem. A comparison of the key findings with those in previous studies highlights the effectiveness of the proposed framework. Ultimately, an ablation study assesses the impact that each constraint has on learning and additionally explores other formulations for the set of fuzzy operators.

6.1 Evaluation metrics

This thesis reports results using the (mean) intersection over union metric used throughout the WSSS literature [49, 52, 77]. Intersection over union (IoU) measures the ratio between the common region of the prediction and ground truth and the total area covered by both [42]. Unlike accuracy, which can be misleading in imbalanced scenarios, IoU is particularly suitable as a metric for segmentation because it accounts for any size ratio between the object of interest and the rest of the image. Given the number of true positives, TP, the number of false positives FP and the number of false negatives FN, the IoU [16] is:

$$IoU = \frac{TP}{TP + FP + FN}. \tag{6.1}$$

In binary segmentation, IoU measures the overlap between the foreground regions of the prediction and target segmentation. In a multi-class setting, the evaluation typically expands to the mean Intersection over Union (mIoU) [24], which corresponds to the average of per-class IoU metrics [42]:

$$mIoU = \frac{1}{C} \sum_{c=1}^{C} IoU_c, \tag{6.2}$$

where C is the number of classes. This time, computing IoU_c follows a binary segmentation paradigm, with class c the positive class and all other classes the background (negative) class.

For Pascal VOC 2012, this thesis reports the mIoU over all 21 classes, including the background class, as per the official evaluation procedure. For the LGG segmentation dataset, which involves binary segmentation, this thesis reports the IoU of the positive class (tumor region). In this case, evaluating the overlap for the tumor class alone is sufficient to assess model performance.

6.2 Pascal VOC 2012

The evaluation first proceeds on the benchmark Pascal VOC 2012 dataset [24] used in the majority of WSSS works. Originally, it contained 4,369 images of 20 object classes (21 classes, including the background), split into training, validation, and testing, with 1,464, 1,449, and 1,456 images, respectively [49]. Researchers later extended the training set to 10,582 images using the SBD dataset [29]. This thesis follows the consensus among WSSS works, which is to use the augmented set for training. The proposed method uses the bounding box annotations provided and the scribble annotations obtained by Lin et al. [50]. The original pixel-wise annotations facilitate the evaluation. An important note is that the test set is private, thus obtaining results implies sending the predictions to the official evaluation server. In the following sections, the *train* set refers to the original set of 1,464 images, the *trainaug* set refers to the extended set of training images, while the *val* and *test* sets stand for the original validation and testing sets.

6.2.1 Evaluation of the first stage

To set expectations for the performance of the second-stage network, it is important to evaluate the pseudo-masks produced in the first stage. To this end, Table 6.1 shows the quality of pseudo-labels obtained by this method on the Pascal VOC 2012 *train* set. The proposed method extracts pseudo-labels of higher quality than previous methods by leveraging the additional weak annotation data it is trained on. Specifically, the produced pseudo-masks exhibit a **94.3% mIoU** on the *train* set. This result shows that fine-tuning improves SAM from the 91.5% mIoU obtained by Jiang et al. [32] when selecting the third mask the multi-mask output and the baseline of 90.5% mIoU when selecting the mask with the highest score. Figure 6.1 displays several pseudo-labels alongside the ground truth masks, confirming the high mIoU score, while Figure 6.2 shows some erroneous results. By leveraging bounding box annotations as prompts for SAM, the desired object instances are always present in the segmentation. However, errors include incorrectly segmenting the background co-occurring with classes of interest and imprecisely outlining objects with fine details.

Table 6.2 shows an in-depth look at per-class results. The method corrects significant mistakes and aligns SAM's predictions to the semantics of the dataset. For instance, the initial poor score on the *table* class shows that SAM segments

Method	Annotations	mIoU (%)
CLIP-ES [52]	Image-level, Language	75.0
SemPLeS [49]	Image-level	78.4
VPL [93]	Image-level, Language	80.1
FMA-WSS [94]	Image-level	80.4
Sun et al. [77]	Image-level	88.3
Box2TagBack [31]	Boxes	90.2
Jiang et al. [32]	Image-level	61.9
	Points	71.7
	Scribbles	89.7
	Boxes	91.5
Baseline SAM-huge prompted with boxes	Boxes	90.5
This method	Boxes, Scribbles	**94.3**

TABLE 6.1: Pseudo-label quality comparison on the Pascal VOC 2012 *train* set

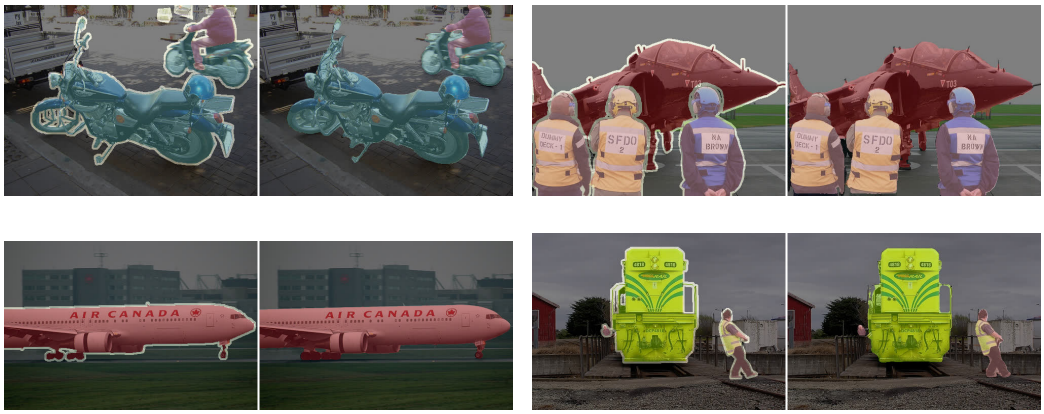


FIGURE 6.1: Visualization of pseudo-masks generated by the fine-tuned SAM. On the left is the ground truth, and on the right is the produced segmentation mask.

the tablecloth and does not include plates or cutlery in the mask, even though the Pascal VOC labels do not make this distinction. Fine-tuning the model corrects these issues, and the result thus significantly improves from 49.7% to 94.7% mIoU. However, refining fine details for classes with a high score or classes such as *bike* is challenging due to the coarse nature of the weak labels.

bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
96.9	97.7	66.2	96.5	94.8	96.8	98.1	96.6	96.4	88.0	96.8	49.7	95.4	97.0	91.2	93.5	77.1	97.7	83.9	97.0	91.6
98.5	96.7	63.9	96.1	93.7	96.7	98.7	97.0	97.9	90.9	96.9	94.7	97.4	97.3	95.1	96.8	91.4	97.7	90.1	97.8	95.9

TABLE 6.2: Per-class IoU (%) of the 21 classes in Pascal VOC 2012. First and second rows correspond to the baseline and fine-tuned versions of SAM, respectively.

Figure 6.2: Visualization of erroneous pseudo-labels produced by SAM. On the left is the ground truth, and on the right is the produced segmentation mask.

In addition to assessing the quality of predictions, the thesis also evaluates how often pseudo-labels satisfy the imposed constraints. More precisely, Table 6.3 shows the constraint accuracy for each constraint ψ that SAM is trained with, i.e., the fraction of results that satisfy constraint ψ. Depending on the constraint, this fraction is *image-wise* (what percentage of images satisfy the constraint entirely), *pixel-wise* (what percentage of pixels satisfy the constraint) or *row/column-wise* (what percentage of rows and columns of bounding boxes satisfy $\psi_{bbox\_tightness}$). For constraints that stem from weak annotations ($\psi_{background}$, $\psi_{bbox\_tightness}$ and $\psi_{scribbles}$), imperfect annotations likely account for the difference to perfect accuracy. For instance, not all bounding boxes are *tight* around the object, directly impacting the training and the constraint accuracy of $\psi_{bbox\_tightness}$ and $\psi_{background}$. Figure 6.3 illustrates this point. For the low-level structural constraints $\psi_{neighborhood}$ and ψ_{fill}, the difference is significant when aggregating at the image level. Lastly, at first glance, the ψ_{border} constraint appears to be ineffective. However, learning fine details is hard in a weakly supervised setting, and the omission of ψ_{border} results in coarse predictions with a ψ_{border} constraint satisfaction of 43.14%, significantly lower than even the baseline.

Constraint	Aggregation level	Baseline (%)	Fine-tuned (%)
$\psi_{background}$	Pixel-wise	99.88	99.96
$\psi_{bbox\_tightness}$	Row/Column-wise	98.15	98.97
$\psi_{scribbles}$	Pixel-wise	93.86	97.62
$\psi_{neighborhood}$	Image-wise	41.49	73.45
ψ_{fill}	Image-wise	48.29	88.69
ψ_{border}	Pixel-wise	54.66	55.95

Table 6.3: Comparison of constraint satisfaction accuracy of the pre-trained SAM and the fine-tuned SAM on Pascal VOC 2012 *trainaug* set

6.3 Evaluation of the second stage

The pseudo-label quality offers valuable insight into the potential performance reached by the second-stage segmentation network. Higher quality correlates with better

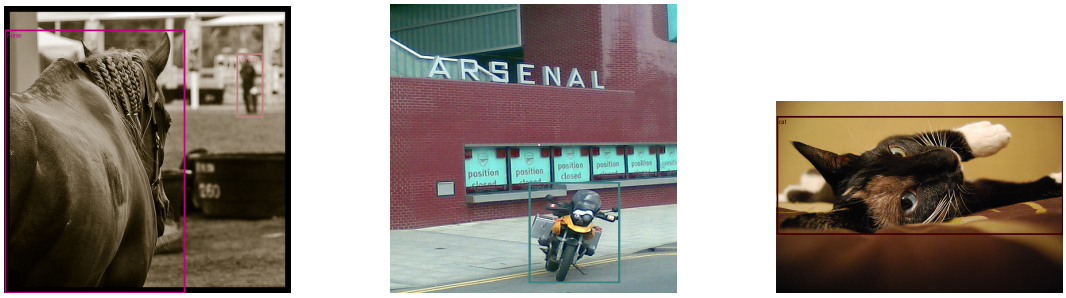


FIGURE 6.3: Examples of *loose* bounding box annotations.

segmentation outcomes. Since the ultimate goal of WSSS is precisely to obtain the second-stage network, it is also essential to evaluate its performance.

Most WSSS methods train the DeepLabV2 model with a ResNet-101 backbone; therefore, this thesis employs this standard. Table 6.4 shows the results in this case. The pseudo-masks help this subsequent training outperform other methods on the test set, with a 79.6% mIoU when using CRF post-processing. This score is close to the 79.7% mIoU obtained by the training supervised by the original pixel-wise annotations.

Method	Annotations	mIoU_{val} (%)	$\text{mIoU}_{\text{test}}$ (%)
Full supervision [11]	Pixel-wise labels	76.3 (77.7*)	79.7*
Scribble hides class [97]	Scribbles	75.3	75.3
TEL [48]	Scribbles	75.2	75.6
AGMM [86]	Scribbles	74.2	75.7
Chan et al. [8]	Scribbles	76.2	-
Box2TagBack [31]	Boxes	76.3	75.8
Sun et al. [77]	Image-level	77.2*	77.1*
CLIP-ES [52]	Image-level, Language	73.8	73.9
CLIP-CPAL [78]	Image-level, Language	74.5	74.7
VPL [93]	Image-level, Language	**79.3***	79.0*
Jiang et al. [32]	Image-level	71.1	72.2
	Points	69.0	68.7
	Scribbles	75.9	76.6
	Boxes	76.3	75.8
This method (DeepLabV2)	Boxes, Scribbles	77.6 (79.1*)	**78.2 (79.6*)**

TABLE 6.4: Comparison of WSSS methods with DeepLabV2 segmentation networks on the Pascal VOC 2012 *val* and *test* sets. "*" denotes CRF post-processing [10].

To reflect the advances in segmentation, several works have shifted toward training networks with a Vision Transformer (ViT) backbone. Following this trend, this thesis

employs Mask2Former [15], a transformer-based segmentation model, and trains it the pseudo-labels produced in the first stage. Table 6.5 shows a comparison with other transformer-based WSSS methods. With an mIoU of 88.4% on the *val* set and 87.9% on the *test* set, this framework outperforms previous techniques. Notably, there is a large margin of +5.0% mIoU on the *test* set over existing methods, mainly because the methods that perform comparatively on the DeepLabV2 benchmark have not trained the Mask2Former model.

Method	Annotations	mIoU_{val} (%)	$\text{mIoU}_{\text{test}}$ (%)
Full supervision	Pixel-wise labels	87.1	86.7
CoSA-MS [95]	Image-level	81.4	78.4
WeakTr [102]	Image-level	78.4	79.0
FMA-WSS [94]	Image-level	82.6	81.6
DHR [33]	Image-level	82.3	82.3
SemPLeS [49]	Image-level	83.4	82.9
This method (Mask2Former)	Boxes, Scribbles	**88.4**	**87.9**

Table 6.5: Comparison of WSSS methods with ViT-based segmentation networks on the Pascal VOC 2012 *val* and *test* set.

Figure 6.4 illustrates a set of masks predicted by the Mask2Former model on *val* set images, alongside the target labels. This second-stage training produces a prompt-free network, but the results generally show a slight decline in quality compared to the pseudo-masks produced by SAM. This aligns with expectations, as Segment Anything is a powerful benchmark on in-the-wild images, whose training does not compare in scale to the second-stage training of Mask2Former. Additionally, these are unseen images, whereas the results of the first stage correspond to an *overfitting* scenario, where the primary aim is to produce qualitative labels for the *trainaug* set seen during training.

Table 6.5 shows that training on the generated pseudo-masks outperforms the fully supervised training using the original labels of the dataset. In theory, this result is unexpected since weakly supervised methods rely on sparse annotations and should not surpass the performance of pixel-wise supervision. However, despite being imperfect, the pseudo-labels are comparable to or better than the manually annotated segmentations. Figure 6.5 illustrates this point by showing the two main factors behind this outcome, namely (1) higher quality contours on the augmented part of the dataset and (2) missing annotations for certain objects.

6.4 LGG segmentation

Medical image segmentation is a common task that typically requires fully annotated segmentation labels. In addition to the time-related costs of annotating, it may require domain expertise. To show the potential of the proposed framework in this scenario, this section performs the same experiments on the LGG Segmentation

FIGURE 6.4: Illustration of final segmentation results of the Mask2Former model. From left to right: image, ground truth, Mask2Former prediction.

FIGURE 6.5: Instances with poor pixel-wise annotations. For each pair, the ground truth is on the left, while the pseudo-label is on the right.

dataset [64]. It consists of 3,064 brain MRI images, along with manual annotations of low-grade gliomas (i.e., benign brain tumors) present in these images. Thus, it is a binary segmentation problem with only two classes - background and tumor. Although the dataset does not come with a predefined split, this thesis adopts the same strategy as SimTxtSeg [89] by splitting the dataset into three sets, with a ratio of 8:1:1 for training (2,451 images), validation (306 images), and testing (307 images).

The dataset does not provide bounding box or scribble annotations. Instead, this thesis derives them automatically from the ground truth segmentation masks. Each image thus receives a single bounding box and a single scribble as annotations. The extreme coordinates (topmost, leftmost, bottommost, and rightmost) of the segmented area determine the bounding box. The scribble aggregates the points in a path between two randomly sampled coordinates on the opposite sides of the contour of the segmentation map, as computed by contour-finding OpenCV algorithm [65].

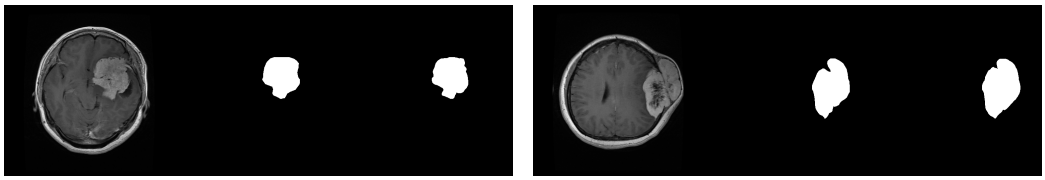


Figure 6.6: Visualization of pseudo-masks generated by the fine-tuned SAM for the LGG training set. Left to right: image, ground truth mask, predicted pseudo-labels.

6.4.1 Evaluation of the first stage

Table 6.6 compares pseudo-label quality on the LGG segmentation dataset [64]. With an 82.3% IoU, the pseudo-labels obtained through this method significantly surpass other WSSS methods. The fine-tuning improves upon the score of 78.0% obtained by the pre-trained SAM model. This improvement is significant, considering that tumors cover small areas of the mask (and thus mistakes are punished more in the IoU computation) and do not have clearly defined edges. To illustrate this point, Figure 6.6 shows a few examples of ground truth labels next to produced pseudo masks.

Method	Annotations	IoU (%)
SimTxtSeg [89]	Language	72.4
Baseline SAM-huge prompted with boxes	Boxes	78.0
This method	Boxes, Scribbles	**82.3**

Table 6.6: Pseudo-label quality on the LGG segmentation *train* set

6.4.2 Evaluation of the second stage

To provide a comparable study on how the pseudo-label quality translates to second-stage training, this thesis adopts the ConvNeXt-tiny backbone used in SimTxtSeg [89]. Although the decoder differs due to the absence of language input, the training setup follows the regime established in SimTxtSeg.

Table 6.7 displays the comparison between this work and SimTextSeg and other medical WSSS works. Again, with an IoU of 73.6% on the test set, this method outperforms existing methods. Here, the larger difference between this WSSS method and full supervision, which obtained 77.7% IoU, suggests that pseudo-labels fall short in quality compared to ground-truth annotations, mainly due to the limited effectiveness of foundational models such as SAM in underrepresented domains.

Method	Annotations	IoU (%)
Full supervision (ConvNeXt)	Pixel-wise labels	77.7
WeakPolyp [83]	Boxes	63.4
BoxPolyp [84]	Boxes	67.4
SimTxtSeg [89]	Language	71.3
This method (ConvNeXt)	Boxes, Scribbles	**73.6**

TABLE 6.7: Comparison of WSSS methods on the LGG segmentation *test* set.

6.5 Ablation study

An ablation study on the Pascal VOC dataset further explains how each constraint affects the first-stage fine-tuning of SAM. To achieve this, the training uses five out of the six constraints, removing one constraint in each experiment. Table 6.8 reports the findings, where F denotes the set of six constraints described in the previous chapter, used in the first-stage training. Scribbles have the largest effect, resulting in a performance decrease of -2.2% without their partial supervision. Figure 6.7 shows that removing ψ_{border} accounts for a loss of quality around object boundaries due to the coarse nature of the learning signals.

Constraints used	mIoU (%)
F	94.37
F $\setminus\{\psi_{background}\}$	94.25 (-0.12)
F $\setminus\{\psi_{bbox\_tightness}\}$	93.80 (-0.57)
F $\setminus\{\psi_{scribbles}\}$	92.28 (-2.09)
F $\setminus\{\psi_{neighborhood}\}$	94.30 (-0.07)
F $\setminus\{\psi_{fill}\}$	94.22 (-0.15)
F $\setminus\{\psi_{border}\}$	93.04 (-1.33)

TABLE 6.8: The effect of removing each constraint, in terms of pseudo-label quality for the Pascal VOC *train* set.

The impact of the $\psi_{background}$ is limited in training SAM because the prompt encoder ensures that predicted masks fit *within* the bounding box prompts. However, it is essential in training an off-the-shelf network in a single stage, as it is the only signal helping the network learn the background class. A one-stage training of a U-Net [72] segmentation network, supervised solely by $\psi_{background}$, $\psi_{bbox\_tightness}$ and $\psi_{scribbles}$ illustrates this point and Figure 6.8 further emphasizes the importance of each constraint derived from weak labels. Training with $F = \{\psi_{background}, \psi_{scribble}\}$ results in coarse masks, while using $F = \{\psi_{background}, \psi_{bbox\_tightness}\}$, produces class activation maps rather than segmentations, showing only the discriminant features of the classes. Training with all three constraints shows how they complement each other in reaching the desired outcome. Appendix D further motivates the choices in

(A) with ψ_{border} (B) without ψ_{border}

Figure 6.7: Visualization of pseudo-masks produced by the trained model with and without the boundary-preserving constraint ψ_{border}.

$\{\psi_{background}, \psi_{scribbles}\}$ $\{\psi_{background}, \psi_{bbox\_tightness}\}$ $\{\psi_{background}, \psi_{scribbles}, \psi_{bbox\_tightness}\}$

Figure 6.8: Predictions of a U-Net model trained with weak supervision, showing the individual contribution of $\psi_{scribbles}$ and $\psi_{bbox\_tightness}$.

the current solution.

Additionally, this ablation study investigates the use of alternative fuzzy operators in place of disjunction and quantifiers. The operators mentioned are those defined in logLTN, namely the Log-Mean-Exp S_{LME} and A^E_{LME}, replacing both the dual T-conorm S_{logP} and the existential aggregator A^E_{logP}, and the batch-size invariant mean aggregator A^U_{mean}, in place of A^U_{logP}. Table 6.9 shows that performance degrades below that of the baseline pre-trained model when using Log-Mean-Exp and only slightly improves when using the mean aggregator. In any case, the current symmetric configuration significantly outperforms these alternatives.

Method	mIoU (%)
Baseline SAM	90.5
This method	94.3
This method with S_{LME} and A^E_{LME}	86.38
This method with A^U_{mean}	91.87

Table 6.9: Ablation study on the fuzzy logic operators used, measuring the quality of pseudo-masks on the Pascal VOC *train* set.

Chapter 7

Conclusion

This final chapter summarizes the work and reflects on whether its goals have been achieved. The chapter then discusses current limitations and, based on these, proposes directions for future work.

7.1 Summary

An in-depth exploration of the related literature showed the main limitations of current weakly supervised segmentation methods. Although WSSS as a topic has received significant attention from researchers, most works derived ad-hoc techniques for dealing with one particular type of weak labels. The few works that explore learning from various forms of weak supervision have only exploited classic weak segmentation labels (i.e., bounding boxes, scribbles, points, and image-level tags) and do not leverage the current advances in the field of segmentation. Other, more recent approaches that leverage large, foundational vision models only focus on image-level or language-level supervision.

To address these limitations, this thesis proposed a framework for learning from a heterogeneous set of weak labels and additional prior knowledge. Designed in two phases, similar to most works in the literature, the proposed method first trains the Segment Anything Model in a weakly supervised setting. Using high-quality pseudo-labels obtained from the fine-tuned Segment Anything Model as ground truth data, the second stage of the process further trains classic, prompt-free segmentation networks. In the first stage, this method uses predicate fuzzy logic to derive differentiable signals from available weak labels and additional prior knowledge expressed as logical constraints. A custom, tensorized implementation of operators with fuzzy semantics in the logarithm space in the spirit of Logic Tensor Networks enables learning at the scale required for segmentation in the context of its high-dimensional output space, i.e., the segmentation mask.

This framework is applied to a benchmark dataset, Pascal VOC 2012, and in a practical setting involving brain tumor segmentation. In the former case, the proposed method established a new state-of-the-art performance of 87.9% mIoU and 79.6% mIoU on the *test* set, using Mask2Former and DeepLabV2 as second-stage

segmentation networks, respectively. This approach also excelled in the medical domain, achieving a 73.6% mIoU when utilizing the ConvNeXt-UPerNet network.

The study aimed to understand and present the key empirical insights. Among these are (1) outperforming full supervision and (2) quantifying the impact of learning from constraints that express prior knowledge. Concerning the first point, the previous chapter illustrated the quality of the pseudo-labels, which, in part, surpasses that of manually annotated masks in the Pascal VOC dataset. Regarding learning from prior knowledge, the ψ_{border} constraint is especially impactful by preventing a degrading performance along object boundaries, given the coarse nature of weak annotations.

7.2 Limitations

While not a limitation in the strictest sense, the primary caveat of the approach is its greater reliance on annotations compared to other WSSS works. While other methods, especially those relying on the powerful, foundational vision models CLIP and SAM, mainly exploit image-level tags and use language supervision, this method *requires* bounding box labels for effective SAM prompting and *greatly benefits* from scribble annotations, as shown by the ablation study. Thus, it perfectly represents the trade-off between segmentation accuracy and annotation effort.

A limitation of existing WSSS methods also present in this work is the lack of a stopping criterion or an indicator for checkpoint selection in the first stage of training. Due to working in a weakly supervised setting, there are no full pixel-wise labels; therefore, it is challenging to evaluate the performance. However, in practice, this limitation is rarely an issue, as a small set of images labeled pixel-wise can serve as evaluation during training in the first stage. An alternative consists of monitoring the evolution of the fuzzy loss on a validation set. However, this does not perfectly align with the objective of the first stage, which is to *overfit* on the training set instead of *generalize* to a validation set. The empirical experiments conducted with the ψ_{border} constraint show no decline in segmentation accuracy, suggesting that a possibility is *training for as long as possible* within the allocated time budget.

Finally, while this framework *can* express constraints to learn from image-level tags or points, the experiments shown in this work do not focus on this set of weak labels. This is because such labels provide only sparse information about object presence and location, which translates to less informative constraints (e.g., $\exists(i,j)\ Class(i,j,c)$ for an image-level tag annotation with class c). In contrast, existing weakly supervised methods for these types of weak labels rely on class activation maps (CAM), which offer dense localization and expansion information. In this sense, this thesis meets the first objective set in Section 4.2 only *partly*, as it can express constraints based on image-level tags and points but does not evaluate how effective these sparse signals are for learning.

7.3 Future work

A direction of future work includes improving the current fuzzy operators. This work also included testing other operators, such as the LogMeanExp operator for disjunction or using the mean instead of the sum for the universal quantifier [3]. These attempts did not improve the results. However, further investigation into these or other fuzzy operators may yield a performance boost.

Additionally, exploring a dataset such as CLEVR [35] can enable expressing and learning from other types of weak labels (e.g., spatial relationship constraints such as "the red cube is to the right of the blue cube"). This can help in understanding the entire scope of the proposed system, with the possibility of learning from even more types of weak signals expressed as first-order logic.

Ultimately, regarding SAM, a question is whether the two-stage approach was indeed necessary. Future work should assess a single-stage training approach that retains only the pre-trained image encoder and trains a custom decoder from scratch, thereby removing the dependency on prompts. Moreover, there is a question of whether SAM-predicted *masks* indeed improve quality-wise, or the method instead improves the alignment of mask *scores*. Lastly, freezing the image and prompt encoders proved to be the right choice for this thesis, enabling rapid development and experimentation. However, with the assumption that more training resources are available, *additive* parameter-efficient fine-tuning methods, such as low-rank adaptation methods (LoRA) [28], might lead to improved segmentation quality.

Appendices

Appendix A

First stage training details

A.1 Configuration file for SAM fine-tuning on Pascal VOC 2012

```
DATASET:
  PATH: "./datasets/pascal_voc_2012"

TRAIN_SETUP:
  CONSTRAINTS: [
    "boxes",
    "scribbles",
    "background",
    "neighborhood",
    "fill",
    "borders"
  ]
  BATCH_SIZE: 64
  EPOCHS: 30
  VALID_EVERY: 1
  LEARNING_RATE: 0.0001
  WEIGHT_DECAY: 0.0
  EXPERIMENTS_DIR: "./experiments"

MODEL:
  N_CLASSES: 21
```

A.2 Example of specifying and implementing a constraint

As an example of implementing a constraint, the following code illustrates the implementation of the $\psi_{neighborhood}$ constraint, building upon the framework of

Flinkow et al. [27]:

Listing A.1: $\psi_{neighborhood}$ implementation

```
import torch
import torch.nn.functional as F
from typing import Callable

from differentiable_logics.logic import Logic
from constraints.constraint import Constraint


class NeighbourConstraint(Constraint):
    def __init__(self, device: torch.device, eps: float):
        super().__init__(device, eps)

    def get_constraint(
        self,
        log_probs: torch.Tensor,
        _labels: list,
    ) -> Callable[[Logic], torch.Tensor]:
        fill_value = -float('inf')

        def constraint_fn(logic: Logic) -> torch.Tensor:
            up = F.pad(log_probs, (0, 0, 1, 0), value=fill_value)[:, :, :-1, :]
            down = F.pad(log_probs, (0, 0, 0, 1), value=fill_value)[:, :, 1:, :]
            left = F.pad(log_probs, (1, 0, 0, 0), value=fill_value)[:, :, :, :-1]
            right = F.pad(log_probs, (0, 1, 0, 0), value=fill_value)[:, :, :, 1:]
            neighbors = torch.stack([up, down, left, right], dim=-1)
            neighbor_exists = logic.EXIS_AGG(neighbors, dim=-1)
            impl = logic.IMPL(log_probs, neighbor_exists)
            return logic.UNIV_AGG(impl)

        return constraint_fn
```

Appendix B

More results on the Pascal VOC 2012 dataset

FIGURE B.1: More pseudo-masks for the Pascal VOC *trainaug* set. On the left, the ground truth segmentation, and on the right, the pseudo-mask produced by SAM.

Figure B.2: More results on Pascal VOC *val* set using Mask2Former. Left to right: image, ground truth segmentation, predicted mask.

Appendix C

More results on the LGG segmentation dataset

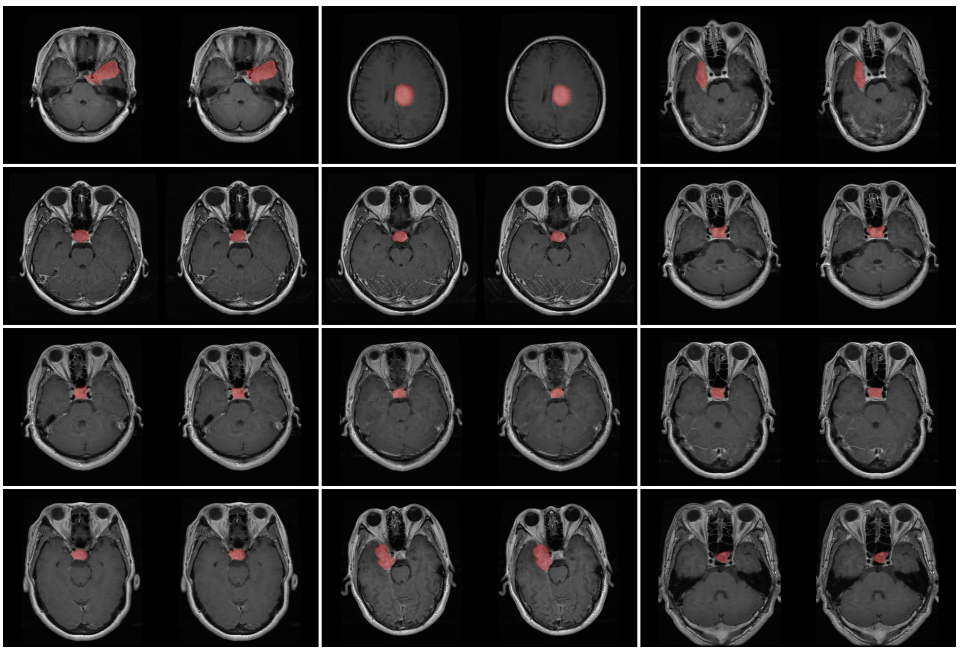


FIGURE C.1: More pseudo-masks for the LGG segmentation *train* set. On the left, the ground truth segmentation, and on the right, the pseudo-mask produced by SAM.

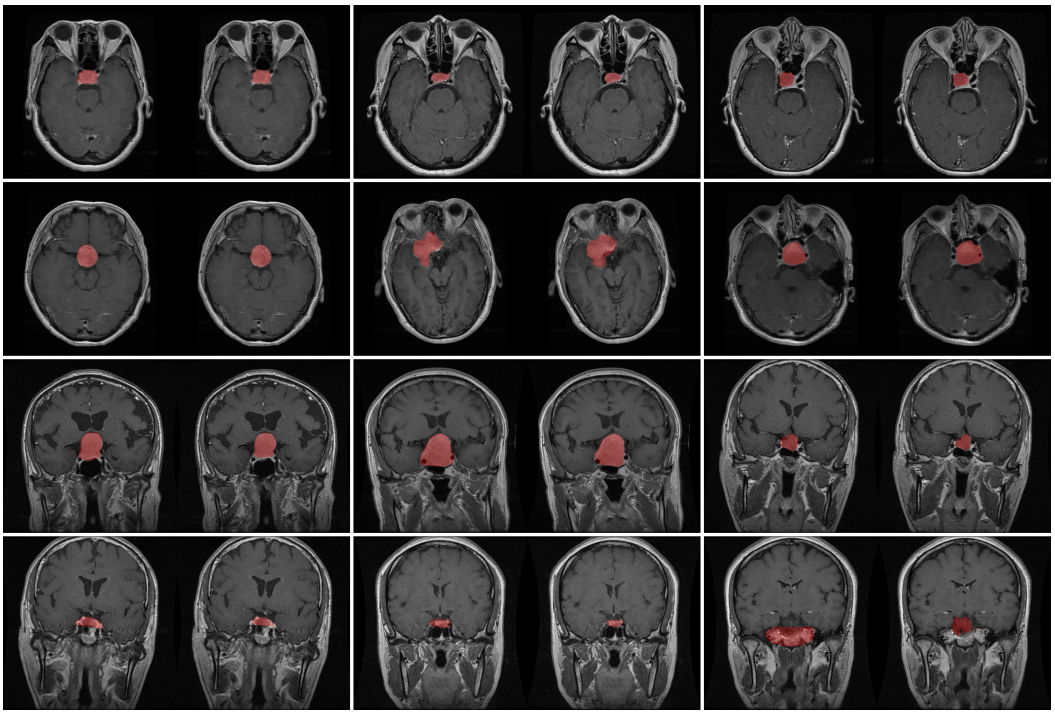


Figure C.2: More results on the LGG segmentation *test* set using ConvNext-UPerNet. On the left, the ground truth segmentation, and on the right the predicted mask.

Appendix D

Single stage U-Net experiments

Research for this thesis included first gathering intuitions about the use of weak supervision in segmentation. Before leveraging recent advances in segmentation through the use of SAM, the experiments involved training a simple U-Net segmentation model in a single stage on a simpler problem - examining only the background and airplane classes of Pascal VOC. Rather than constructing losses with fuzzy operators, this training used probabilistic logic to compute the log probability of success of a large (propositional) formula involving the network outputs as variables. Rather than expressing $\psi_{bbox\_tightness}$ on both rows and columns, it used only rows in the computation. Additionally, rows already accounted for by scribbles were excluded from this computation. In short, this setting aimed to perform a probabilistic inference using closed-formulas rather than WMC, exploiting tensorized operations for a greater scalability to the requirements of segmentation.

Additional results Figure D.1 shows a few more examples of segmentation masks predicted by the U-Net model, after training with $\psi_{background}$, $\psi_{scribbles}$, and $\psi_{bbox\_tightness}$. For an off-the-shelf, untrained network, this result shows the capability of the model to learn in a weakly-supervised setting. Quantitatively, the weakly supervised training achieved an F1 score of 80.6%, while a similar fully supervised training obtained 85.1%.

Experimenting with a volume constraint An interesting constraint in the problem of weakly supervised segmentation is a volume prior, such as "this image is at least 20% filled with a dog.". A learning signal obtained using the dynamic programming counting method proposed by Shukla et al. [74] helps training with such a constraint. More precisely, this loss term can be expressed as:

$$logsumexp_{s=th}^{n} log\ p(\sum_{i=1}^{k} \hat{y}_i = s), \tag{D.1}$$

where n is the total number of pixels in the image, th is a threshold computed based on the number of pixels (e.g., for the 20% threshold, $th = 0.2 \cdot n$) and $log\ p(\sum_{i=1}^{k} \hat{y}_i = s)$ expresses the log probability that exactly s pixels are true (with true here meaning

Figure D.1: Additional results on Pascal VOC *val*. From left to right: image, ground truth segmentation, prediction of the U-Net model.

they belong to the positive class airplane). Although relatively slow, the volume constraint with thresholds of 20%, 40%, 60%, and 80% (more refined intervals are impractical for a human to annotate) improved the results to 81.6% F1. For efficiency reasons, and because this constraint cannot be easily expressed in first-order logic, it was not included in further experiments including the training of SAM.

Switching to fuzzy logic Because of the disjoint-sum problem, computing the success probability for a FOL formula does not generally have a closed-form and easy-to-vectorize equation. Computing the success probability of the bounding box tightness constraint expressed both on rows and columns requires a general-purpose solution for WMC, which does not scale to sizes of bounding boxes relevant to segmentation. In this sense, and for expressing other constraints, the focus shifted to fuzzy operators. Thus, learning from both rows and columns improved the F1 score from 80.6% to 81.2%, even though the loss signals involved no longer represented exact probabilistic inference. This, along with the possibility of expressing more complex constraints and background knowledge, warranted the use of fuzzy logic going forward.

Bibliography

[1] B. Adhikari, J. Peltomaki, J. Puura, and H. Huttunen. Faster bounding box annotation for object detection in indoor scenes. In *2018 7th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6, 2018.

[2] S. Badreddine, A. dAvila Garcez, L. Serafini, and M. Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, Feb. 2022.

[3] S. Badreddine, L. Serafini, and M. Spranger. logltn: Differentiable fuzzy logic in the logarithm space, 2023.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.

[5] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision, 2016.

[6] D. R. Berger, H. S. Seung, and J. W. Lichtman. Vast (volume annotation and segmentation tool): efficient manual and semi-automatic labeling of large 3d image stacks. *Frontiers in neural circuits*, 12:88, 2018.

[7] S. Cakir, M. Gau, K. Hppeler, Y. Ounajjar, F. Heinle, and R. Marchthaler. Semantic segmentation for autonomous driving: Model evaluation, dataset generation, perspective comparison, and real-time capability, 2022.

[8] G. Chan, P. Zhang, H. Dong, S. Ji, and B. Chen. Scribble-supervised semantic segmentation with prototype-based feature augmentation. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6155–6169. PMLR, 21–27 Jul 2024.

[9] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs, 2016.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

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

[13] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. 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.

[14] Z. Chen and Q. Sun. Weakly-supervised semantic segmentation with image-level labels: from traditional models to foundation models, 2024.

[15] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation, 2022.

[16] Y.-J. Cho. Weighted intersection over union (wiou) for evaluating image segmentation. *Pattern Recognition Letters*, 185:101107, Sept. 2024.

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

[18] G. Csurka, R. Volpi, and B. Chidlovskii. Semantic image segmentation: Two decades of research, 2023.

[19] G. B. de Carvalho and J. Almeida. Exploiting the segment anything model (sam) for lung segmentation in chest x-ray images, 2024.

[20] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[21] M. Diligenti, M. Gori, M. Maggini, and L. Rigutini. Bridging logic and kernel machines. *Machine learning*, 86:57–88, 2012.

[22] M. Diligenti, M. Gori, and V. Scoca. Learning efficiently in semantic based regularization. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II 16*, pages 33–46. Springer, 2016.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[24] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

[25] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong. 3d deep shape descriptor. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2319–2328, 2015.

[26] M. Fischer, M. Balunovic, D. Drachsler-Cohen, T. Gehr, C. Zhang, and M. Vechev. DL2: Training and querying neural networks with logic. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1931–1941. PMLR, 09–15 Jun 2019.

[27] T. Flinkow, B. A. Pearlmutter, and R. Monahan. Comparing differentiable logics for learning with logical constraints. *Science of Computer Programming*, 244:103280, 2025.

[28] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[29] B. Hariharan, P. Arbelez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998, 2011.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[31] Z. Ji and O. Veksler. Weakly supervised semantic segmentation: From box to tag and back. In *British Machine Vision Conference*, 2021.

[32] P.-T. Jiang and Y. Yang. Segment anything is a good pseudo-label generator for weakly supervised semantic segmentation, 2023.

[33] S. Jo, F. Pan, I.-J. Yu, and K. Kim. Dhr: Dual features-driven hierarchical rebalancing in inter- and intra-class regions for weakly-supervised semantic segmentation, 2024.

[34] S. Jo and I.-J. Yu. Puzzle-cam: Improved localization via matching partial and full features. In *2021 IEEE International Conference on Image Processing (ICIP)*, page 639643. IEEE, Sept. 2021.

[35] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, 2016.

[36] kazuto1011. DeepLab with PyTorch. https://github.com/kazuto1011/deeplab-pytorch. GitHub repository.

[37] T.-W. Ke, J.-J. Hwang, and S. X. Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning, 2021.

[38] H. Kervadec, J. Dolz, M. Tang, E. Granger, Y. Boykov, and I. Ben Ayed. Constrained-cnn losses for weakly supervised segmentation. *Medical Image Analysis*, 54:8899, May 2019.

[39] H. Kervadec, J. Dolz, S. Wang, E. Granger, and I. B. Ayed. Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision, 2020.

[40] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollr. Panoptic segmentation, 2019.

[41] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollr, and R. Girshick. Segment anything, 2023.

[42] M. Knott, D. Odion, S. Sontakke, A. Karwa, and T. Defraeye. Weakly supervised image segmentation for defect-based grading of fresh produce. *arXiv preprint arXiv:2411.16219*, 2024.

[43] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation, 2016.

[44] O. Kuzelka. Weighted first-order model counting in the two-variable fragment with counting quantifiers, 2020.

[45] D.-H. Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[46] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *2009 IEEE 12th International Conference on Computer Vision*, pages 277–284, 2009.

[47] Y. Li, X. Yu, and N. Koudas. Data acquisition for improving machine learning models. *Proceedings of the VLDB Endowment*, 14(10):18321844, June 2021.

[48] Z. Liang, T. Wang, X. Zhang, J. Sun, and J. Shen. Tree energy loss: Towards sparsely annotated semantic segmentation, 2022.

[49] C.-S. Lin, C.-Y. Wang, Y.-C. F. Wang, and M.-H. Chen. Semantic prompt learning for weakly-supervised semantic segmentation, 2025.

[50] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation, 2016.

[51] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollr. Microsoft coco: Common objects in context, 2015.

[52] Y. Lin, M. Chen, W. Wang, B. Wu, K. Li, B. Lin, H. Liu, and X. He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation, 2023.

[53] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024.

[54] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[55] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s, 2022.

[56] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation, 2015.

[57] Z. Luo, W. Yang, Y. Yuan, R. Gou, and X. Li. Semantic segmentation of agricultural images: A survey. *Information Processing in Agriculture*, 11(2):172–186, 2024.

[58] R. Manhaeve, S. Dumani, A. Kimmig, T. Demeester, and L. D. Raedt. Deepproblog: Neural probabilistic logic programming, 2018.

[59] R. A. McEver and B. S. Manjunath. Pcams: Weakly supervised semantic segmentation using point supervision, 2020.

[60] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey, 2020.

[61] MMSegmentation Contributors. OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation, July 2020. GitHub repository.

[62] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[63] M. Mchler. Accurately computing log(1 exp(|a|)) assessed by the rmpfr package, 09 2015.

[64] T. C. G. A. R. Network. Comprehensive, integrative genomic analysis of diffuse lower-grade gliomas. *New England Journal of Medicine*, 372(26):2481–2498, 2015.

[65] Opencv. Open source computer vision library. https://github.com/opencv/opencv, 2015.

[66] Y. Ouassit, S. Ardchir, M. Ghoumari, and M. Azouazi. A brief survey on weakly supervised semantic segmentation. *International Journal of Online and Biomedical Engineering (iJOE)*, 18:83–113, 07 2022.

[67] A. Pelez-Vegas, P. Mesejo, and J. Luengo. A survey on semi-supervised semantic segmentation, 2023.

[68] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[69] M. Rajchl, M. C. H. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz, and D. Rueckert. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks, 2016.

[70] Ren and Malik. Learning a classification model for segmentation. In *Proceedings ninth IEEE international conference on computer vision*, pages 10–17. IEEE, 2003.

[71] D. Rolnick, A. Veit, S. Belongie, and N. Shavit. Deep learning is robust to massive label noise, 2018.

[72] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[73] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85117, Jan. 2015.

[74] V. Shukla, Z. Zeng, K. Ahmed, and G. V. den Broeck. A unified approach to count-based weakly-supervised learning, 2023.

[75] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from noisy labels with deep neural networks: A survey, 2022.

[76] J. Sun, Y. Shi, Y. Gao, L. Wang, L. Zhou, W. Yang, and D. Shen. Interactive medical image segmentation via point-based interaction and sequential patch learning, 2018.

[77] W. Sun, Z. Liu, Y. Zhang, Y. Zhong, and N. Barnes. An alternative to wsss? an empirical study of the segment anything model (sam) on weakly-supervised semantic segmentation problems, 2023.

[78] F. Tang, Z. Xu, Z. Qu, W. Feng, X. Jiang, and Z. Ge. Hunting attributes: Context prototype-aware learning for weakly supervised semantic segmentation, 2024.

[79] E. van Krieken, E. Acar, and F. van Harmelen. Analyzing differentiable fuzzy logic operators. *Artificial Intelligence*, 302:103602, Jan. 2022.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[81] I. A. Vezakis, K. Georgas, D. Fotiadis, and G. K. Matsopoulos. Effisegnet: Gastrointestinal polyp segmentation through a pre-trained efficientnet-based network with a simplified decoder, 2024.

[82] J. Wang and B. Xia. Weakly supervised image segmentation beyond tight bounding box annotations, 2023.

[83] J. Wei, Y. Hu, S. Cui, S. K. Zhou, and Z. Li. Weakpolyp: You only look bounding box for polyp segmentation, 2023.

[84] J. Wei, Y. Hu, G. Li, S. Cui, S. K. Zhou, and Z. Li. Boxpolyp:boost generalized polyp segmentation using extra coarse bounding box annotations, 2022.

[85] J. Wu, W. Ji, Y. Liu, H. Fu, M. Xu, Y. Xu, and Y. Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation, 2023.

[86] L. Wu, Z. Zhong, L. Fang, X. He, Q. Liu, J. Ma, and H. Chen. Sparsely annotated semantic segmentation with adaptive gaussian mixtures. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15454–15464, 2023.

[87] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding, 2018.

[88] J. Xie, X. Hou, K. Ye, and L. Shen. Cross language image matching for weakly supervised semantic segmentation, 2022.

[89] Y. Xie, T. Zhou, Y. Zhou, and G. Chen. Simtxtseg: Weakly-supervised medical image segmentation with simple text cues, 2024.

[90] J. Xu, A. G. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3781–3790, 2015.

[91] J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. V. den Broeck. A semantic loss function for deep learning with symbolic knowledge, 2018.

[92] L. Xu, Y. Chen, R. Huang, F. Wu, and S. Lai. Structured click control in transformer-based interactive segmentation, 2024.

[93] Z. Xu, F. Tang, Z. Chen, Y. Su, Z. Zhao, G. Zhang, J. Su, and Z. Ge. Toward modality gap: Vision prototype learning for weakly-supervised semantic segmentation with clip, 2024.

[94] X. Yang and X. Gong. Foundation model assisted weakly supervised semantic segmentation, 2023.

[95] X. Yang, H. Rahmani, S. Black, and B. M. Williams. Weakly supervised co-training with swapping assignments for semantic segmentation, 2024.

[96] L. Zhang, R. Tanno, M. Xu, Y. Huang, K. Bronik, C. Jin, J. Jacob, Y. Zheng, L. Shao, O. Ciccarelli, et al. Learning from multiple annotators for medical image segmentation. *Pattern Recognition*, 138:109400, 2023.

[97] X. Zhang, L. Zhu, H. He, L. Jin, and Y. Lu. Scribble hides class: Promoting scribble-based weakly-supervised semantic segmentation with its class label. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):73327340, Mar. 2024.

[98] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021.

[99] H. Zheng, Y. Zhang, L. Yang, C. Wang, and D. Z. Chen. An annotation sparsification strategy for 3d medical image segmentation via representative selection and self-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6925–6932, Apr. 2020.

[100] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization, 2015.

[101] T. Zhou, W. Xia, F. Zhang, B. Chang, W. Wang, Y. Yuan, E. Konukoglu, and D. Cremers. Image segmentation in foundation model era: A survey, 2024.

[102] L. Zhu, Y. Li, J. Fang, Y. Liu, H. Xin, W. Liu, and X. Wang. Weaktr: Exploring plain vision transformer for weakly-supervised semantic segmentation, 2023.