BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Optimizing CAM Refinement with Swin-based Feature Affinity for Weakly Supervised Semantic Segmentation

Anika Farzana
200041204
K. M. Abesh Ahsan
200041225
Sayemah Amin
200041234

Department of Computer Science and Engineering
Islamic University of Technology
September, 2025

Optimizing CAM Refinement with Swin-based Feature Affinity for Weakly Supervised Semantic Segmentation

Anika Farzana
200041204
K. M. Abesh Ahsan
200041225
Sayemah Amin
200041234

Department of Computer Science and Engineering
Islamic University of Technology
September, 2025

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Anika Farzana**, **K. M. Abesh Ahsan**, and **Sayemah Amin** under the supervision of **Dr. Md. Hasanul Kabir**, Professor, Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Dr. Md. Hasanul Kabir
Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: September 29, 2025

Anika Farzana
Student ID: 200041204
Date: September 29, 2025

K. M. Abesh Ahsan
Student ID: 200041225
Date: September 29, 2025

Sayemah Amin
Student ID: 200041234
Date: September 29, 2025

Dedicated to our parents, whose constant love, guidance, and support have been the foundation of our achievements and the inspiration behind our efforts.

Contents

Abstract **xi**

1 Introduction **1**
1.1 Semantic Segmentation 1
1.1.1 Fully Supervised Semantic Segmentation 3
1.1.2 Weakly Supervised Semantic Segmentation 4
1.2 Motivation 6
1.3 Problem Statement 7
1.4 Challenges of WSSS 7
1.5 Contribution 8
1.6 Organization 9

2 Related Works **10**
2.1 Fully Supervised Semantic Segmentation 10
2.1.1 Early Approaches 10
2.1.2 Convolutional Neural Networks (CNNs) 11
2.1.3 Transformers 13
2.1.4 Problem with Fully Supervised Semantic Segmentation 15
2.2 Weakly Supervised Semantic Segmentation 15
2.2.1 Types of WSSS 16
2.2.2 Image-Level Label Based Weak Supervision 16
2.2.3 Class Activation Maps 17
2.3 Component-Wise Literature Review of WSSS Models 23
2.3.1 Feature Extraction From Backbone 23
2.3.2 CAM Generation 24
2.3.3 Pseudo Label Generation 25
2.3.4 Segmentation Prediction 26
2.3.5 Refinement of Pseudo Labels 28

2.4 WSSS Training Approaches 29
2.4.1 Multistage 29
2.4.2 Single Stage 30

3 Proposed Methodology **31**
3.1 Overview 31
3.2 Feature Extraction from Backbone 33
3.2.1 Contrastive Language-Image Pre-Training (CLIP) 33
3.2.2 CLIP Architecture 34
3.2.3 Characteristics of CLIP 34
3.2.4 Unified Contrastive Learning (UniCL) 35
3.2.5 Swin Transformer 37
3.3 CAM Generation 39
3.4 Segmentation Prediction 40
3.4.1 Encoder Feature Aggregation 40
3.4.2 Generating Final Prediction 41
3.5 CAM Refinement and Pseudo-Label Generation 41
3.5.1 Affinity based CAM Refinement 41
3.5.2 Extracting Affinity Map from the Decoder 43
3.5.3 Selecting Affinity Maps of the Image Encoder 44
3.5.4 Utilizing the Frozen Encoder Affinity Maps 44
3.5.5 Constructing the Semantic Transition Matrix 45
3.5.6 Random Walk Propagation 45
3.5.7 Pixel Adaptive Refinement module 46
3.5.8 Pseudo-Label Generation 48
3.6 Loss Function 48
3.6.1 Affinity Loss 48
3.6.2 Segmentation Loss 49

4 Results and Discussion **50**
4.1 Data and Experimental Setup 50
4.1.1 Dataset 50
4.1.2 Experimental Setup 50
4.1.3 Evaluation Metric 51
4.2 Quantitative Analysis 51
4.2.1 Per-Class Performance Analysis 52
4.2.2 Comparison with Exisiting Methods 55
4.2.3 Summary of Quantitative Analysis 58

4.3 Qualitative Analysis . . . 58
4.3.1 Observations . . . 59
4.3.2 Summary of Qualitative Analysis . . . 60
4.4 Ablation Study . . . 61
4.4.1 Quantitative Study . . . 61
4.4.2 Qualitative Study . . . 62
4.5 Interpretation of Results . . . 64
4.6 Strengths of the Approach . . . 68
4.6.1 Multi-Modal Learning and Unified Objectives . . . 68
4.6.2 Hierarchical Feature Extraction with Swin Transformer . . . 68
4.6.3 Affinity-Based Refinement for Semantic Consistency . . . 68
4.6.4 Efficiency and Scalability . . . 69
4.6.5 Overall Impact . . . 69
4.7 Limitations and Challenges . . . 69
4.7.1 Architectural Constraints . . . 69
4.7.2 Pseudo-Label Generation Limitations . . . 70
4.7.3 Practical Training Challenges . . . 71
4.8 Key Findings . . . 71
4.9 Other Explored Approaches . . . 73

5 Conclusion **76**

References **78**

List of Figures

1.1 Original image and Segmented Image. Two object classes are present in this image: *Person* and *Motorbike*. 1
1.2 Pipeline of the fully supervised semantic segmentation framework, where ground-truth pixel-level annotations are used directly to train the segmentation model. 4
1.3 Pipeline of the WSSS framework illustrating CAM generation, refinement, and pseudo-label creation for segmentation training. Image-level labels serve as weak supervision to generate Class Activation Maps (CAMs), which are subsequently refined into pseudo-labels for training the segmentation network. 5

2.1 Fully Convolutional Network (FCN) for semantic segmentation (adapted from [33]). 11
2.2 Architecture of DeepLabv3+ (adapted from [11]). 12
2.3 An illustration of the SegNet architecture (adapted from [6]). 13
2.4 The Segmenter framework (adapted from [46]). 14
2.5 An illustration of the SegNet architecture (adapted from [54]). 15
2.6 Basic Classification 18
2.7 CAM Generation Process 19
2.8 Grad-CAM Generation Process 20
2.9 Overall architecture of AffinityNet (adapted from [4]). 23
2.10 Overall architecture of WeCLIP (adapted from [61]). 24
2.11 AFA framework for WSSS (adapted from [42]). 26
2.12 Overall framework of ToCo (adapted from [43]). 27
2.13 Overview of CLIP-ES framework (adapted from [31]). 28

3.1 Architecture of *UniCL-AffSeg* 32
3.2 Comparison of Swin Transformer [32] and ViT [15] Architectures. Image taken from [32]. 38

3.3 Original image and CAM generated by our method for class *Bird*. . . . 39
3.4 CAM Generation Process using UniCL 40
3.5 Decoder and Refinement Module 42
3.6 CAM refinement by exploiting affinity map 46
3.7 Ground Truth Label and Pseudolabel generated by our method for class *Bird*. 47

4.1 Pascal VOC 2012 class color map for visualization. Each class is represented by a unique color for easy identification in segmentation maps. 51
4.2 Qualitative comparison of CAMs between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *val* set. 56
4.3 Qualitative comparison of CAMs between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *test* set. 57
4.4 Qualitative ablation on the Pascal VOC dataset showing the effect of the Refinement Module (RFM) [61] using our method. Each pair compares pseudo-labels generated without (left) and with (right) RFM. The RFM enhances boundary precision and recovers missing object regions, leading to more complete and accurate segmentation masks. 63
4.5 Qualitative comparison of pseudo-labels between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *val* set. 65
4.6 Qualitative comparison of pseudo-labels between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *test* set. 66
4.7 Qualitative comparison of pseudo-labels between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *val* and *test* set for person and chair classes. 67
4.8 Comparison of CAM visualization methods on the Pascal VOC dataset for single class scenario. Columns show Grad-CAM, Layer-CAM, and Grad-CAM++ respectively. Each row corresponds to a different example image. 72
4.9 Comparison of Grad-CAM, Layer-CAM, and Grad-CAM++ visualizations for multiple multi-class images from the Pascal VOC dataset. Each subfigure shows CAMs for two distinct object classes within the same image. 74

List of Tables

4.1 Per-Class IoU Performance on PASCAL VOC (Validation vs Test Sets) 53

4.2 Overall Performance Metrics on PASCAL VOC 53

4.3 Comparison of multi-stage and single-stage weakly supervised semantic segmentation methods on the PASCAL VOC 2012 validation and test sets (mIoU, %). Here, **I** denotes image-level supervision, **L** denotes language supervision, and **S** denotes saliency supervision. Our method (**UniCL-AffSeg**) employs Swin-B as the backbone or image encoder. Without description, all have used Dense CRF during inference. 54

4.4 Per-Class IoU Comparison on PASCAL VOC: Without vs With RFM . 61

Acknowledgement

I am profoundly grateful to my supervisor, Dr. Md. Hasanul Kabir, Professor, Department of Computer Science and Engineering, for his outstanding guidance, support, and feedback throughout this research journey. His expertise and encouragement have been pivotal in shaping the direction and quality of this thesis.

His patience and dedication to his work have inspired me to aim for excellence and approach challenges with curiosity and determination. His constructive feedback and insightful suggestions have significantly deepened my understanding of the domain and inspired me to think critically and creatively.

It has been an honor to work under his supervision, and I deeply appreciate his invaluable time and effort in guiding me through this research.

Abstract

Semantic segmentation is a fundamental computer vision task requiring pixel-level understanding of images. While fully supervised methods achieve high accuracy, they rely on costly pixel-level annotations. Weakly Supervised Semantic Segmentation (WSSS) mitigates this by using weaker supervision, such as image-level labels, to train effective models. Recent WSSS progress leverages Class Activation Maps (CAMs), though their sparsity and poor boundary localization remain challenges. This study enhances CAM quality through multi-modal backbones like UniCL and hierarchical transformers such as Swin Transformer for stronger feature extraction, coupled with an affinity-based framework that fuses encoder and decoder affinities for semantically coherent pseudo-labels. A Pixel-Adaptive Refinement (PAR) module further improves object boundaries using local similarity cues. Experiments on the PASCAL VOC 2012 dataset yield mean IoUs of 50.3% (validation) and 50.8% (test), with strong performance on large, distinctive classes but weaker results for small or human-centric ones due to CAM bias and dataset imbalance. Overall, our findings demonstrate that UniCL and Swin Transformer significantly improve CAM quality and segmentation under weak supervision while highlighting the need for strategies that handle object size variation and reduce model bias.

Chapter 1

Introduction

1.1 Semantic Segmentation

In recent years, computer vision has experienced remarkable progress, fueled by the availability of large annotated datasets, advances in computational power, and the development of sophisticated deep learning models. Within this field, semantic segmentation has gained significant importance as a challenging yet essential task with wide-ranging applications in areas such as autonomous driving, medical diagnostics, remote sensing, robotics, and augmented reality. Unlike traditional image classification, which assigns a single category to an entire image, semantic segmentation provides a fine-grained, pixel-level interpretation of visual scenes by labeling each pixel with its corresponding semantic category, as illustrated in Figure 1.1. This capability allows models to not only identify objects but also precisely delineate their shapes and boundaries within an image.

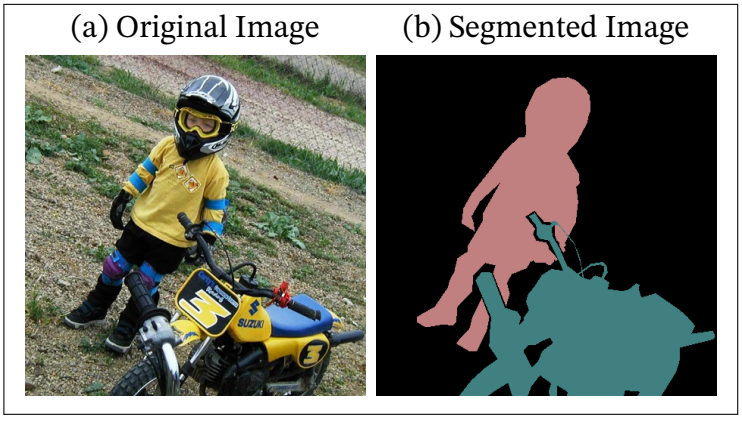


Figure 1.1: Original image and Segmented Image. Two object classes are present in this image: *Person* and *Motorbike*.

The goal of semantic segmentation is to transform raw visual data into structured and meaningful representations that align with human perception. For example, in an autonomous driving scenario, semantic segmentation allows a vehicle to distinguish between roads, pedestrians, vehicles, and traffic signs, ensuring both accurate decision-making and safety. In medical imaging, it aids in the precise delineation of organs, tissues, or pathological regions, thereby assisting in diagnosis and treatment planning. Such applications highlight the importance of fine-grained visual understanding, making semantic segmentation a cornerstone task for achieving scene comprehension in artificial intelligence systems.

Earlier approaches to semantic segmentation primarily depended on manually engineered features combined with classical machine learning algorithms such as random forests, support vector machines (SVMs), and conditional random fields (CRFs). Although these techniques performed reasonably well on certain benchmarks, they struggled to model the complex spatial and contextual dependencies present in natural images. A major turning point occurred with the emergence of deep convolutional neural networks (CNNs), which transformed the field by enabling automatic learning of hierarchical feature representations directly from data. The development of Fully Convolutional Networks (FCNs) was particularly influential, as it replaced traditional fully connected layers with convolutional ones to support dense, pixel-wise prediction. Building on this foundation, subsequent architectures—such as U-Net, SegNet, DeepLab, and PSPNet—introduced innovations like encoder–decoder structures, multi-scale feature aggregation, and attention modules, leading to substantial improvements in segmentation precision and robustness.

Despite these advances, semantic segmentation still faces several challenges. High-quality pixel-level annotations are expensive and time-consuming to obtain, especially for large datasets. Models must also handle issues such as class imbalance, occlusion, scale variation, and boundary precision, which can significantly affect performance. Furthermore, deep models typically require substantial computational resources, making real-time inference a major concern in resource-constrained environments such as mobile or embedded systems. Recent research has thus explored various directions to address these limitations, including weakly supervised, semi-supervised, and unsupervised approaches that reduce the dependency on dense annotations, as well as lightweight architectures optimized for speed and efficiency.

The increasing focus on semantic segmentation has also sparked interest in

integrating it with other vision tasks, such as instance and panoptic segmentation, which aim to provide a more comprehensive understanding of the scene by distinguishing individual object instances. Additionally, combining segmentation with temporal and 3D information, as in video and point cloud segmentation, continues to push the boundaries of scene understanding beyond static 2D imagery.

In summary, semantic segmentation represents a vital step toward achieving full scene understanding in computer vision. By providing detailed, structured, and interpretable representations of visual data, it bridges the gap between low-level pixel information and high-level semantic concepts. This research report explores the principles, methodologies, and recent advancements in semantic segmentation, with particular emphasis on deep learning-based approaches and their applications. Furthermore, it discusses the current challenges and potential future directions that could lead to more efficient, generalizable, and human-like visual perception systems.

1.1.1 Fully Supervised Semantic Segmentation

Fully supervised semantic segmentation is the conventional and most extensively utilized framework for training segmentation networks. In this setting, models are trained using datasets that provide detailed pixel-level annotations, where each pixel in an image is manually assigned to a specific semantic category. Such rich supervision enables the model to effectively learn fine-grained spatial structures and contextual relationships among objects and their boundaries. Benchmark datasets like PASCAL VOC [17], Cityscapes [13], and ADE20K [67] have been instrumental in driving progress within this paradigm.

In a typical fully supervised setting, a deep convolutional neural network (CNN) or transformer-based architecture learns to map an input image to a dense output map where each pixel corresponds to a predicted semantic label. Models like Fully Convolutional Networks (FCNs), U-Net [40], SegNet [6], and DeepLab [9] exemplify the success of this paradigm. These models leverage encoder-decoder designs, skip connections, and multi-scale feature extraction to achieve fine-grained segmentation results. The backbone or the encoder captures global context, while the decoder reconstructs spatial details to produce high-resolution predictions. The pipeline is outlined in Figure 1.2.

However, fully supervised learning comes with a major limitation—its dependence on large-scale, densely annotated datasets. The annotation process is not only

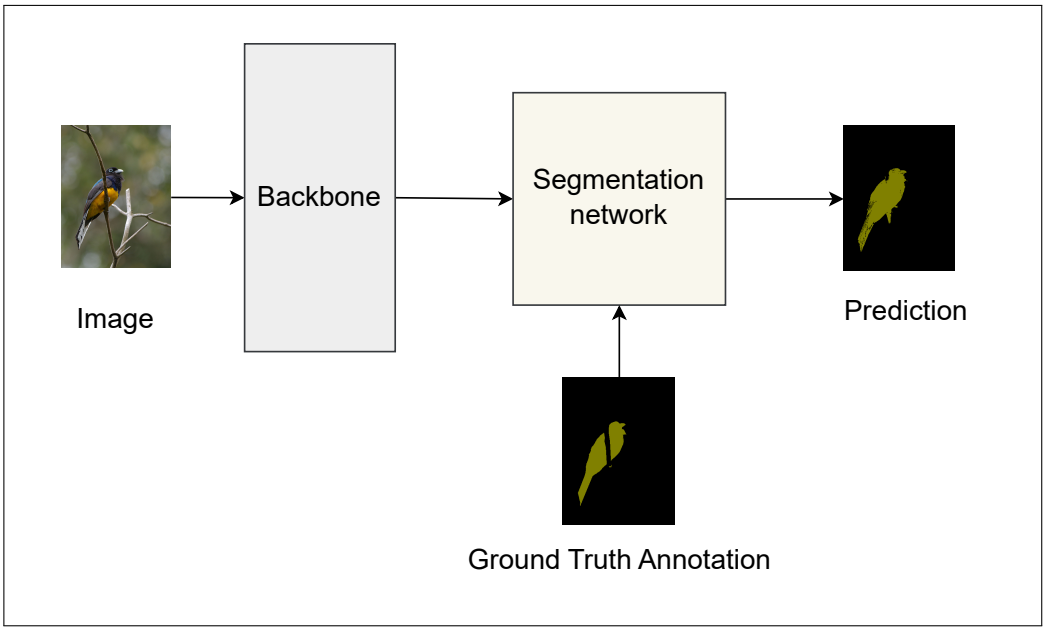


Figure 1.2: Pipeline of the fully supervised semantic segmentation framework, where ground-truth pixel-level annotations are used directly to train the segmentation model.

labor-intensive but also time-consuming and costly, particularly for complex scenes with numerous small or overlapping objects. For instance, annotating a single high-resolution image in the Cityscapes dataset [13] can take several hours. This restricts scalability and makes it difficult to apply such methods to specialized domains like medical imaging or remote sensing, where expert labeling is required. Moreover, fully supervised models may overfit to specific dataset distributions, limiting their generalization to unseen environments. Despite these challenges, fully supervised approaches continue to serve as the benchmark and foundation upon which more flexible and efficient paradigms are built.

1.1.2 Weakly Supervised Semantic Segmentation

To mitigate the annotation burden associated with fully supervised learning, weakly supervised semantic segmentation (WSSS) has emerged as a compelling alternative. Instead of depending on costly pixel-level annotations, weakly supervised approaches train models using simpler forms of supervision, such as image-level tags, bounding boxes, scribbles, or point labels. The primary aim is to maintain competitive segmentation accuracy while greatly reducing the manual labeling effort.

In WSSS, the central challenge lies in bridging the gap between coarse supervision and fine-grained pixel-level prediction. For example, when only image-level labels

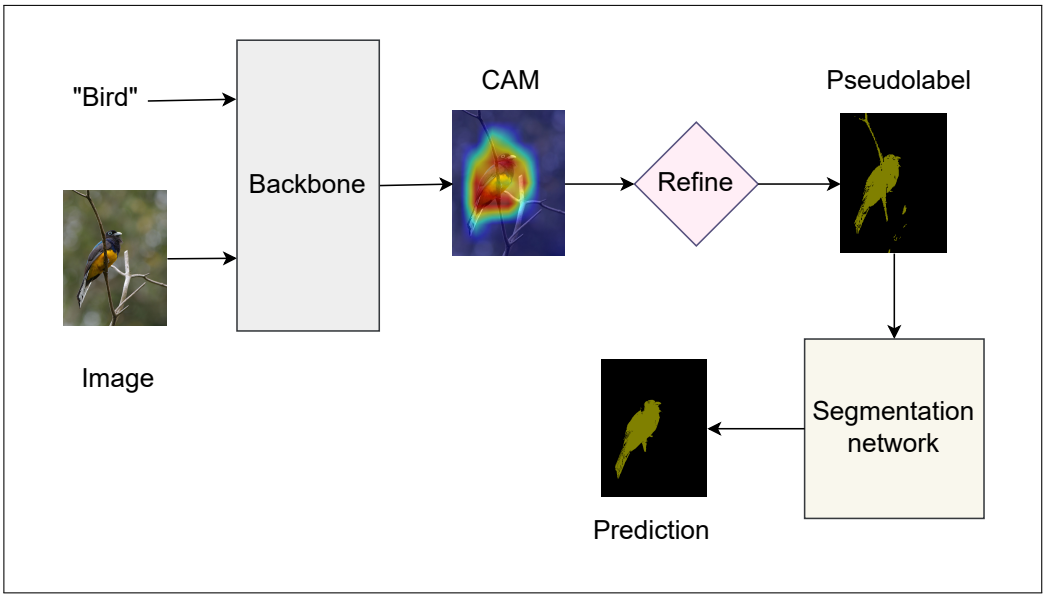


Figure 1.3: Pipeline of the WSSS framework illustrating CAM generation, refinement, and pseudo-label creation for segmentation training. Image-level labels serve as weak supervision to generate Class Activation Maps (CAMs), which are subsequently refined into pseudo-labels for training the segmentation network.

are available (e.g., "cat" or "car"), the model must infer which specific regions of the image correspond to each class. Techniques such as Class Activation Maps (CAMs) and region refinement strategies are commonly employed to identify and expand the most discriminative regions into complete object masks. Additionally, saliency maps, pseudo-label generation, and consistency regularization are often used to improve localization accuracy and enforce smoothness in segmentation outputs. The overall pipeline of this process is outlined in Figure 1.3.

Weakly supervised approaches are particularly valuable in domains where annotated data is scarce or costly to obtain, such as biomedical imaging or satellite imagery. They also facilitate large-scale training by leveraging vast amounts of web images or unlabeled data. However, because supervision is less precise, WSSS models may suffer from incomplete object coverage, ambiguous boundaries, or noise in pseudo-labels. To mitigate these issues, hybrid approaches that combine weak supervision with semi-supervised or self-supervised learning are increasingly being explored.

Overall, weakly supervised semantic segmentation represents an important step toward scalable and data-efficient learning. By reducing the dependence on exhaustive manual labeling while maintaining reasonable accuracy, it brings the field closer to the ultimate goal of achieving human-like visual understanding with minimal supervision.

1.2 Motivation

Semantic segmentation has established itself as a fundamental task in computer vision, with broad applications in areas such as autonomous driving, medical imaging, and robotics. In these domains, a precise pixel-level understanding of the environment is not merely desirable but often critical: for example, safe navigation of self-driving vehicles relies on reliable scene parsing, and accurate delineation of anatomical structures can directly impact medical diagnosis and treatment.

Despite its importance, conventional semantic segmentation has been heavily reliant on fully supervised learning. This paradigm demands large-scale datasets with dense pixel-level annotations, which are costly and labor-intensive to produce. Annotating high-resolution images can take hours per image, and the process remains prone to human error. Beyond annotation effort, fully supervised methods face additional hurdles: (i) class imbalance in real-world datasets often biases models toward dominant categories, (ii) complex scenes with occlusion, lighting variation, and fine structural detail challenge the robustness of predictions, and (iii) models trained on specific datasets frequently fail to generalize across domains due to dataset-specific biases.

These challenges naturally motivate the exploration of weakly supervised semantic segmentation (WSSS). By replacing dense annotations with weaker forms of supervision—such as image-level labels, bounding boxes, or scribbles—WSSS reduces annotation cost while still enabling model training. The key idea is to leverage weak supervision to generate class activation maps (CAMs) and corresponding pseudo-labels, which can then guide the segmentation process. Although CAMs are often coarse or noisy, refinement strategies allow them to approximate dense ground truth, making WSSS a practical compromise between annotation effort and segmentation performance.

The motivation for this research is thus twofold: first, to address the scalability and practicality issues of fully supervised methods, and second, to improve the quality and reliability of WSSS pipelines. By advancing WSSS, we aim to narrow the performance gap with fully supervised segmentation while ensuring that solutions remain feasible for deployment in diverse and data-constrained real-world scenarios.

1.3 Problem Statement

Weakly Supervised Semantic Segmentation (WSSS) presents a compelling alternative to fully supervised methods by significantly reducing the dependence on costly pixel-level annotations. However, current WSSS techniques rely on Class Activation Maps (CAMs) [66] generated from image-level labels to identify object regions. While this approach has shown promise, it often falls short in terms of spatial precision and completeness, leading to suboptimal segmentation results.

In the context of WSSS, the challenge lies in effectively leveraging the image-level labels to produce accurate and detailed segmentation maps. The dependence on CAMs, which are typically generated from global features, can result in sparse and coarse activation maps that fail to capture the fine details of object boundaries. This limitation is particularly pronounced when dealing with complex scenes or occlusions, where precise localization is crucial.

Additionally, the refinement techniques applied to these CAMs often fail to fully leverage the rich affinity information inherent in modern transformer architectures. Consequently, the resulting pseudo-labels lack the spatial precision required for high-quality segmentation, ultimately impacting the overall performance. This highlights the need for more robust backbone architectures capable of effectively capturing both local and global features, as well as advanced CAM refinement strategies to narrow the performance gap between weakly and fully supervised segmentation methods.

So, keeping in mind the above challenges, we aim to develop a weakly supervised semantic segmentation model that can effectively leverage image-level labels to produce accurate and detailed segmentation maps. Our approach will focus on enhancing the spatial precision and completeness of the generated CAMs.

1.4 Challenges of WSSS

While weakly supervised semantic segmentation (WSSS) reduces the annotation burden compared to fully supervised methods, it introduces its own set of challenges:

- **Incomplete Object Localization:** Class activation maps (CAMs) derived from image-level labels typically highlight only the most discriminative regions, leaving large portions of the object unlabeled.

- **Noisy Pseudo-labels:** The process of refining CAMs into pixel-level labels often introduces noise and errors, which can propagate during training and degrade performance.
- **Boundary Precision:** Weak supervision lacks explicit boundary cues, making it difficult to segment fine object details and separate adjacent instances accurately.
- **Background Confusion:** Without strong pixel-level supervision, models often misclassify diverse background regions as foreground, or vice versa.
- **Multi-class Co-occurrence:** In scenes containing multiple objects, weak labels struggle to capture clear distinctions, leading to overlapping or missing class activations.
- **Class Imbalance:** Underrepresented classes may not produce strong activations in CAMs, resulting in poor segmentation for rare categories.
- **Dependence on External Cues:** Many WSSS methods rely on saliency maps or additional priors to improve CAMs, but these external cues may be dataset-specific and limit generalization.

Addressing these challenges is crucial for advancing WSSS methods toward practical applications where annotation resources are limited.

1.5 Contribution

In this work, we make the following key contributions:

- We explore the use of the UniCL framework [58] with a Swin Transformer backbone [32] to enhance CAM generation, leveraging Swin's ability to capture both local fine details and global context through its hierarchical structure.
- We adapt the affinity-based CAM refinement technique from WeCLIP [61] to the Swin Transformer backbone by computing pixel affinities from Swin's hierarchical features, allowing the refinement to propagate class activations effectively despite the absence of a global attention map.
- We integrate a Pixel-Adaptive Refinement Module (PAR) [42] that incorporates both color and spatial information, further refining the pseudo-labels and enhancing boundary accuracy.
- We integrate the refined CAMs and Pixel-Adaptive Refinement into the

standard WSSS training pipeline, training the model to leverage these improved pseudo-labels for better segmentation performance.

1.6 Organization

The remainder of this thesis report is organized as follows:

- **Chapter 2: Related Works** provides a comprehensive review of existing literature on semantic segmentation. It discusses fully supervised and weakly supervised methods, key architectures like U-Net, DeepLab, and Vision Transformers, and identifies gaps in current research.
- **Chapter 3: Proposed Methodology** details the proposed approach, including the use of the UniCL framework with a Swin Transformer backbone for CAM generation. It also describes the affinity-based CAM refinement technique and the Pixel-Adaptive Refinement Module (PAR) for pseudo-label generation and segmentation refinement.
- **Chapter 4: Results and Discussion** presents the experimental setup, quantitative and qualitative analysis, including per-class performance, visualizations of CAMs and segmentation masks, ablation study, interpretation of results, discussion of limitations and challenges, and key findings. Additionally, it covers other explored approaches and insights gained from them.
- **Chapter 5: Conclusion** summarizes the study's contributions and major insights, emphasizing the broader implications of the findings. It also outlines potential future research directions.

This structure ensures a logical flow, starting from the foundational concepts and related works, progressing through the proposed methodology, and concluding with the results, discussions, and supplementary materials.

Chapter 2

Related Works

2.1 Fully Supervised Semantic Segmentation

In fully supervised semantic segmentation, the model is trained on a dataset with pixel-level annotations. The model learns to predict the class of each pixel in an image based on the provided annotations. This approach typically requires a large amount of labeled data, which can be expensive and time-consuming to obtain.

2.1.1 Early Approaches

The early approaches to semantic segmentation relied heavily on hand-crafted features and traditional machine learning techniques. Researchers designed algorithms that extracted low-level image features such as color histograms, texture descriptors (e.g., Local Binary Patterns, Gabor filters), and shape or edge-based descriptors. These features were then fed into classifiers like Support Vector Machines (SVMs), Random Forests, or Conditional Random Fields (CRFs) to assign labels to each pixel or superpixel.

One prominent approach involved the use of superpixels, where an image is first divided into small, perceptually meaningful regions, and then features were aggregated within these regions for classification. This helped reduce computational complexity and provided some spatial regularization. Other methods incorporated graph-based models where pixels or superpixels were treated as nodes, and edges encoded spatial or appearance-based relationships, allowing the use of graph cuts or CRFs to enforce label consistency. Despite their innovation, these methods faced significant limitations. The hand-crafted features were often not robust to variations in lighting, viewpoint, or scale, and the segmentation quality heavily depended on

the design of features and hyperparameters. Additionally, traditional classifiers could not effectively capture high-level semantic relationships, limiting their ability to distinguish complex objects or handle cluttered scenes.

The advent of deep learning marked a turning point in semantic segmentation. With the introduction of Convolutional Neural Networks (CNNs), models became capable of learning hierarchical representations of images directly from raw data. Fully Convolutional Networks (FCNs) were particularly influential, allowing end-to-end training for dense pixel-wise prediction. This eliminated the need for manual feature extraction and enabled models to capture both low-level textures and high-level semantic information simultaneously. Over time, more advanced architectures such as U-Net, SegNet, and DeepLab improved segmentation accuracy by incorporating encoder-decoder structures, skip connections, and multi-scale context aggregation.

These developments laid the foundation for modern segmentation pipelines and highlighted the shift from manual feature engineering to data-driven representation learning, paving the way for both fully supervised and weakly supervised approaches in subsequent research.

2.1.2 Convolutional Neural Networks (CNNs)

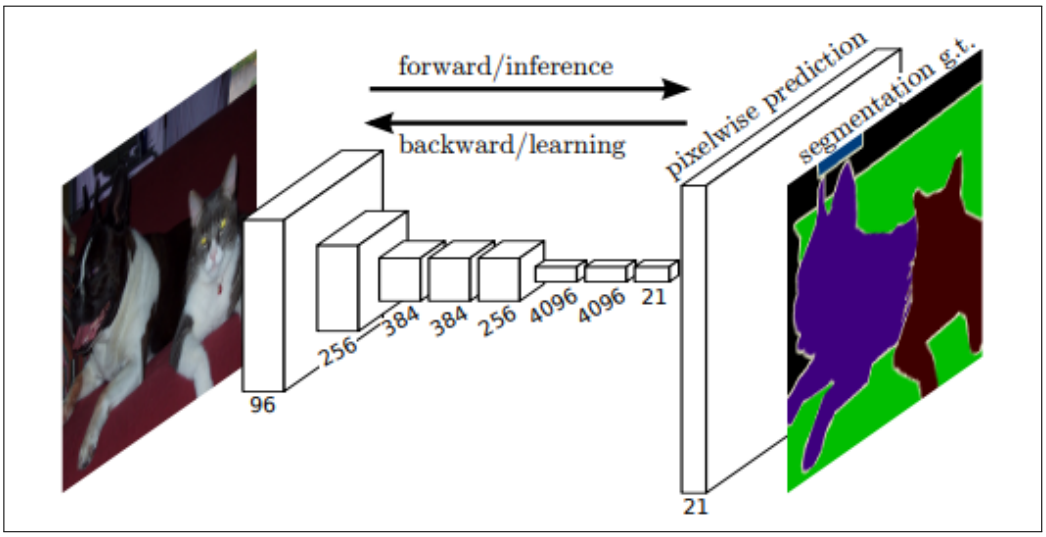


Figure 2.1: Fully Convolutional Network (FCN) for semantic segmentation (adapted from [33]).

The first fully convolutional network (FCN) for semantic segmentation was proposed by Long et al. [33], which replaced the fully connected layers in traditional CNNs with convolutional layers, as shown in Figure 2.1. This allowed the network to produce dense predictions for each pixel in the input image. The FCN architecture was further

improved by adding skip connections, which helped to preserve spatial information and improve segmentation accuracy.

Then came the introduction of U-Net [40], a popular architecture for semantic segmentation that is widely used in medical imaging and other applications. U-Net consists of an encoder-decoder structure, where the encoder captures context information, and the decoder enables precise localization. The *skip connections* between the encoder and decoder blocks play a crucial role by retaining spatial information, making U-Net particularly effective for tasks with limited training data.

The field of semantic segmentation has continued to evolve with the introduction of various architectures and techniques.

The DeepLab series [8–11] introduced atrous convolution and spatial pyramid pooling to capture multiscale context information; DeepLabv1 [9] also used a method called CRF (Conditional Random Field) to refine the segmentation results. But it was removed in later versions of the model. In DeepLabv2 [8], the atrous convolution method evolved into a more general form called atrous spatial pyramid pooling (ASPP), which allows the model to capture features at multiple scales. The ASPP module consists of parallel atrous convolutions with different rates, enabling the model to learn multiscale context information effectively.

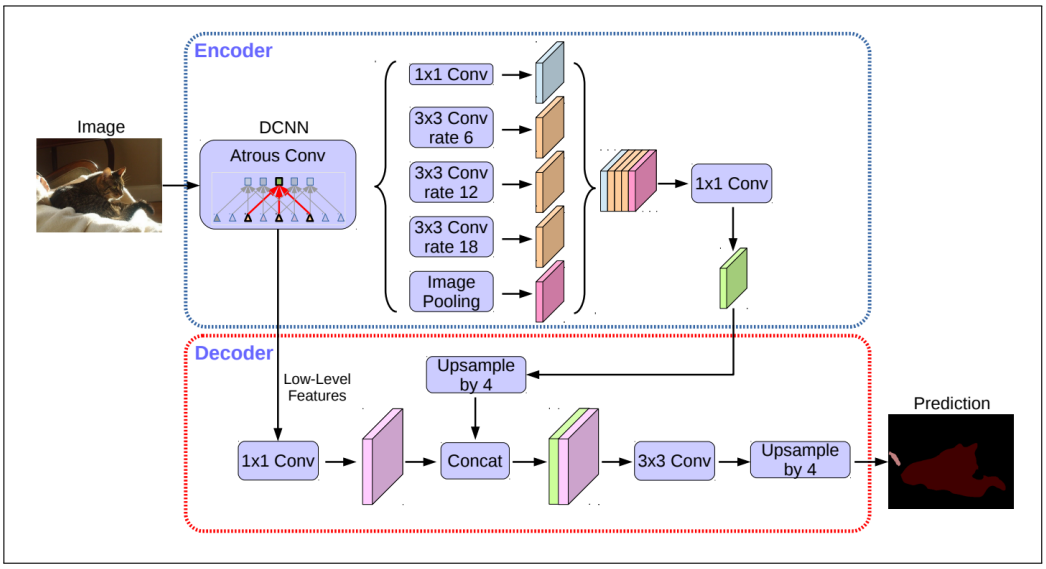


Figure 2.2: Architecture of DeepLabv3+ (adapted from [11]).

Finally, DeepLabv3+ [11] introduced a new decoder module that refines the segmentation results by combining low-level features from the encoder with high-level features from the ASPP module, as shown in Figure 2.2. This approach improves the localization of object boundaries and enhances the overall segmentation performance.

Zhao et al. [64] introduced the Pyramid Scene Parsing Network (PSPNet), which uses a pyramid pooling module to capture global context information at different scales. The pyramid pooling module aggregates features from different regions of the image, allowing the model to learn rich contextual information for better segmentation.

SegNet [6] is an encoder-decoder architecture that focuses on efficient upsampling of feature maps.

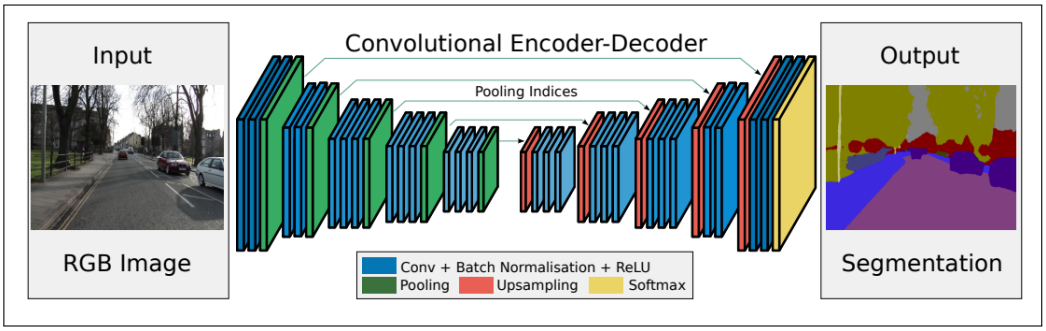


Figure 2.3: An illustration of the SegNet architecture (adapted from [6]).

SegNet uses a series of convolutional and pooling layers in the encoder to extract features, followed by a corresponding decoder that upsamples the feature maps to produce the final segmentation output, as demonstrated in Figure 2.3. The main innovation in SegNet is the use of unpooling layers, which store the indices of the max-pooling operation during encoding, instead of the feature maps themselves. This allows for more efficient memory usage and faster inference times, making SegNet suitable for real-time applications.

2.1.3 Transformers

The introduction of transformers in computer vision has led to significant advancements in semantic segmentation. Vision Transformer (ViT) [15] and its variants have shown promising results in various tasks, including semantic segmentation. ViTs use self-attention mechanisms to capture long-range dependencies and global context information, making them suitable for dense prediction tasks. Later on, transformer-based architectures have gained popularity in semantic segmentation.

Swin Transformer [32] introduces a hierarchical architecture with shifted windows to capture both local and global context information. Swin works like a transformer but its hierarchical feature extraction is analogous to that of CNNs. It has shown state-of-the-art performance on various benchmark datasets, including ImageNet-1k [14] for image classification, COCO [30] for object detection, and

ADE20K [67] for semantic segmentation. The Swin Transformer has thus been widely adopted in various applications.

Leveraging the strengths of Vision Transformer, Zheng et al. [65] first proposed the SETR (Semantic Segmentation Transformer) architecture, which replaces the traditional CNN backbone with a transformer encoder. The SETR architecture consists of a ViT encoder that processes the input image and generates a set of feature maps, followed by a decoder that upsamples the feature maps to produce the final segmentation output. The SETR model has shown competitive performance on various benchmark datasets, demonstrating the effectiveness of transformers in semantic segmentation tasks.

The Segmenter [46] architecture combines a ViT backbone with a lightweight decoder for efficient semantic segmentation.

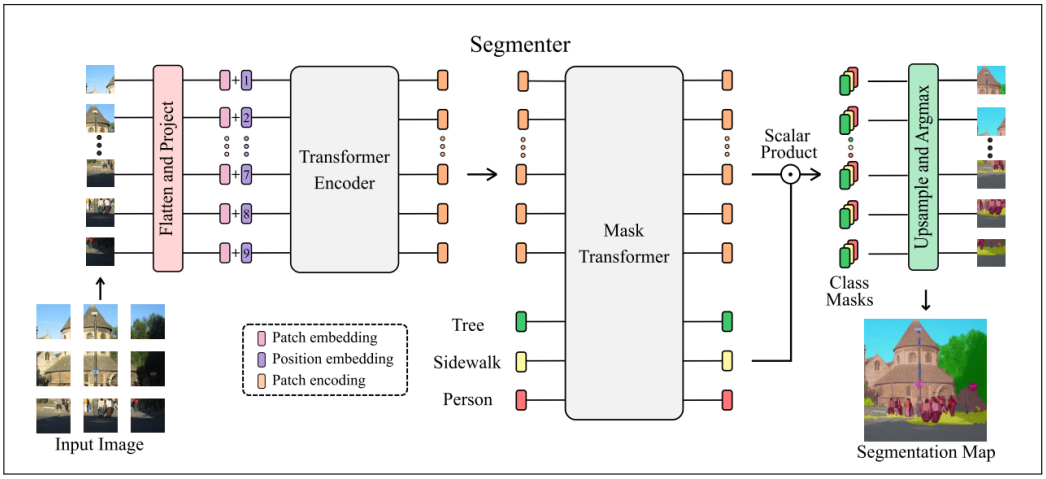


Figure 2.4: The Segmenter framework (adapted from [46]).

The Segmenter model, as shown in Figure 2.4 uses a transformer encoder to capture global context information and a simple decoder to produce the final segmentation output. The Segmenter architecture is designed to be computationally efficient while maintaining high accuracy, making it suitable for real-time applications.

SegFormer [54] is a transformer-based architecture that combines the strengths of both CNNs and transformers for semantic segmentation. SegFormer, as illustrated in Figure 2.5 uses a hierarchical transformer encoder to capture multi-scale features and a lightweight decoder to produce the final segmentation output. The SegFormer model has shown state-of-the-art performance on various benchmark datasets, demonstrating the effectiveness of combining CNNs and transformers in semantic segmentation tasks.

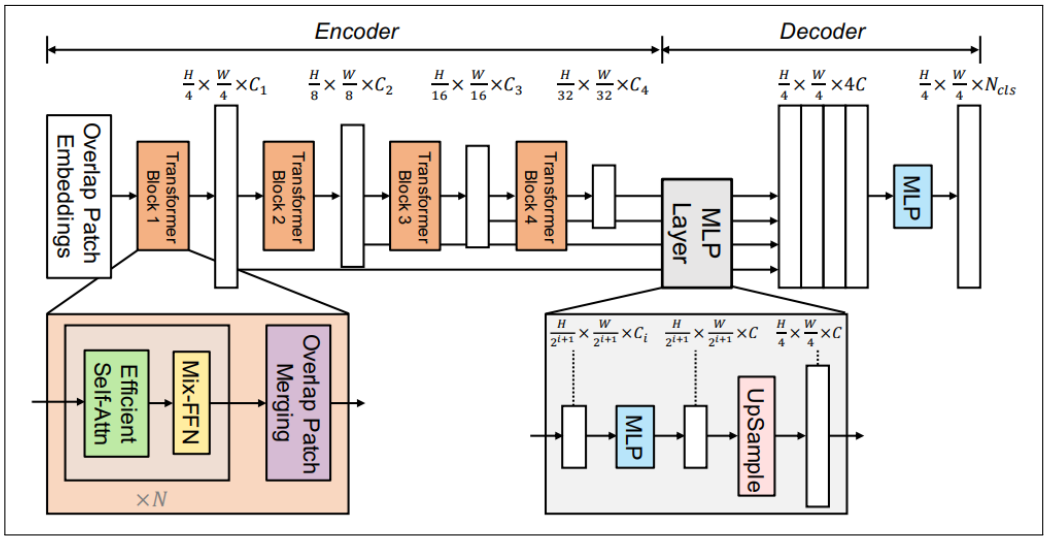


Figure 2.5: An illustration of the SegNet architecture (adapted from [54]).

The field of semantic segmentation has seen significant advancements with the introduction of various architectures and techniques. The combination of CNNs and transformers has led to improved performance and efficiency in semantic segmentation tasks. As the field continues to evolve, we can expect further innovations and breakthroughs in this area.

2.1.4 Problem with Fully Supervised Semantic Segmentation

Fully supervised semantic segmentation has achieved remarkable success, largely driven by powerful models and richly annotated datasets. However, this success comes at a significant cost: acquiring dense pixel-level annotations is both expensive and time-consuming. Each image must be meticulously labeled, often requiring expert knowledge and hours of manual effort. This high annotation burden severely limits the scalability of fully supervised methods, especially for large or specialized datasets. To overcome this bottleneck, the research community has increasingly turned toward weakly supervised semantic segmentation (WSSS), an approach that seeks to train effective segmentation models with minimal supervision.

2.2 Weakly Supervised Semantic Segmentation

Fully supervised semantic segmentation depends on pixel-level segmentation masks annotated by humans. However, generating such dense annotations is tedious, time consuming, and costly. Furthermore, crowd-sourced annotators must undergo special training to handle the complexity of pixel-level labeling, restricting the scale

and diversity of available datasets. Consequently, most curated datasets are limited to a small set of object categories. In contrast, unlabeled or weakly annotated images can be collected in abundance, at much lower cost, and in shorter time. This has motivated research into weakly supervised semantic segmentation (WSSS) to make semantic segmentation models more scalable.

2.2.1 Types of WSSS

A wide range of weak supervision has been explored, including bounding boxes, scribbles, points, image-level labels, eye tracks, free-form squiggles, or noisy web tags. Bounding boxes provide rough object boundaries, offering useful localization cues, though they still require annotators to draw accurate boxes. Scribble-based supervision allows annotators to roughly mark object regions without outlining exact object boundaries. Point supervision, by contrast, typically uses a single annotated pixel per object, giving coarse location information. While less costly than pixel-accurate masks, these methods still involve some level of manual annotation, making large-scale labeling expensive.

2.2.2 Image-Level Label Based Weak Supervision

Image-level annotation represents a cost-effective and efficient form of supervision for weakly supervised semantic segmentation. Here, each training image is provided only with class labels indicating which object categories are present, but without any information about their spatial locations. The main challenge, therefore, is to correctly associate these image-level labels with the appropriate pixels in the image.

The initial approaches attempted to train segmentation models directly from image-level labels [35], but the performance was unsatisfactory. Later methods introduced discriminative localization techniques such as Class Activation Maps (CAMs) [66], which highlight class-relevant regions. These coarse cues were then refined using auxiliary information such as superpixels [37], segmentation proposals [37], or motion information from videos [49]. Some works, such as Adversarial Erasing [51], expanded object coverage progressively by iteratively searching new regions. Others, like Kolesnikov and Lampert [23], trained networks to approximate the dense CRF [25] applied on CAMs for refinement.

Some approaches learn to predict affinity matrices at the pixel level [36], to refine the output of dCRF through random walk. AffinityNet [4] predicts class-agnostic pixel affinities to propagate CAM activations via random walks. Similarly, Seeded Region

Growing (SRG) [2] expands initial seed regions using similarity criteria, while its deep learning extension DSRG [20] leverages high-level semantic features to grow regions more effectively. P2P [16] further narrows the supervision gap by using pixel-to-prototype contrastive learning. However, a common limitation is that most of these CNN-based methods inherit the restricted locality of convolutional features.

Recent advances incorporate transformers into WSSS [18, 48]. TS-CAM [18] leverages the global information capturing ability of ViT by combining class-token attention with CAMs. [48] refines class-specific maps by using attention gradients. MCTformer [56] expands this idea by embedding multiple class tokens to learn attention maps for different categories. AFA [42] instead derives semantic affinities directly from attention maps to refine coarse pseudo-labels. WeCLIP [61] pushes this further by exploiting the frozen CLIP backbone to generate high-quality pseudo-labels, which are dynamically updated using a refinement module (RFM).

2.2.3 Class Activation Maps

Class Activation Maps (CAMs) serve as a crucial interpretability mechanism in deep learning, enabling the visualization of spatial regions within an image that most strongly influence a model's classification decisions. By highlighting class-discriminative areas, CAMs facilitate a deeper understanding of the model's internal reasoning and are foundational for generating localization cues in weakly supervised semantic segmentation. This section provides a comprehensive overview of CAM generation methodologies, with particular emphasis on their implementation in advanced frameworks such as UniCL and architectures like the Swin Transformer.

Basic Classification

In a standard image classification task, the model learns to assign input images to specific categories. To generate CAMs, global average pooling is applied to the feature maps from the final convolutional layer, condensing spatial information into a single vector. This process produces a heatmap that emphasizes the image regions most influential for the model's classification decision.

In the Figure 2.6, we can see a high-level overview of the basic classification task. The process can be summarized as follows:

1. The input image is passed through the model, and the feature maps are generated. This deep feature extraction block can be either CNNs (as it

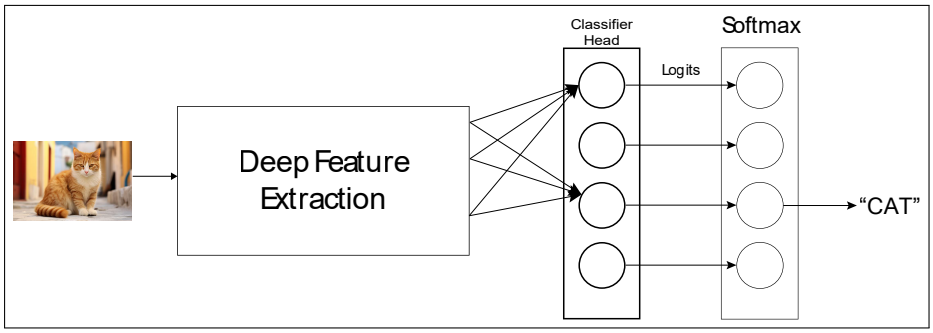


Figure 2.6: Basic Classification

initially was) or Transformers.

2. The *MLP (Multi-Layer Perceptron)*, also called the *Classifier head*, is used to produce the final classification scores.
3. The scores are then normalized using a *softmax* function to obtain the class probabilities.

No matter how the feature maps are generated or what the backbone is, the MLP head is used to produce the final classification scores. For CAM generation, we only need the feature maps and the class scores, i.e, upto the Classifier head.

Convolutional Neural Networks (CNNs) are often considered "black-box" models due to the limited interpretability of their internal decision-making process. To address this, Class Activation Mapping (CAM) was introduced by Zhou et al. [66], demonstrating that CNNs can act as unsupervised object detectors by highlighting discriminative image regions relevant to classification. CAM and its variants have since become central to Weakly Supervised Semantic Segmentation (WSSS), where such localization cues are refined into pixel-level pseudo-labels for segmentation training.

Vanilla CAM

The original CAM [66] requires a specific network architecture where fully connected layers are replaced by a Global Average Pooling (GAP) layer followed by a linear classifier. The process is illustrated in Figure 2.7:

1. The input image is passed through a CNN to obtain convolutional feature maps.
2. GAP is applied to produce a compact vector representation of the image.
3. This representation is fed into the classifier head to generate class scores.
4. The neuron corresponding to the predicted class is identified.

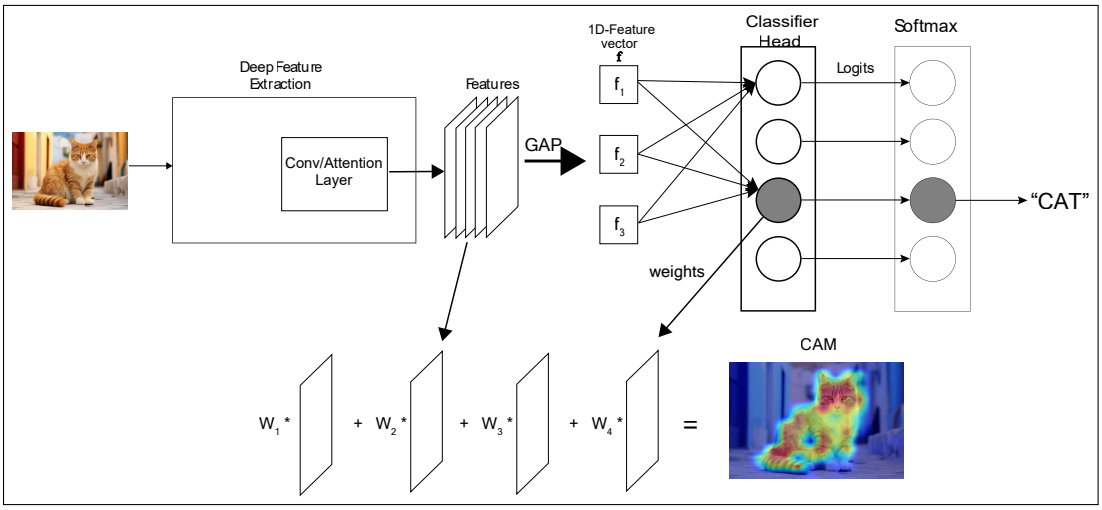


Figure 2.7: CAM Generation Process

5. The learned weights associated with this class are used to compute a weighted sum of the feature maps, yielding a class-specific heatmap.
6. The heatmap is upsampled to match the original image resolution for visualization.

Although effective, CAM requires the network to be modified by inserting GAP, which limits its general applicability.

Grad-CAM

To overcome CAM's architectural constraint, Selvaraju et al. [44] proposed Gradient-weighted CAM (Grad-CAM). Unlike CAM, Grad-CAM can be applied to any CNN-based model without requiring structural changes. Instead of relying solely on classifier weights, Grad-CAM leverages the gradient of the target class score with respect to feature maps. The process is illustrated in Figure 2.8:

1. Forward pass the input to obtain feature maps from the last convolutional layer.
2. Compute the classification scores through the classifier head.
3. Select the target class neuron (often the top prediction, but any class can be chosen).
4. Backpropagate to compute gradients of this class score with respect to the feature maps.
5. Average the gradients spatially to obtain weights for each feature map channel.
6. Combine the feature maps using these weights to create a coarse localization

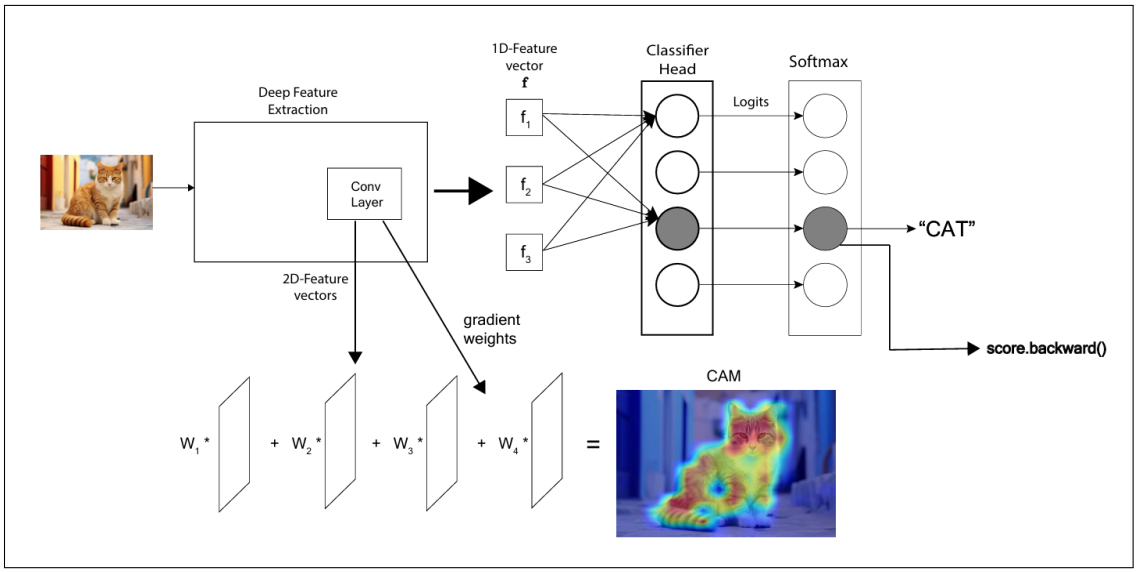


Figure 2.8: Grad-CAM Generation Process

heatmap.

7. Apply ReLU to suppress negative contributions.
8. Upsample the heatmap to align with the input image.

Grad-CAM preserves the architectural flexibility missing in CAM, while still providing class-discriminative localization.

Grad-CAM++

Grad-CAM++ [7] was proposed to address two main limitations of Grad-CAM: (1) poor handling of multiple object instances of the same class, and (2) sensitivity to object size and distribution. Instead of relying only on first-order gradients, Grad-CAM++ incorporates higher-order derivatives of the class score with respect to feature maps.

1. The forward pass generates feature maps as in Grad-CAM.
2. For the target class, first-order and higher-order gradients are computed.
3. These gradients are used to calculate the pixel-wise weights, where each spatial location contributes differently to the final importance score.
4. The weighted sum of the feature maps is constructed using these refined weights, resulting in a sharper and more spatially precise heatmap.
5. Like Grad-CAM, the heatmap is processed with ReLU and upsampled.

This method improves localization in complex scenarios, such as small objects or multiple instances, while requiring only one backward pass, keeping computational costs similar to Grad-CAM.

LayerCAM

LayerCAM[22] builds on the intuition that meaningful localization is not limited to the last convolutional layer. Earlier and intermediate layers often capture fine-grained features such as edges and textures, which can complement coarse semantic features from deeper layers.

LayerCAM[22] computes CAMs at multiple levels of the network:

1. For each chosen convolutional layer, the gradients with respect to the target class are calculated.
2. Instead of spatially averaging the gradients, LayerCAM[22] assigns weights at every spatial location by multiplying activation values with their corresponding gradients.
3. These location-aware weights allow the method to produce more precise heatmaps per layer.
4. CAMs from different layers are then combined to form a multiscale localization map that captures both coarse object regions and fine details.

By integrating multilayer signals, LayerCAM[22] enhances both interpretability and localization accuracy compared to Grad-CAM.

Attention Rollout and Attention Flow

Beyond CNN-based visualization, transformer-based networks introduced attention-based interpretability. Abnar and Zuidema [1] proposed two techniques:

- **Attention Rollout**: Recursively multiplies the attention matrices across layers, incorporating residual connections. This assumes that the token identity propagates linearly through attention weights, providing a global view of how input tokens influence the final decision.
- **Attention Flow**: Formulates interpretability as a maximum flow problem, modeling information propagation as a flow network. A maximum flow algorithm is then applied to trace how embeddings in deeper layers are connected back to input tokens.

Both methods generate token-level importance maps that correlate strongly with gradient-based measures, improving interpretability for transformer architectures.

RelevanceCAM

RelevanceCAM [26] incorporates Layer-wise Relevance Propagation (LRP) into CAM generation. Unlike gradient-based methods that can suffer from the shattered gradient problem, LRP redistributes the prediction score backward through the network based on relevance conservation principles.

1. Starting from the target class score, relevance values are propagated layer by layer toward the input.
2. Each neuron's contribution is quantified based on how much it supports the class prediction.
3. These relevance scores are aggregated spatially to form heatmaps in multiple layers.
4. The final activation map combines information from both shallow and deep layers, enhancing robustness and interpretability.

RelevanceCAM [26] provides more stable and faithful explanations, particularly in intermediate layers, and demonstrates improved localization performance over traditional Grad-CAM variants.

Summary

Overall, CAM-based methods have evolved from the original GAP-based CAM [66] to gradient-driven approaches like Grad-CAM [44], higher-order generalizations such as Grad-CAM++ [7], multi-layer extensions like LayerCAM [22], attention-based methods for transformers [1], and LRP-driven methods like RelevanceCAM [26]. These advances have been pivotal in WSSS, where class-discriminative heatmaps serve as initial localization cues that are refined into pixel-level pseudo-labels for segmentation.

2.3 Component-Wise Literature Review of WSSS Models

2.3.1 Feature Extraction From Backbone

The architecture of AffinityNet [4] relies on three DNNs: a classification network for generating CAMs, the AffinityNet module itself, and a segmentation network.

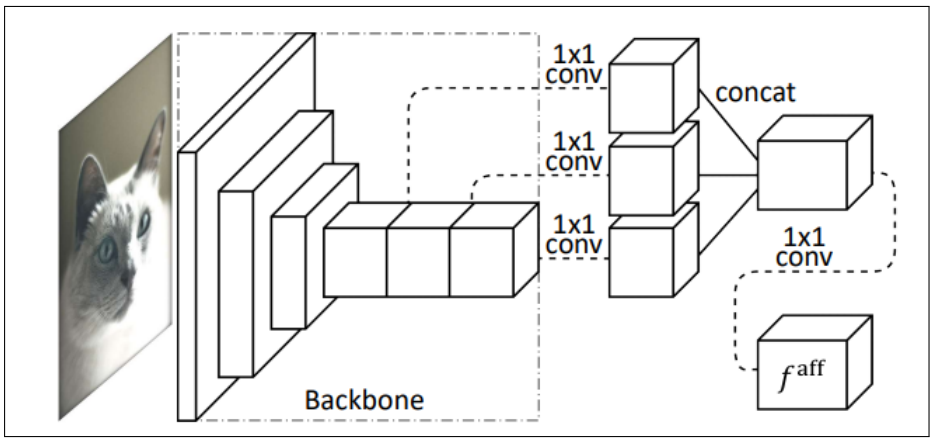


Figure 2.9: Overall architecture of AffinityNet (adapted from [4]).

All three share the same backbone, as illustrated in Figure 2.9. The backbone is a modified variant of Model A1 [53], commonly referred to as ResNet38, which consists of 38 convolutional layers with wider channels. In this adaptation, the GAP and fully connected layers from the original design are removed, and the final three convolutional stages are replaced with atrous convolutions with stride 1. The dilation rates are adjusted so that the resulting feature maps maintain a stride of 8.

In DSRG [20], the classification branch employs a slightly altered version of the 16-layer VGG model [24], while the segmentation branch is built upon DeepLabv2 [8]. Both are initialized with VGG-16 weights pre-trained on ImageNet.

AFA [42] adopts the Mix Transformer (MiT) backbone introduced in SegFormer [54], which is more suitable for image segmentation tasks compared to the vanilla ViT. The MiT parameters are initialized using ImageNet-1k pre-trained weights.

ToCo [43] uses the ViT-Base (ViT-B) backbone initialized with ImageNet pre-trained weights. Within the ViT encoder, an auxiliary classification head is introduced to produce CAMs. These auxiliary CAMs are then used to form pseudo labels guiding the Pixel-Token Contrast (PTC) module, and also to generate proposals for cropping positive and negative local patches for the Cross-Token Contrast (CTC) module. The

final CAM is generated through the classification layer and subsequently transformed into pseudo labels.

CLIP-ES [31] employs the CLIP ViT-B/16 backbone, where the image encoder extracts visual features and the text encoder extracts linguistic features. Since CLIP is pre-trained on roughly 400 million image-text pairs, it provides strong multimodal representations for segmentation.

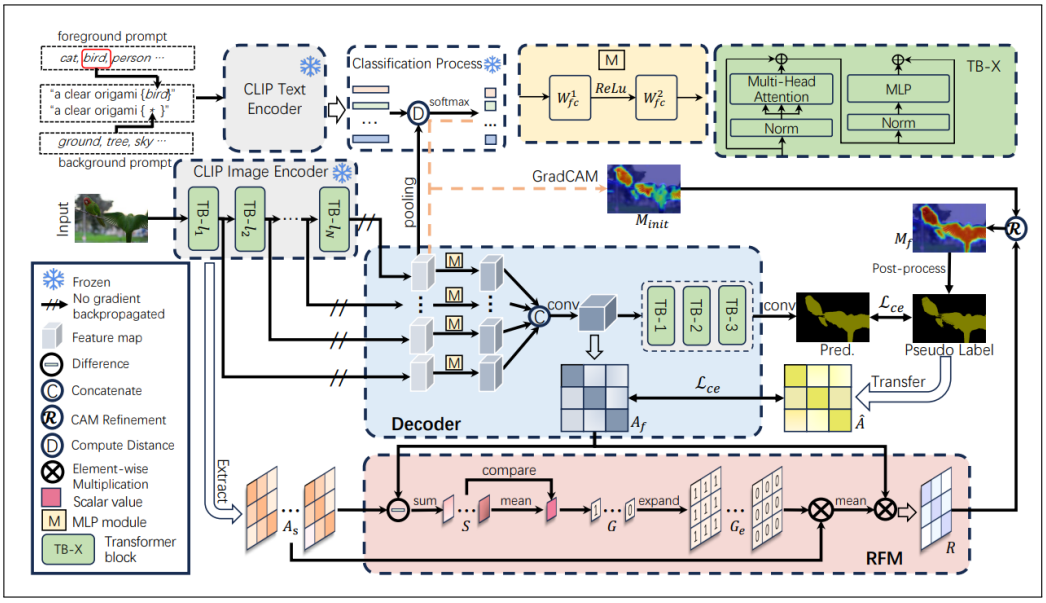


Figure 2.10: Overall architecture of WeCLIP (adapted from [61]).

The whole framework of WeCLIP[61] consists of four major components: a frozen CLIP backbone (comprising a ViT-Base/16 image encoder and a text encoder) [15], a classification module for generating initial CAMs, a decoder for segmentation prediction, and a refinement module (RFM) to enhance CAMs into pseudo labels for supervision, as shown in Figure 2.10.

2.3.2 CAM Generation

In AffinityNet [4], CAMs are generated by appending three layers to the backbone: a 3×3 convolutional layer with 512 channels for task adaptation, a global average pooling layer to aggregate spatial information, and a fully connected layer for classification. The class activation maps are computed by weighting the feature maps with the class-specific weights from the final layer. These maps are normalized such that their maximum value is 1, and the background map is obtained by subtracting the maximum class activation at each pixel from 1.

In DSRG [20], CAMs [66] are used to localize foreground regions. The classification branch applies a fully connected layer to the conv7 features, producing a heatmap

for each class. Foreground seed cues are extracted by thresholding these heatmaps, while background cues are identified from regions in normalized saliency maps with low pixel intensities.

AFA [42] similarly employs CAMs to generate initial pseudo labels. These maps are formed by linearly combining feature maps from the classification layer using learned weights, followed by a ReLU activation to suppress negative values. The outputs are then scaled to the [0,1] range, and an additional background score is included to differentiate foreground from the background.

ToCo [43] incorporates an auxiliary classification layer to extract semantic information and produce CAMs. Patch tokens are aggregated using global max pooling and subsequently passed through a fully connected layer, resulting in auxiliary CAMs that guide the model's training.

In WeCLIP[61], image features are extracted from the frozen CLIP image encoder, while class-specific text prompts are processed by the CLIP text encoder, both remaining fixed during training. Classification scores are calculated as distances between pooled image and text features, and Grad-CAM [44] is applied to these scores to generate the initial CAMs.

2.3.3 Pseudo Label Generation

AffinityNet[4] predicts convolutional feature maps in which the semantic affinity between two feature vectors is measured using their L1 distance. To train this network, semantic affinity labels are derived from CAMs. Specifically, confident foreground and background regions are identified, while ambiguous regions are treated as neutral. Pairwise affinities are then defined according to class labels — assigned as 1 for pixels of the same class, 0 for different classes, and ignored for neutral labels. Once trained, AffinityNet refines CAMs through a random walk guided by the semantic transition matrix, thereby enhancing CAM quality and producing more accurate pseudo segmentation labels.

In DSRG[20], pseudo labels are generated by expanding initial seed cues into unlabeled regions using the classical Seeded Region Growing (SRG) algorithm [2]. For each class, the seed cues are visited in row first manner and its 8-connectivity neighborhood pixels are checked for similarity criteria. If they satisfy the criteria , those pixels are added to the set of seed cues for that particular class, and this process is repeated for every object class. The dynamically updated seed sets act as supervision, continuously refining pseudo labels during training.

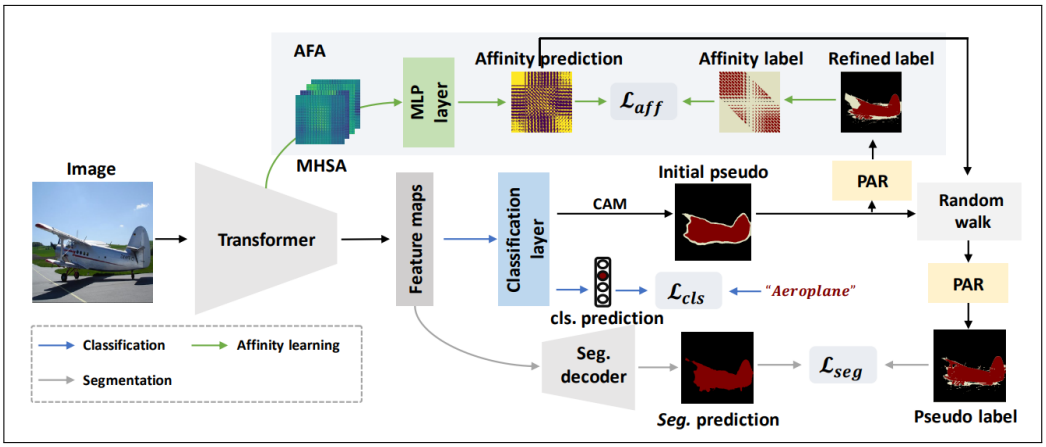


Figure 2.11: AFA framework for WSSS (adapted from [42]).

The AFA[42] module computes semantic affinities directly by linearly combining multi-head attention outputs through an MLP layer. To ensure matrix symmetry, the affinity matrix is summed with its transpose, under the assumption that nodes with the same semantics should be equivalent.

Supervision is provided by pseudo affinity labels derived from refined CAMs: pixels are assigned to the class with the highest activation, and affinities are set positive if two pixels share the same class; otherwise negative. These pseudo affinity labels supervise the affinity prediction process, as shown in Figure 2.11. The learned semantic affinities are then used in a random walk propagation to refine the CAMs.

In ToCo[43], auxiliary CAMs are segmented into pseudo token labels by applying two background thresholds, categorizing tokens into reliable foreground, background, and uncertain regions. Positive token pairs are defined as those sharing the same label, while others are treated as negative. To counteract over-smoothing, the Patch Token Contrast (PTC) module maximizes similarity between positive pairs of patch tokens and minimizes it for negative ones. Additionally, as shown in Figure 2.12, the Class Token Contrast (CTC) module enforces consistency across entire object regions by aligning representations of global and local class tokens. Specifically, it reduces the gap between projected global class tokens and projected local tokens cropped from uncertain or background regions, thereby improving pseudo label quality.

2.3.4 Segmentation Prediction

In AffinityNet[4], the segmentation model is constructed by adding two atrous convolution layers on top of the backbone. The segmentation predictions are then supervised using refined CAMs, which are generated by propagating pairwise pixel

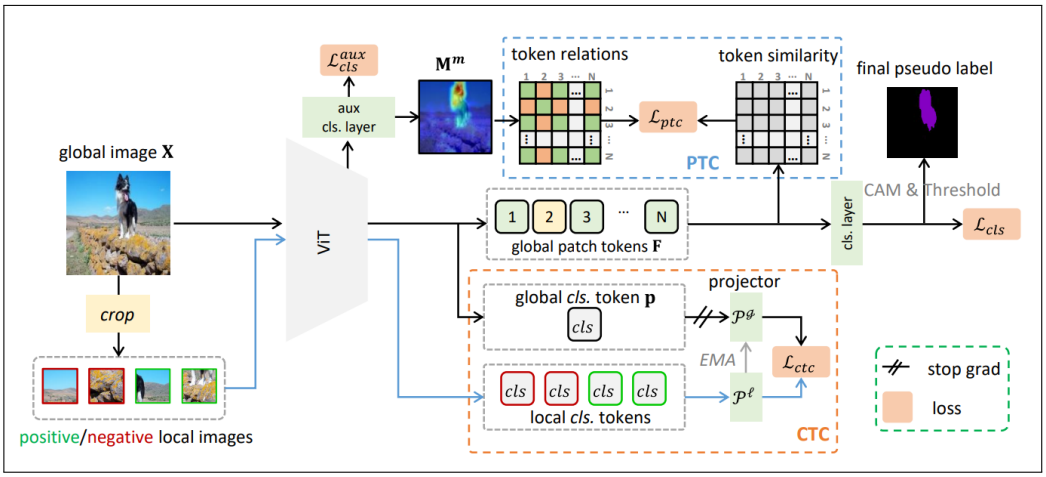


Figure 2.12: Overall framework of ToCo (adapted from [43]).

affinities.

For DSRG[20], once seed cues are obtained from initial CAMs, an image semantic segmentation network is trained with these cues. A balanced seeding loss is applied so that the network's predictions are encouraged to match only the seed regions defined by the classification network, while ignoring unlabeled pixels. As training progresses, the seed cues are iteratively refined into pseudo labels, and the segmentation model is retrained with the updated supervision.

In ToCo[43], the segmentation decoder is deliberately simple, consisting of two 3×3 convolutional layers (each with dilation rate 5) followed by a 1×1 prediction layer. The pseudo labels produced by ToCo are passed through a Pixel-Adaptive Refinement (PAR) module, and the improved labels serve as supervision for the decoder to generate the final segmentation output.

For AFA[42], an MLP-based decoder head is employed. This head fuses features across multiple levels through lightweight MLP layers to produce segmentation predictions. The supervision is provided by refined pseudo labels obtained through affinity propagation.

In WeCLIP[61], segmentation is guided by a feature decoder that extracts intermediate outputs from each transformer block of the CLIP image encoder. For each feature map, a dedicated MLP is applied to generate enhanced feature representations, which are concatenated and then passed through a convolutional layer to form a fused feature map. This fused representation is subsequently processed by multiple stacked multi-head transformer blocks, producing the final segmentation prediction P.

2.3.5 Refinement of Pseudo Labels

A major challenge in weakly supervised semantic segmentation is the generation of high-quality pseudo labels from weak annotations. The quality of these pseudo labels directly impacts the performance of the segmentation model. Therefore, refinement techniques are often employed to improve the quality of the pseudo labels. These techniques can include post-processing methods, such as conditional random fields (CRFs), or affinity based methods that leverage the relationships between pixels to refine the segmentation results.

AffinityNet [4] is a notable example of a refinement technique that uses affinity information to improve the segmentation results. After generating initial CAMs from image-level labels, AffinityNet trains a CNN network to classify whether two adjacent pixels belong to the same semantic region. A random walk algorithm, using these affinities, then propagates object labels across the image, enhancing coherence of the segmentation masks.

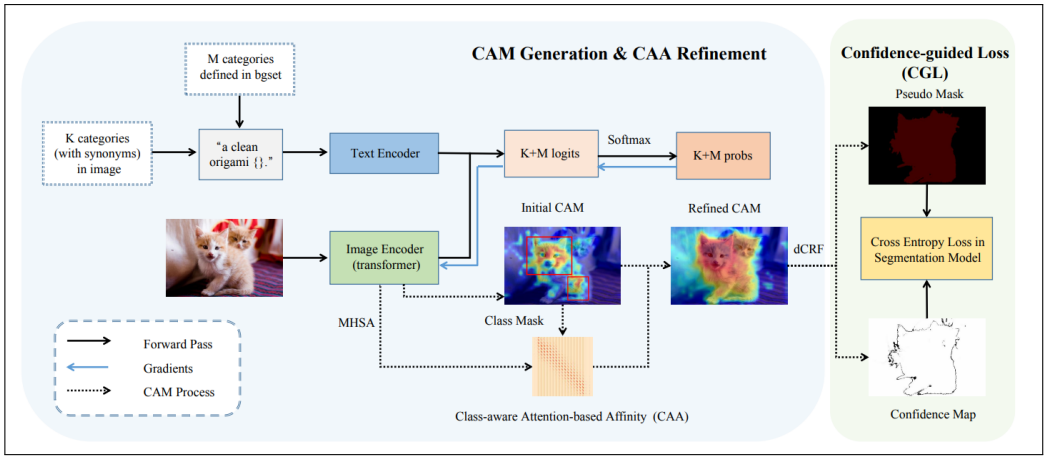


Figure 2.13: Overview of CLIP-ES framework (adapted from [31]).

AFA [42] advanced this idea by showing that pixel affinities can be efficiently derived from attention maps of vision transformers, rather than training a separate affinity prediction network. Specifically, AFA extracts affinity information from the self-attention layers of the transformer backbone, using which it performs random walk propagation like AffinityNet. This approach simplifies the refinement pipeline and benefits from the strong semantic relationships already captured by attention mechanisms. In addition, it introduces the Pixel Adaptive Refinement (PAR) module, which uses local color and spatial information to further refine the initial pseudolabels. PAR is based on pixel adaptive convolution and less costly compared to DCRFs. It also dilates the kernels to cover a larger area, allowing it to capture more contextual information.

CLIP-ES [31] uses a similar approach, but instead of using the attention maps from the transformer, it uses the attention maps from the CLIP model. The CLIP model is a powerful vision-language model that has been shown to be effective in object detection and segmentation tasks.

By leveraging the attention maps from the CLIP model, CLIP-ES is able to generate high-quality pseudo labels that are more accurate and robust than those generated by traditional methods. It uses the random walk propagation like AFA, along with a bounding box mask to mask certain affinities while covering more object regions, as shown in Figure 2.13.

WeCLIP[61] is another technique that uses the CLIP model to generate pseudo labels. In addition to using the attention maps from the CLIP model, it uses the decoder to generate pseudo labels. Since its backbone is frozen, it requires a decoder to enable the model to learn the affinities of the features. Then it weights the decoder affinities with the frozen CLIP attention maps to get the final affinity map. After that, it uses the random walk propagation with box mask like CLIP-ES and applies the PAR module of AFA to get the final segmentation pseudo labels.

2.4 WSSS Training Approaches

Weakly supervised semantic segmentation has two main solutions based on their training processes: multistage approaches [3, 21, 27] and single stage approaches [42, 60].

2.4.1 Multistage

Multistage WSSS has multiple steps. Typically, a classification model is first trained to generate the CAM, which are then converted into initial pixel-level pseudo labels. These pseudo labels are refined through affinity matrix and random walk propagation [4, 42], or seeded region growing [20] or contrastive learning. Then these refined pseudo labels are used for supervising a segmentation model.

Du et al. [16] introduced a pixel-to-prototype contrastive method that enforces semantic consistency at the feature level, leading to improved pseudo-label quality. MCTformer [56] extended transformer-based models with multiple class tokens, enabling the generation of category-specific attention maps for refined CAMs. More recently, researchers have incorporated CLIP into the WSSS pipeline. For instance, CLIMS [55] used CLIP to highlight more complete object regions while suppressing

confident background activations. Similarly, CLIP-ES [31] applied a softmax-based GradCAM [44] guided by carefully designed text prompts, allowing CLIP to produce reliable pseudo labels for segmentation supervision.

2.4.2 Single Stage

In single-stage weakly supervised semantic segmentation, the model tries to learn segmentation directly from weak supervision (like image-level labels) in one shot. A network is trained that produces segmentation outputs without going through separate phases of label generation and refinement. There's no explicit intermediate step to improve pseudo labels — the model directly predicts segmentation maps during training based on weak supervision signals.

Earlier work in this line used ImageNet-pretrained backbones [14] to jointly optimize classification and segmentation, with most efforts devoted to improving supervision quality or constraining the learning process. AA&AR [63] proposed an adaptive affinity loss that facilitates semantic propagation within the segmentation branch. AFA [42] introduced an affinity branch to refine CAMs online, yielding stronger pseudo labels during training. ToCo [43] further advanced this direction by employing token-level contrastive learning to mitigate over-smoothing in CAM generation, thereby providing better online supervision.

In summary, weakly supervised semantic segmentation has evolved significantly, with various types of weak supervision being explored. Image-level labels have emerged as a cost-effective form of supervision, leading to the development of techniques like Class Activation Maps (CAMs) for localization. Numerous methods have been proposed to refine these CAMs into high-quality pseudo labels, leveraging affinity matrices, random walk propagation, and contrastive learning. Both multistage and single-stage training approaches have been employed, each with its own advantages. Recent advancements have also integrated powerful models like CLIP to enhance pseudo label generation. Overall, these developments have contributed to more effective and scalable WSSS models.

Chapter 3

Proposed Methodology

In our proposed methodology, we aim to leverage the strengths of both the UniCL framework and the Swin Transformer architecture to enhance the performance of our model. The methodology is structured into several key components, each addressing a specific aspect of the overall process. The following sections provide a detailed overview of each component, including the backbone architecture, CAM generation, pseudo label generation, segmentation prediction, and refinement.

3.1 Overview

In this section, we provide a high-level overview of the proposed methodology. The methodology is designed to address the challenges of weakly supervised semantic segmentation (WSSS) by leveraging the strengths of multi-modal learning frameworks like UniCL and advanced vision architectures such as the Swin Transformer. The key components of the methodology include:

- **Feature Extraction from Backbone**: The backbone network is responsible for generating Class Activation Maps (CAMs) from input images. We experiment with UniCL as the backbone due to its multi-modal capabilities and state-of-the-art performance in generating high-quality CAMs.
- **CAM Generation**: CAMs are generated using Grad-CAM techniques adapted for multi-modal models like UniCL. These CAMs highlight the regions of interest in the image for each class.
- **Segmentation Prediction**: The feature maps from the backbone are aggregated and passed through a decoder to produce segmentation predictions.

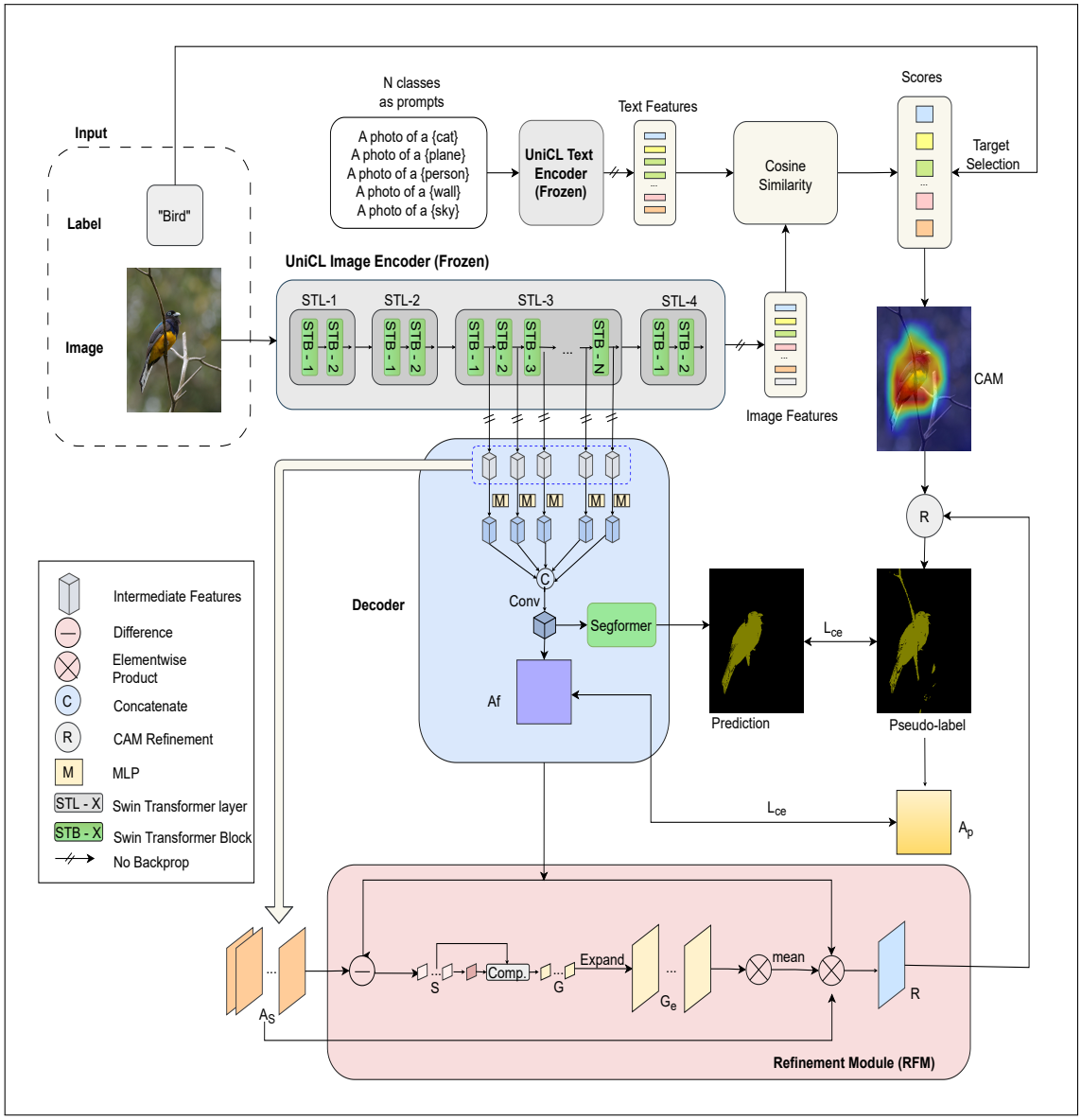


Figure 3.1: Architecture of *UniCL-AffSeg*

The decoder employs multi-head transformer layers to refine the predictions.

- **CAM Refinement and Pseudo-Label Generation**: Initial CAMs are refined using affinity-based techniques and random walk propagation to generate high-quality pseudo-labels. These pseudo-labels serve as supervision for training the segmentation model.
- **Loss Function**: A combination of segmentation loss and affinity loss is used to optimize the model. The affinity loss ensures consistency between the affinity maps of the decoder and the pseudo-labels, while the segmentation loss aligns the predictions with the refined CAMs.

The proposed methodology integrates these components into a cohesive framework, enabling the generation of accurate segmentation maps from weak supervision signals. Figure 3.1 illustrates the overall architecture of the proposed methodology, highlighting the flow of data through each component.

3.2 Feature Extraction from Backbone

In weakly supervised semantic segmentation (WSSS), two components play a crucial role: class activation maps (CAMs) and pseudo labels. The CAMs are first generated from the backbone network, and these are subsequently used to construct pseudo labels. The pseudo labels then serve as supervision for training the segmentation head—or the full segmentation model in the case of a multi-stage approach.

The accuracy of the initial CAMs directly affects the quality of the pseudo labels and, consequently, the performance of the segmentation model. To achieve robust feature representations, we employ UniCL [58], a multi-modal contrastive learning model built upon the CLIP framework. In this section, we first introduce CLIP, outlining its core architecture and characteristics. We then discuss UniCL and its innovations over CLIP, followed by a description of the Swin Transformer backbone that underpins the image encoding process.

3.2.1 Contrastive Language-Image Pre-Training (CLIP)

CLIP [38] is a multi-modal learning framework that utilizes contrastive learning to align images and text within a shared embedding space. It is designed to acquire representations that generalize well across different tasks without the need for task-specific fine-tuning. By training on extensive datasets of image-text pairs, CLIP learns to capture meaningful associations between visual and textual modalities.

3.2.2 CLIP Architecture

CLIP Components

CLIP comprises two principal components:

- **Image Encoder**: Responsible for processing images and extracting feature representations. This typically involves a Swin Transformer, pooling layers, and additional modules to capture spatial hierarchies and relationships within image data.
- **Text Encoder**: Processes textual data and converts it into a format suitable for comparison with image features. It generally includes transformer layers and attention mechanisms to capture semantic relationships within the text.

CLIP is trained using a contrastive loss function that aligns image and text features in a shared embedding space. This enables the model to learn representations applicable to a variety of tasks, such as image classification, object detection, and image segmentation. In the context of this work, CLIP's capabilities are leveraged for weakly supervised semantic segmentation, which relies solely on image-level classification.

3.2.3 Characteristics of CLIP

Multi-Modal Learning

Multi-modal learning involves integrating information from different modalities, such as images and text, to enhance model performance. By combining complementary cues from multiple sources, models can develop a deeper and more comprehensive understanding of the data. For instance, in classification tasks, leveraging both visual and textual data enables the model to capture more nuanced semantic relationships. A key advantage of this approach is the potential for zero-shot learning, where the model can recognize and classify images from previously unseen categories. This is accomplished by aligning visual features with corresponding textual descriptions, allowing the model to generalize beyond the training set. Such capabilities are particularly valuable, as classification often underpins many downstream applications.

Contrastive Learning

Contrastive learning is a self-supervised approach focused on learning representations by distinguishing between positive and negative pairs. The objective

is to bring representations of related inputs (such as an image and its corresponding textual description) closer in a shared embedding space, while pushing apart representations of unrelated inputs. This is typically achieved using a contrastive loss function, which minimizes the distance between positive pairs and maximizes it for negative pairs. Such a strategy enables models to acquire robust, discriminative, and generalizable feature representations.

Training Strategy

The training strategy for multi-modal learning generally integrates both supervised and self-supervised techniques. Models are trained on large-scale datasets containing paired samples from different modalities, facilitating the learning of meaningful representations through methods such as contrastive learning. Data augmentation is commonly applied to enhance the diversity and robustness of the learned features, thereby improving the model's generalization to unseen data.

Task-Agnostic Learning

Task-agnostic learning refers to the development of representations that are not specialized for a single task but are applicable across a range of downstream applications. This is accomplished by focusing on learning generalizable features through training on diverse datasets and leveraging multi-modal information. As a result, models can adapt their learned representations to various tasks, including image classification, object detection, and image segmentation. This adaptability makes task-agnostic learning a valuable approach for constructing versatile and reusable models.

3.2.4 Unified Contrastive Learning (UniCL)

UniCL, or Unified Contrastive Learning [58], is a multi-modal learning framework that extends the ideas of CLIP by unifying different contrastive learning strategies within a single architecture. This enables more effective and flexible learning from multiple modalities.

UniCL [58] is more than a straightforward extension of CLIP; it introduces several important innovations to improve both performance and adaptability.

A key feature of UniCL is its unified approach to contrastive learning, integrating image-label and text-label associations into a joint image-label-text space. This unified framework allows the model to leverage more positive pairs during training. While

CLIP treats only the provided image-text pair as a positive match, UniCL recognizes that multiple text descriptions or images can correspond to the same category. For instance, both "a photo of a cat" and "a photo of a kitten" may be valid positive pairs for an image of a cat. UniCL's contrastive loss encourages the model to learn similarities across all features sharing the same category, resulting in a more comprehensive and robust shared representation space for images and text.

Unified Image-Text-Label Contrast in UniCL

UniCL employs a bidirectional learning objective between image-text pairs:

$$\min_{\{\theta,\phi\}} \mathcal{L}_{\text{BiC}} = \mathcal{L}_{i2t} + \mathcal{L}_{t2i}, \tag{3.1}$$

where $\mathcal{L}_{i2t}$ and $\mathcal{L}_{t2i}$ are the image-to-text and text-to-image contrastive losses, respectively. And θ and ϕ are the parameters of the image and text encoders, respectively.

To align the representations of images and their corresponding textual descriptions within a batch, the image-to-text contrastive loss is formulated by [58]. This loss encourages the model to associate each image with all relevant text features that share the same class label, thereby strengthening the alignment between visual and textual modalities in the shared embedding space:

$$\mathcal{L}_{i2t} = -\sum_{i\in\mathcal{B}} \frac{1}{|\mathcal{P}(i)|} \sum_{k\in\mathcal{P}(i)} \log \frac{\exp(\tau \mathbf{u}_i^\top \mathbf{v}_k)}{\sum_{j\in\mathcal{B}} \exp(\tau \mathbf{u}_i^\top \mathbf{v}_j)}, \tag{3.2}$$

where $k \in \mathcal{P}(i) = \{k | k \in \mathcal{B}, y_k = y_i\}$, i.e., the set of all images in the batch that belong to the same class as image i.

Conversely, the text-to-image contrastive loss, which aligns each text feature in a batch with its corresponding image features, is formulated as:

$$\mathcal{L}_{t2i} = -\sum_{j\in\mathcal{B}} \frac{1}{|\mathcal{P}(j)|} \sum_{k\in\mathcal{P}(j)} \log \frac{\exp(\tau \mathbf{u}_k^\top \mathbf{v}_j)}{\sum_{i\in\mathcal{B}} \exp(\tau \mathbf{u}_i^\top \mathbf{v}_j)}, \tag{3.3}$$

where $k \in \mathcal{P}(j) = \{k \mid k \in \mathcal{B}, y_k = y_j\}$, i.e., the set of all text features in the batch that belong to the same class as text j.

Comparison with CLIP

In case CLIP [38], for an image, there is only one positive text feature. In other words, $\mathcal{P}(i) = i \in \mathcal{B}$; and $\mathcal{P}(j) = j \in \mathcal{B}$. So, the image-to-text contrastive loss is defined as:

$$\mathcal{L}_{i2t} = -\sum_{i \in \mathcal{B}} \log \frac{\exp(\tau \mathbf{u}_i^\top \mathbf{v}_i)}{\sum_{j \in \mathcal{B}} \exp(\tau \mathbf{u}_i^\top \mathbf{v}_j)}, \tag{3.4}$$

where $i \in \mathcal{B}$ is the index of the image in the batch.

And the text-to-image contrastive loss is defined as:

$$\mathcal{L}_{t2i} = -\sum_{j \in \mathcal{B}} \log \frac{\exp(\tau \mathbf{u}_j^\top \mathbf{v}_j)}{\sum_{i \in \mathcal{B}} \exp(\tau \mathbf{u}_i^\top \mathbf{v}_j)}, \tag{3.5}$$

where $j \in \mathcal{B}$ is the index of the text feature in the batch.

The equations 3.1-3.5 are taken from [58].

This means that $\mathcal{L}_{BiC}$ becomes CLIP training objective. The main property in Equation 3.3 is that for each image feature, any of the text features in the batch can be used as a positive pair. And so is the case for Equation 3.3.

Second, UniCL replaces CLIP's ViT backbone with a Swin Transformer [32] backbone. The Swin Transformer is a hierarchical vision transformer that captures both local and global information in images, making it more suitable for various vision tasks.

3.2.5 Swin Transformer

The Swin Transformer [32] is a hierarchical vision transformer architecture that utilizes a shifted windowing mechanism to effectively capture both local and global image information. It is engineered for computational efficiency while delivering strong performance across a range of computer vision tasks. The model is organized into multiple stages, each comprising distinct transformer blocks, which enables the processing of images at varying resolutions and facilitates the extraction of multi-scale features.

Figure 3.2 presents a comparative overview of the Swin Transformer and Vision Transformer (ViT) architectures. The principal difference between the two lies in their respective processing paradigms: the Swin Transformer adopts a hierarchical framework, operating on feature maps at multiple scales, whereas ViT processes the

entire image globally at a single resolution. This hierarchical design empowers the Swin Transformer to simultaneously capture fine-grained local details and broader contextual information, making it particularly advantageous for applications that demand both.

Within the Swin Transformer, hierarchical feature maps are constructed by progressively merging image patches (shown in gray) at deeper network layers. Self-attention is computed exclusively within localized windows (indicated in red), resulting in linear computational complexity with respect to the input image size. This approach not only enhances efficiency but also renders the Swin Transformer highly suitable for dense prediction tasks, such as image segmentation and classification.

Conversely, conventional vision transformers like ViT generate feature maps at a single, lower resolution and apply self-attention globally, which incurs quadratic computational complexity relative to the input size. While global attention can be beneficial for certain tasks, it is less efficient for dense recognition scenarios compared to the localized attention employed by the Swin Transformer.

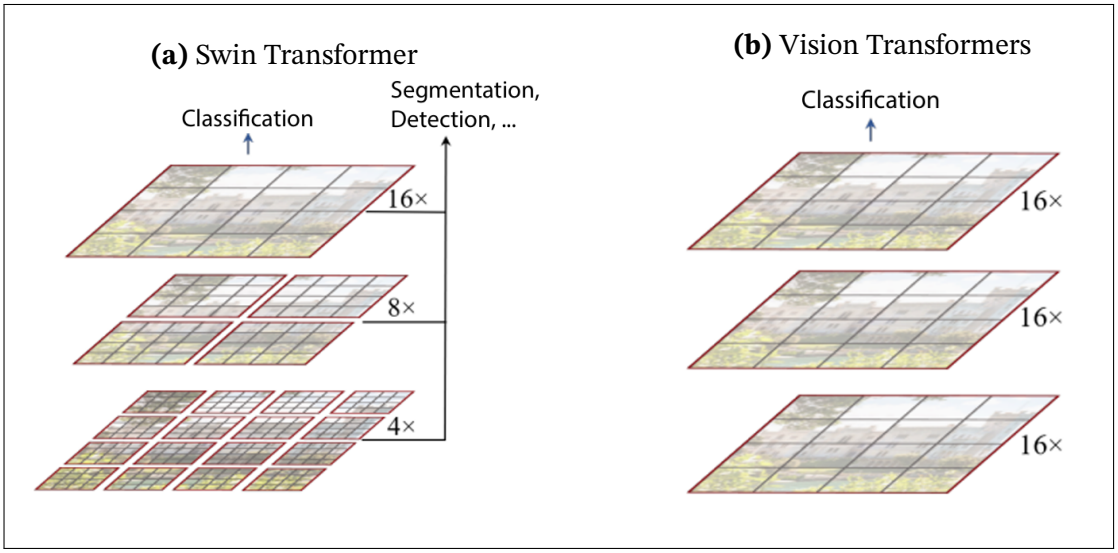


Figure 3.2: Comparison of Swin Transformer [32] and ViT [15] Architectures. Image taken from [32].

Owing to its hierarchical structure, the Swin Transformer is exceptionally well-suited for tasks that necessitate the integration of global context—essential for discerning the overall image structure—and local information, which is critical for identifying intricate features such as edges and textures. This makes it a compelling choice for image segmentation applications.

Furthermore, integrating the contrastive and multi-modal learning strengths of

UniCL with the Swin Transformer's hierarchical design can significantly enhance the model's capacity to learn robust and discriminative representations from both visual and textual data. This synergy facilitates a more holistic understanding of multimodal inputs, thereby improving overall model performance.

With this foundational context established, the subsequent section details the proposed methodology, which comprises several core components: the backbone architecture, class activation map (CAM) generation, pseudo-label creation, segmentation prediction, and refinement. Each of these elements is integral to optimizing the model's effectiveness.

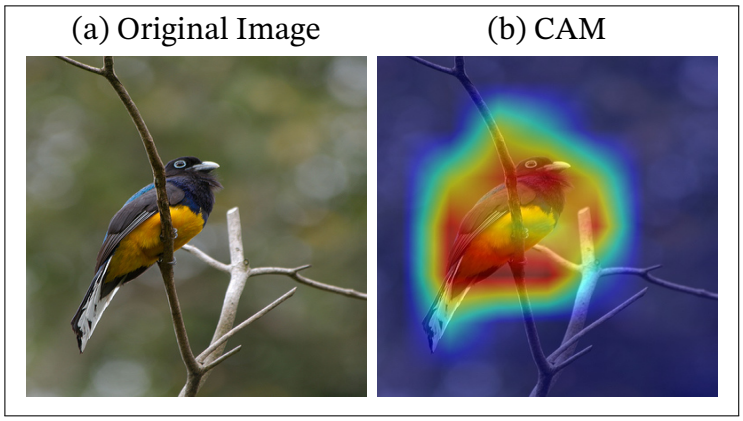


Figure 3.3: Original image and CAM generated by our method for class *Bird*.

3.3 CAM Generation

Figure 3.4 shows the process of generating CAMs using CLIP/UniCL. We have used Grad-CAM. The process is similar to the method discussed in section 2.2.3, which is used to generate CAMs using CNN-based models. But there are some differences.

AS CLIP is a multi-modal model, the main classification task is to find similarity between the image and text. If we want to generate the CAM for N classes, we need to provide N text features as input. So, at the very beginning, we need to encode the text features using the CLIP text encoder, and store them.

An MLP layer takes the linear combination of the input features and the weights. We do the same thing when finding similarity score between the image and text features, take the linear combination of the image features and the text features. So, we can consider these text features as the "neuron weights" of the classifier head.

Then, we need to pass the image through the CLIP image encoder to get the feature maps. The rest of the process is the same as Grad-CAM. An example of the produced

CAM for the *Motorbike* class is shown in Figure 3.3.

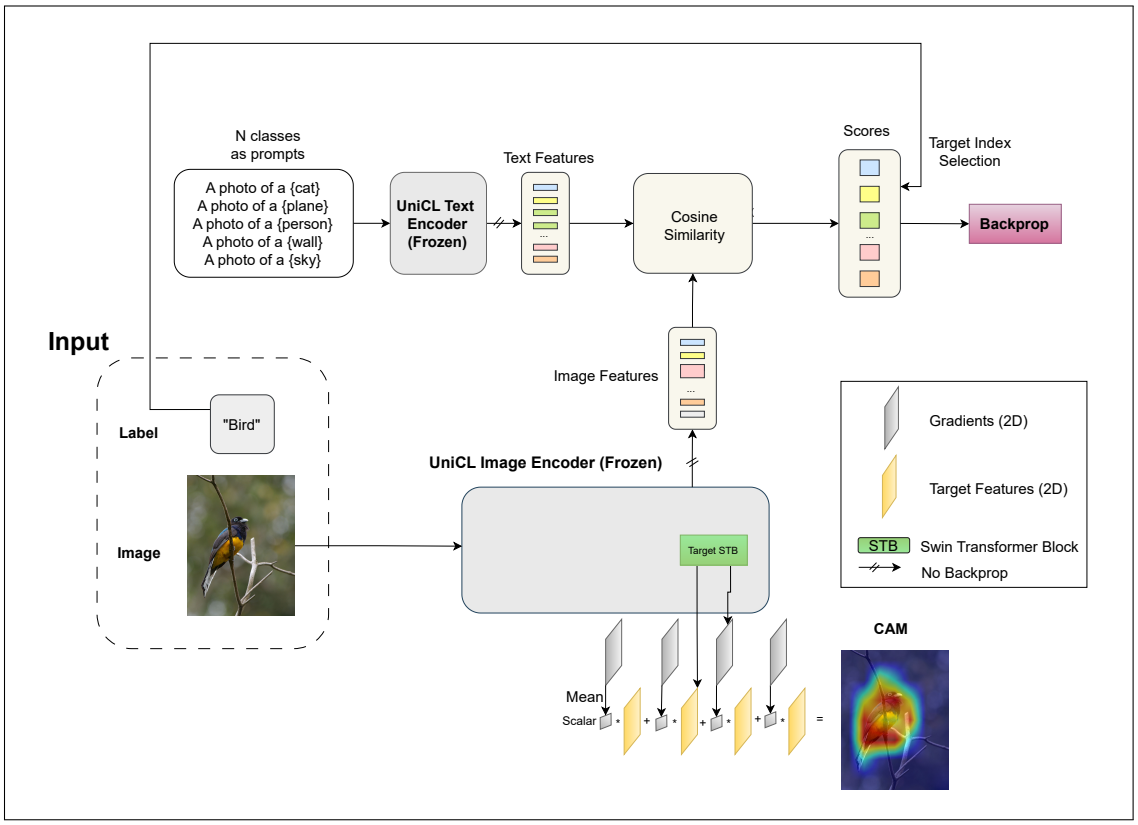


Figure 3.4: CAM Generation Process using UniCL

3.4 Segmentation Prediction

The segmentation prediction module is responsible for producing pixel-level class predictions for the input image. It comprises two primary stages: aggregation of encoder features and a decoder built with multi-head transformer layers. The design is inspired by the approach in [61].

3.4.1 Encoder Feature Aggregation

In accordance with the methodology outlined in [61], the feature representations extracted from each transformer block of the UniCL image encoder [58] are forwarded to the decoder for further processing. Denote the output feature map from the l-th transformer block as F_l^{init}, where $l = 1, \ldots, N$. To harmonize the feature dimensions and enhance representational capacity, each F_l^{init} is individually transformed via a dedicated multilayer perceptron (MLP). Specifically, the

transformation is defined as:

$$F_l^{\text{new}} = W_1^{\text{fc}} \left(\text{ReLU} \left(W_2^{\text{fc}} (F_l^{\text{init}}) \right) \right), \tag{3.6}$$

where W_1^{fc} and W_2^{fc} denote learnable fully connected layers, and ReLU$(\cdot)$ represents the rectified linear unit activation function. This procedure ensures that the multi-scale features from different encoder stages are suitably projected for subsequent aggregation and decoding.

After that, all new feature maps F_l^{new} for $l = 1, \ldots, N$ are concatenated together, which are then processed by a convolution layer to generate a fused feature map F_u [61]:

$$F_u = \text{Conv} \left(\text{Concat} \left[F_1^{\text{new}}, F_2^{\text{new}}, \ldots, F_N^{\text{new}} \right] \right), \tag{2}$$

where $F_u \in \mathbb{R}^{d \times h \times w}$, and d, h, and w represent the channel dimension, height, and width of the feature map, respectively. Conv$(\cdot)$ is a convolutional layer, and Concat$[\cdot]$ denotes the concatenation operation.

3.4.2 Generating Final Prediction

Several sequential multi-head transformer layers are designed to generate the final prediction P [61]:

$$P = \text{Conv}(\phi(F_u)) \uparrow, \tag{3.7}$$

where $P \in \mathbb{R}^{C \times H \times W}$, C is the number of classes including background, and ϕ represents the sequential multi-head transformer blocks [12]. Each transformer block contains a multi-head self-attention module, a feed-forward network, and two normalization layers. The operator $\uparrow$ denotes an upsampling operation to align the prediction map size with the original image.

3.5 CAM Refinement and Pseudo-Label Generation

3.5.1 Affinity based CAM Refinement

The initial CAMs produced by the backbone tend to focus only on sparse, highly discriminative regions. These activations are often noisy and insufficient to be used directly as supervision. Therefore, a refinement module is required where the initial CAM is improved. The refined CAM is then used as a pseudolabel for supervision.

The most common strategy for rectifying CAMs is by exploiting the feature

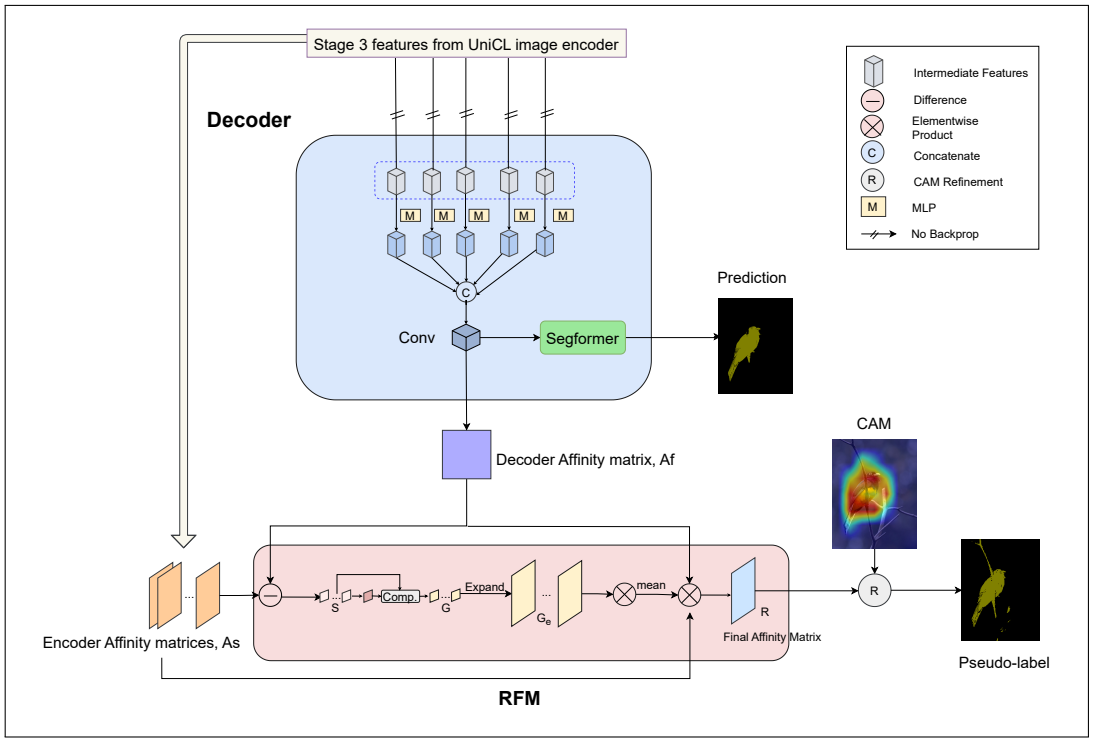


Figure 3.5: Decoder and Refinement Module

relationships within the backbone that generates them. This is referred to as affinity-based CAM refinement, first introduced by [4]. In transformer-based backbones, the attention maps naturally encode semantic-level affinities between tokens or patches. From these attention maps, an affinity matrix A_f can be constructed and employed for CAM refinement via random walk propagation. In parallel, another affinity matrix, A, can be derived from the predicted pseudolabel produced by the decoder. The cross-entropy loss between these two affinity matrices is then incorporated into the training objective.

However, in our case, the backbone responsible for generating the CAMs is **frozen**, meaning that its attention maps remain fixed and do not update during training. To address this limitation, Frozen CLIP [61] proposes utilizing the affinity derived from intermediate features of the encoder instead of directly relying on the backbone's frozen attention. The backbone attention maps are then used only to influence this encoder-derived affinity, producing a final refined matrix R, which is further transformed into a transition matrix T for random walk propagation.

In our setup, the UniCL backbone is based on the Swin Transformer, whose shifted-window design restricts the attention mechanism to **local regions**. Consequently, the attention maps fail to capture the **global semantic affinities** necessary for effective CAM propagation, and we could not use them directly for

refinement. To overcome this, we compute affinities from the **intermediate feature maps of the Swin encoder**.

Formally, let $f^{(k)} \in \mathbb{R}^{C \times L}$ denote the feature map from the k-th encoder block, where C is the channel dimension and $L = h \times w$ represents the flattened spatial resolution. Each feature map is first normalized along the channel dimension:

$$\tilde{f}_i^{(k)} = \frac{f_i^{(k)}}{\|f_i^{(k)}\|_2}, \quad i = 1, \ldots, L \tag{3.8}$$

The affinity between spatial positions i and j within block k is then computed as the dot-product similarity:

$$A_{ij}^{(k)} = \tilde{f}_i^{(k)\top} \tilde{f}_j^{(k)} \tag{3.9}$$

We compute such affinity matrices across multiple intermediate encoder blocks to capture complementary semantic relations at different abstraction levels. The resulting set of affinities is represented as:

$$\mathcal{A} = \{A^{(1)}, A^{(2)}, \ldots, A^{(N)}\}, \quad \mathcal{A} \in \mathbb{R}^{N \times hw \times hw} \tag{3.10}$$

This multi-block affinity representation provides richer contextual cues for propagating and refining class activation maps.

3.5.2 Extracting Affinity Map from the Decoder

Based on the feature map F_u obtained from the decoder (as defined in Eq. (2)), an affinity map is constructed as follows:

$$A_f = \text{Sigmoid}(F_u^\top F_u), \tag{3.11}$$

where $F_u \in \mathbb{R}^{d \times h \times w}$ is reshaped into a matrix of size $d \times hw$ before the computation. The Sigmoid$(\cdot)$ function ensures that the values in the resulting affinity map lie in the range $[0, 1]$. Consequently, the computed affinity map A_f has the dimensions $\mathbb{R}^{hw \times hw}$. Here, $\top$ denotes the matrix transpose.

3.5.3 Selecting Affinity Maps of the Image Encoder

Following [61], we extract all the affinity maps from the frozen CLIP image encoder, denoted as $\{A^l\}_{l=1}^N$, where each affinity map $A^l \in \mathbb{R}^{hw \times hw}$. To evaluate the reliability of each affinity map A^l, we compare it with the previously computed decoder affinity map A_f using the following deviation score:

$$S^l = \sum_{i=1}^{hw} \sum_{j=1}^{hw} \left| A_f(i,j) - A^l(i,j) \right|, \tag{3.12}$$

where S^l quantifies how much the affinity map (encoder) A^l deviates from the reference affinity (decoder) A_f.

Based on this score, we assign a binary weight G^l to each affinity map:

$$G^l = \begin{cases} 1, & \text{if } S^l < \dfrac{1}{N - N_0 + 1} \sum_{m=N_0}^{N} S^m, \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

Here, $G^l \in \{0,1\}$ is expanded to $G_e^l \in \mathbb{R}^{hw \times hw}$ for subsequent operations. The threshold is chosen as the average of the scores $\{S^m\}_{m=N_0}^N$ from the later layers of the encoder. If S^l is lower than this threshold, the corresponding affinity map is considered reliable and retained ($G^l = 1$); otherwise, it is filtered out ($G^l = 0$).

This selection mechanism ensures that only high-quality affinity maps consistent with the encoder-derived relationships are preserved, thereby improving the robustness of the CAM refinement process.

3.5.4 Utilizing the Frozen Encoder Affinity Maps

Using the affinity map A_f and the filtered affinity maps of the encoder, we construct a refining map R by weighing the decoder affinity matrix with the mean of the selected encoder affinity maps. This method of selection is adopted first by [61]. The equation is given as:

$$R = A_f \odot \frac{\sum_{l=1}^{N} G_e^l A^l}{N_m}, \tag{3.14}$$

where $\odot$ denotes element-wise multiplication, and N_m is the number of valid encoder affinity maps that have passed the filtering stage, defined as:

$$N_m = \sum_{l=N_0}^{N} G^l. \tag{3.15}$$

Here, $G_e^l \in \mathbb{R}^{hw \times hw}$ is the expanded binary filter corresponding to G^l. The refining map R effectively integrates the reliable encoder affinity maps weighted by the affinity map A_f, enhancing meaningful relationships while suppressing noisy or irrelevant information.

3.5.5 Constructing the Semantic Transition Matrix

Now that we have the affinity matrix R, to apply the random walk propagation, we need to at first normalize the rows for converting it into a transition probability matrix. In addition, column normalization ensures consistency and symmetric behavior. Sinkhorn normalization [45] is thus used to convert R into a double stochastic matrix R_{nor}.

Next, R_{nor} is converted into a symmetric matrix by adding its transpose and normalizing the sum:

$$T = \frac{R_{\text{nor}} + R_{\text{nor}}^T}{2}, \quad \text{where} \quad R_{\text{nor}} = \text{Sinkhorn}(R)$$

The matrix T is now symmetric whose rows and columns are normalized. This is the transition probability matrix to be used for random walk.

3.5.6 Random Walk Propagation

The refined class activation map (CAM) for a specific class c, denoted as M_f^c is found by random walk propagation as follows:

$$M_f^c = B^c \odot T^\alpha \cdot M_{\text{init}}^c \tag{3.16}$$

where, $M_{\text{init}}^c \in \mathbb{R}^{hw \times 1}$ is the initial CAM for class c, reshaped into vector form. T is the transition probability matrix, α is the hyperparameter that controls the strength of the refinement we want to apply. It is equivalent to the number of iterations of the random walk propagation. $B^c \in \mathbb{R}^{1 \times hw}$ is the box mask obtained from the CAM of class c, and $\odot$ denotes the Hadamard product. This masking is needed to restrict the refining region spatially. It is obtained for each target class c by thresholding the

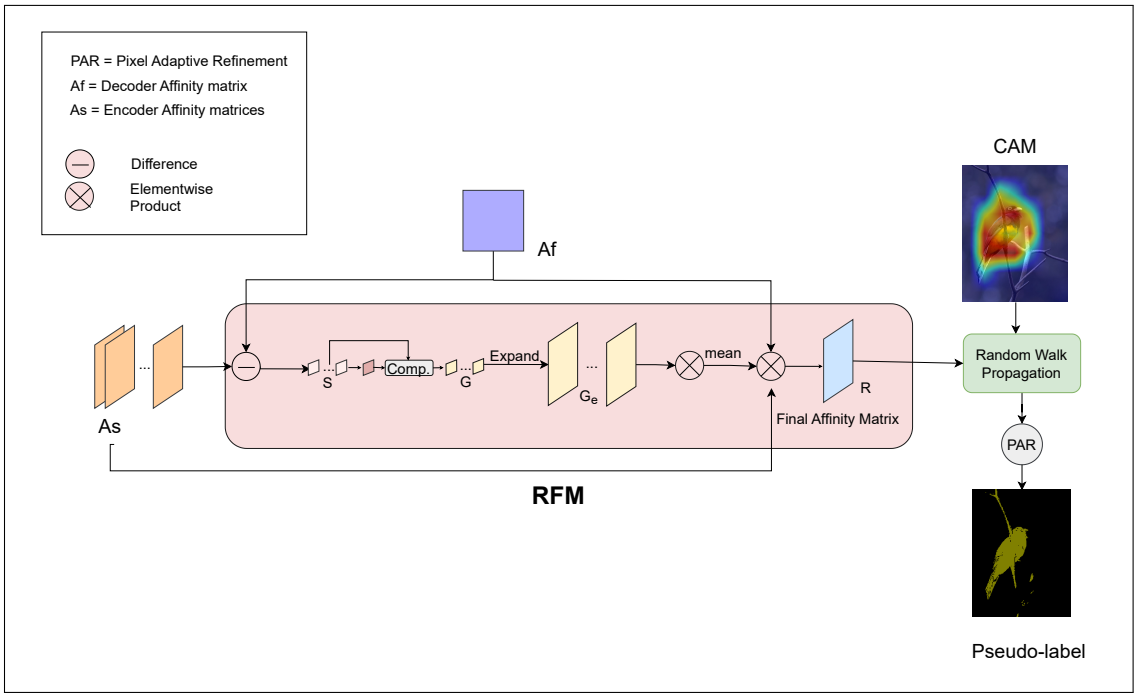


Figure 3.6: CAM refinement by exploiting affinity map

CAM of this class by a constant. The connected regions of the mask map are found and covered by minimum rectangle bounding boxes. These boxes mask the affinity of distant pixels, preventing over-expansion.

Finally, the refined CAM M_f^c is passed through an online post-processing module - specifically, the pixel-adaptive refinement module proposed in [42].

3.5.7 Pixel Adaptive Refinement module

To ensure local consistency, pixel-adaptive convolution is introduced by [42]. It uses local RGB and spatial information to define the low-level pair-wise semantic affinity of two pixels. Given an input image $I \in \mathbb{R}^{h \times w \times 3}$, the pairwise affinities between two pixels at positions (i, j) and (k, l) is calculated as follows:

- $\kappa^{\text{rgb}}_{ij,kl}$ measures color similarity (the closer the colors, the higher the affinity).

- $\kappa^{\text{pos}}_{ij,kl}$ measures spatial closeness (pixels closer in space have higher affinity).

These are defined as:

$$\kappa^{\text{rgb}}_{ij,kl} = -\frac{\|I_{ij} - I_{kl}\|^2}{w_1 \sigma^2_{\text{rgb}}}, \quad \kappa^{\text{pos}}_{ij,kl} = -\frac{\|P_{ij} - P_{kl}\|^2}{w_2 \sigma^2_{\text{pos}}}$$

Here: - I_{ij} and P_{ij} are the RGB values and 2D spatial coordinates at pixel (i, j), - σ_{rgb}

and σ_{pos} are the standard deviations of color and position differences, - w_1 and w_2 are weights that control smoothness.

Then, the affinity kernel $\kappa_{ij,kl}$ for each pixel pair is computed by applying a softmax to normalize both terms across the local neighborhood $\mathcal{N}(i, j)$, and combining them:

$$\kappa_{ij,kl} = \frac{\exp(\kappa^{\text{rgb}}_{ij,kl})}{\sum_{(x,y)\in\mathcal{N}(i,j)} \exp(\kappa^{\text{rgb}}_{ij,xy})} + w_3 \cdot \frac{\exp(\kappa^{\text{pos}}_{ij,kl})}{\sum_{(x,y)\in\mathcal{N}(i,j)} \exp(\kappa^{\text{pos}}_{ij,xy})}$$

Here, w_3 is a weight to balance the influence of the position term.

This affinity kernel is used to iteratively refine a CAM $M \in \mathbb{R}^{h\times w\times C}$. At iteration t, each pixel value $M^t_{i,j,c}$ for class c is updated by aggregating class scores from neighboring pixels weighted by the affinity:

$$M^t_{i,j,c} = \sum_{(k,l)\in\mathcal{N}(i,j)} \kappa_{ij,kl} \cdot M^{t-1}_{k,l,c} \tag{3.17}$$

The neighborhood $\mathcal{N}(i, j)$ is defined as the 8-connected neighbors (i.e., a 3×3 window) with multiple dilation rates allowing the refinement to capture both local and semi-local context.

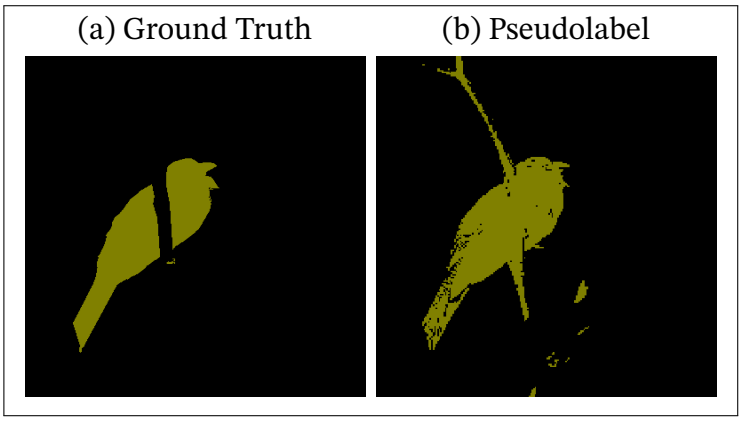


Figure 3.7: Ground Truth Label and Pseudolabel generated by our method for class *Bird*.

3.5.8 Pseudo-Label Generation

The refined CAMs are used to generate the pseudo-labels or segmentation labels, which assign a class label to each pixel (patch),

$$M_p(x, y) = \arg \max_{c \in \{1,\dots,C\}} M_c(x, y) \tag{3.18}$$

where $M_c \in \mathbb{R}^{h \times w}$ is the refined CAM and $M_p \in \mathbb{R}^{h \times w}$ is the pseudo-label. The generated pseudo-labels are used as supervision for the segmentation network.

3.6 Loss Function

The loss function defined by [61]:

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda \mathcal{L}_{aff} \tag{3.19}$$

where, $\mathcal{L}_{aff}$ is the affinity loss and $\mathcal{L}_{seg}$ is the segmentation loss. λ is the weighting parameter.

3.6.1 Affinity Loss

In the refinement module, we used the affinity map of the decoder features, A_f (Equation 3.11). The quality of the final pseudolabel directly depends on it. We need to ensure that the pairwise affinity of the pseudolabel matches the pairwise affinity of the decoder output, because that is what the random walk propagation implies. We compute the affinity labels for the predicted pseudolabel as follows:

$$A_p = O_h(M_p)^T O_h(M_p) \tag{3.20}$$

where $O_h(.)$ is the one-hot encoding of M_p, and $A \in \mathbb{R}^{hw \times hw}$ is the affinity label. . This means that the value of $A_p(i, j)$ will be 1 for the (i, j) pair which have the same class label, and 0 otherwise. The affinity loss is the Cross-Entropy Loss of A_f and A_p:

$$\mathcal{L}_{aff} = \mathcal{L}_{ce}(A_f, A_p) \tag{3.21}$$

3.6.2 Segmentation Loss

We obtain the segmentation prediction, P, from the decoder (Equation 3.7). As we are using the refined CAM or pseudolabel, M_p as supervision, the segmentation loss is computed:

$$\mathcal{L}_{seg} = \mathcal{L}_{ce}(P, M_p \uparrow) \tag{3.22}$$

where L_{ce} is the cross-entropy loss, and $M_p \uparrow \in \mathbb{R}^{H \times W}$. H and W are the original height and width of the image respectively.

Chapter 4

Results and Discussion

4.1 Data and Experimental Setup

4.1.1 Dataset

Experiments were performed on the PASCAL VOC 2012 dataset [17], which comprises 20 object classes plus one background category. In line with standard practice, the training set was expanded using additional images from the Semantic Boundaries Dataset (SBD) [19], yielding a total of 10,582 training images. The original VOC 2012 split contains 1,464 training images, 1,449 validation images, and 1,456 test images. As the ground-truth labels for the test set are not publicly accessible, all evaluations of model performance were carried out on the validation set.

4.1.2 Experimental Setup

All experiments were implemented using the **PyTorch** framework and executed on a single **NVIDIA Tesla T4 GPU** with **16 GB memory**, accessed through Google Colab (free tier).

During training, the input images were randomly resized within the range of 512 to 2048 pixels, rescaled by factors between 0.5 and 2.0, and then cropped to a fixed resolution of 224×224.

Model optimization was performed using the **AdamW** optimizer with a learning rate of 5×10^{-5}, momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.01. Training was run for **30,000 iterations** with an effective batch size of 8 (per-

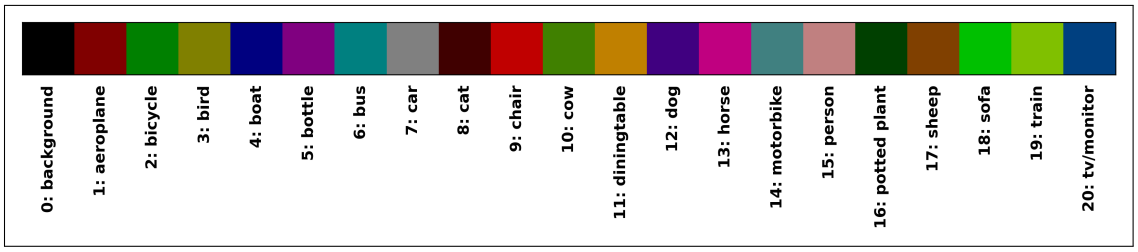


Figure 4.1: Pascal VOC 2012 class color map for visualization. Each class is represented by a unique color for easy identification in segmentation maps.

GPU batch size of 4, with gradient accumulation over 2 steps). A linear learning rate warmup was applied for the first 500 iterations, with a warmup ratio of 10^{-3}. The best model checkpoints were selected based on validation set performance.

4.1.3 Evaluation Metric

The primary evaluation criterion was the **mean Intersection over Union (mIoU)**. For a given class c, the Intersection over Union (IoU) is defined as:

$$\text{IoU}(c) = \frac{TP_c}{TP_c + FP_c + FN_c}$$

where TP_c, FP_c, and FN_c denote the number of true positives, false positives, and false negatives for class c, respectively. The mean IoU is then computed as the average IoU across all K semantic classes:

$$\text{mIoU} = \frac{1}{K} \sum_{c=1}^{K} \text{IoU}(c)$$

This metric is widely used in semantic segmentation as it provides a balanced assessment of model performance across both frequent and infrequent classes.

4.2 Quantitative Analysis

We evaluated the performance of our proposed method on the PASCAL VOC 2012 **validation** and **test** sets using standard metrics: **mean Intersection over Union (mIoU)**, **pixel accuracy (pAcc)**, and **mean accuracy (mAcc)**. The results are summarized in Table 4.2.

On the **validation set**, our method achieves a **mean IoU of 50.3%**, a pixel accuracy of 80.1%, and a mean accuracy of 69.99%. On the **test set**, the model achieves a slightly

higher **mean IoU of 50.8%**, with a pixel accuracy of 79.6% and mean accuracy of 71.36%.

These results indicate that the model generalizes well across both validation and test sets, consistently segmenting dominant classes effectively. In particular, the strong performance on large and distinctive categories such as **background**, **bus**, **sheep**, and **cow** highlights the model's ability to capture prominent visual structures. However, performance remains weaker on small or less frequent categories such as **person**, **chair**, and **bicycle**, suggesting that future improvements should focus on enhancing recognition of rare and small objects.

4.2.1 Per-Class Performance Analysis

To gain a deeper understanding of our model's behavior, we analyzed the per-class Intersection over Union (IoU) scores on both the PASCAL VOC 2012 validation and test sets. This joint analysis highlights how the model performs across different object categories, providing insights beyond the overall metrics.

High-Performing Classes

On the **validation set**, large and distinctive categories achieved strong results such as **background** (78.2%), **bus** (67.8%), **sheep** (62.0%), **horse** (60.9%), and **cat** (65.2%). On the **test set**, similar trends were observed with strong performance for **cow** (72.5%), **bus** (67.8%), **sheep** (65.5%), **horse** (64.9%), and **background** (77.6%).

Low-Performing Classes

On the **validation set**, small or less frequent categories such as **person** (16.9%), **chair** (26.1%), and **potted plant** (28.7%) exhibited the lowest IoUs. On the **test set**, the weakest categories remained consistent, with **person** (16.7%), **chair** (26.1%), and **bicycle** (23.3%) showing poor performance.

Observations

Across both validation and test sets, the model demonstrates strong performance on large and visually distinctive classes such as animals and vehicles, while it consistently underperforms on humans, furniture, and small objects. Notably, some categories show improved generalization on the test set compared to validation — for example, **boat** improves from 43.1% (val) to 50.4% (test), and **sofa** improves from 46.5% (val) to 53.3% (test).

Table 4.1: Per-Class IoU Performance on PASCAL VOC (Validation vs Test Sets)

Index	Class	Val IoU	Test IoU
0	background	0.782	0.776
1	aeroplane	0.566	0.625
2	bicycle	0.330	0.233
3	bird	0.610	0.623
4	boat	0.431	0.504
5	bottle	0.382	0.333
6	bus	0.678	0.678
7	car	0.495	0.436
8	cat	0.652	0.637
9	chair	0.261	0.261
10	cow	0.595	0.725
11	diningtable	0.420	0.346
12	dog	0.660	0.648
13	horse	0.609	0.649
14	motorbike	0.576	0.548
15	person	0.169	0.167
16	pottedplant	0.287	0.333
17	sheep	0.620	0.655
18	sofa	0.465	0.534
19	train	0.558	0.600
20	tvmonitor	0.419	0.355

Table 4.2: Overall Performance Metrics on PASCAL VOC

Metric	Validation	Test
Pixel Accuracy (pAcc)	0.801	0.796
Mean Accuracy (mAcc)	0.700	0.714
Mean IoU (mIoU)	0.503	0.508

A major factor behind the weak performance on certain categories is the composition of the datasets used for UniCL pretraining. UniCL was pretrained on **ImageNet-21K (IN-21K)** and **YFCC-14M**, which together cover a very large number of classes (21K+ in IN-21K) and a broad variety of images, but they do not include or sparsely represent some categories present in PASCAL VOC, such as **person**, **chair**, and **potted plant**. Consequently, the model has limited exposure to these classes during pretraining and struggles to learn transferable representations for them, leading to very low IoU scores — for example, 16.9% (val) and 16.7% (test) for **person**.

This severe underperformance on frequently occurring but underrepresented categories substantially lowers the overall mean IoU, despite strong performance on large and distinctive objects. These findings highlight the importance of incorporating multi-scale features, class-specific augmentation, and addressing

Table 4.3: Comparison of multi-stage and single-stage weakly supervised semantic segmentation methods on the PASCAL VOC 2012 validation and test sets (mIoU, %). Here, **I** denotes image-level supervision, **L** denotes language supervision, and **S** denotes saliency supervision. Our method (**UniCL-AffSeg**) employs Swin-B as the backbone or image encoder. Without description, all have used Dense CRF during inference.

Approach	**Backbone**	**Sup.**	**val**	**test**
multi-stage weakly supervised approaches				
$\text{RCA}_{\text{CVPR'22}}$ [68]	ResNet101	I+S	72.2	72.8
$\text{L2G}_{\text{CVPR'22}}$ [21]	ResNet101	I+S	72.1	71.7
$\text{Mat-label}_{\text{ICCV'23}}$ [50]	ResNet101	I+S	73.3	**74.0**
$\text{S-BCE}_{\text{ECCV'22}}$ [52]	ResNet38	I+S	68.1	70.4
$\text{RIB}_{\text{NeurIPS'21}}$ [27]	ResNet38	I	68.3	68.6
$\text{W-OoD}_{\text{CVPR'22}}$ [28]	ResNet101	I	69.8	69.9
$\text{ESOL}_{\text{NeurIPS'22}}$ [29]	ResNet101	I	69.9	69.3
$\text{VML}_{\text{IJCV'22}}$ [41]	ResNet101	I	70.6	70.7
$\text{AETF}_{\text{ECCV'22}}$ [59]	ResNet38	I	70.9	71.7
$\text{MCTformer}_{\text{CVPR'22}}$ [56]	ViT+Res38	I	70.4	70.0
$\text{CDL}_{\text{IJCV'23}}$ [62]	ResNet101	I	72.4	72.2
$\text{ACR}_{\text{CVPR'23}}$ [47]	ViT	I	72.4	72.4
$\text{BECO}_{\text{CVPR'23}}$ [39]	MIT-B2	I	73.7	73.5
$\text{FPR}_{\text{ICCV'23}}$ [12]	ResNet101	I	70.0	70.6
$\text{USAGE}_{\text{ICCV'23}}$ [36]	ResNet38	I	71.9	72.8
$\text{CLIMS}_{\text{CVPR'22}}$ [55]	ViT+Res101	I+L	70.4	70.0
$\text{CLIP-ES}_{\text{CVPR'23}}$ [31]	ViT+Res101	I+L	**73.8**	73.9
single-stage weakly supervised approaches				
$\text{1Stage}_{\text{CVPR'20}}$ [5]	ResNet38	I	62.7	64.3
$\text{RRM}_{\text{AAAI'20}}$ [60]	ResNet38	I	62.6	62.9
$\text{AA\&AR}_{\text{ACMMM'21}}$ [63]	ResNet38	I	63.9	64.8
$\text{SLRNet}_{\text{IJCV'22}}$ [34]	ResNet38	I	67.2	67.6
$\text{AFA}_{\text{CVPR'22}}$ [42]	MIT-B1	I	66.0	66.3
$\text{TSCD}_{\text{AAAI'23}}$ [57]	MIT-B1	I	67.0	67.5
$\text{ToCo}_{\text{CVPR'23}}$ [43]	ViT	I	71.1	72.2
WeCLIP (w/o CRF)	ViT	I+L	74.9	75.2
WeCLIP (w/ CRF)	ViT	I+L	**76.4**	**77.2**
Ours - UniCL-AffSeg(w/oCRF)	Swin-B	I+L	50.0	50.0

pretraining dataset biases to improve performance on underrepresented and small-object categories.

4.2.2 Comparison with Exisiting Methods

Table 4.3 compares our proposed approach (UniCL) with both multi-stage and single-stage weakly supervised semantic segmentation (WSSS) methods. While state-of-the-art single-stage approaches such as WeCLIP achieve strong performance (76.4%/77.2% with CRF post-processing), our current UniCL-AffSeg implementation yields relatively modest results (50.0% on both validation and test sets).

It is important to note that UniCL-AffSeg follows a pipeline similar to WeCLIP, but with a critical distinction in how localization cues are obtained. Whereas WeCLIP relies on attention maps derived from a ViT backbone, UniCL-AffSeg leverages affinity maps computed from the Swin Transformer image encoder in combination with a ViT text encoder. Due to the window-based self-attention mechanism in Swin, the resulting feature representations tend to be more *local* and less globally coherent than ViT attention maps. Consequently, the affinity maps fail to capture holistic object-level relationships, leading to weaker class activation maps (CAMs) at initialization. These initial CAMs often exhibit fragmented activations, particularly for large or complex objects, which limits the effectiveness of subsequent refinement stages.

Furthermore, the feature maps extracted from the Swin image encoder, while rich in local details, lack the global semantic consistency necessary for reliable region affinity learning. This discrepancy explains why UniCL-AffSeg underperforms compared to WeCLIP, despite both methods operating within a broadly similar weakly supervised framework.

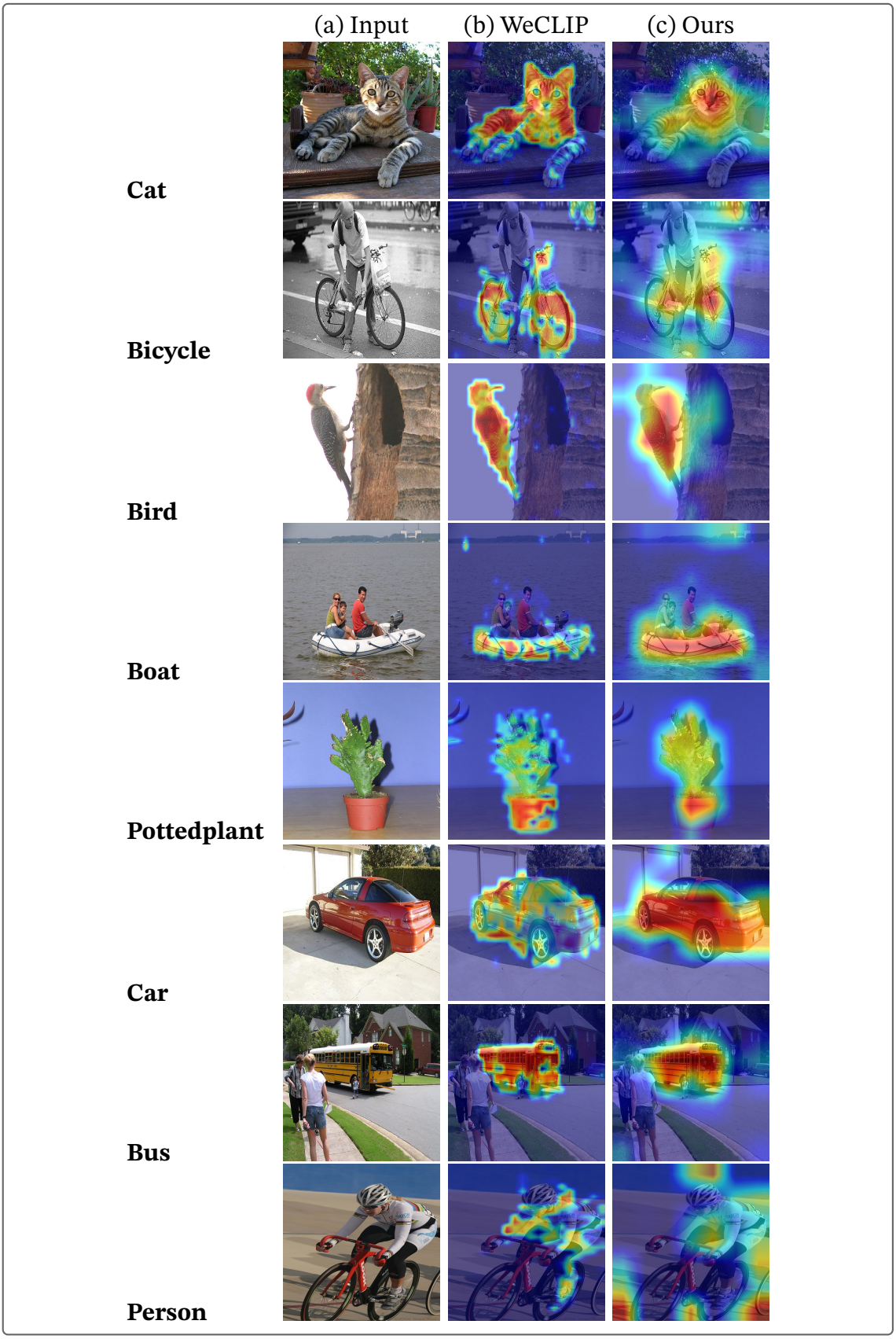


Figure 4.2: Qualitative comparison of CAMs between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *val* set.

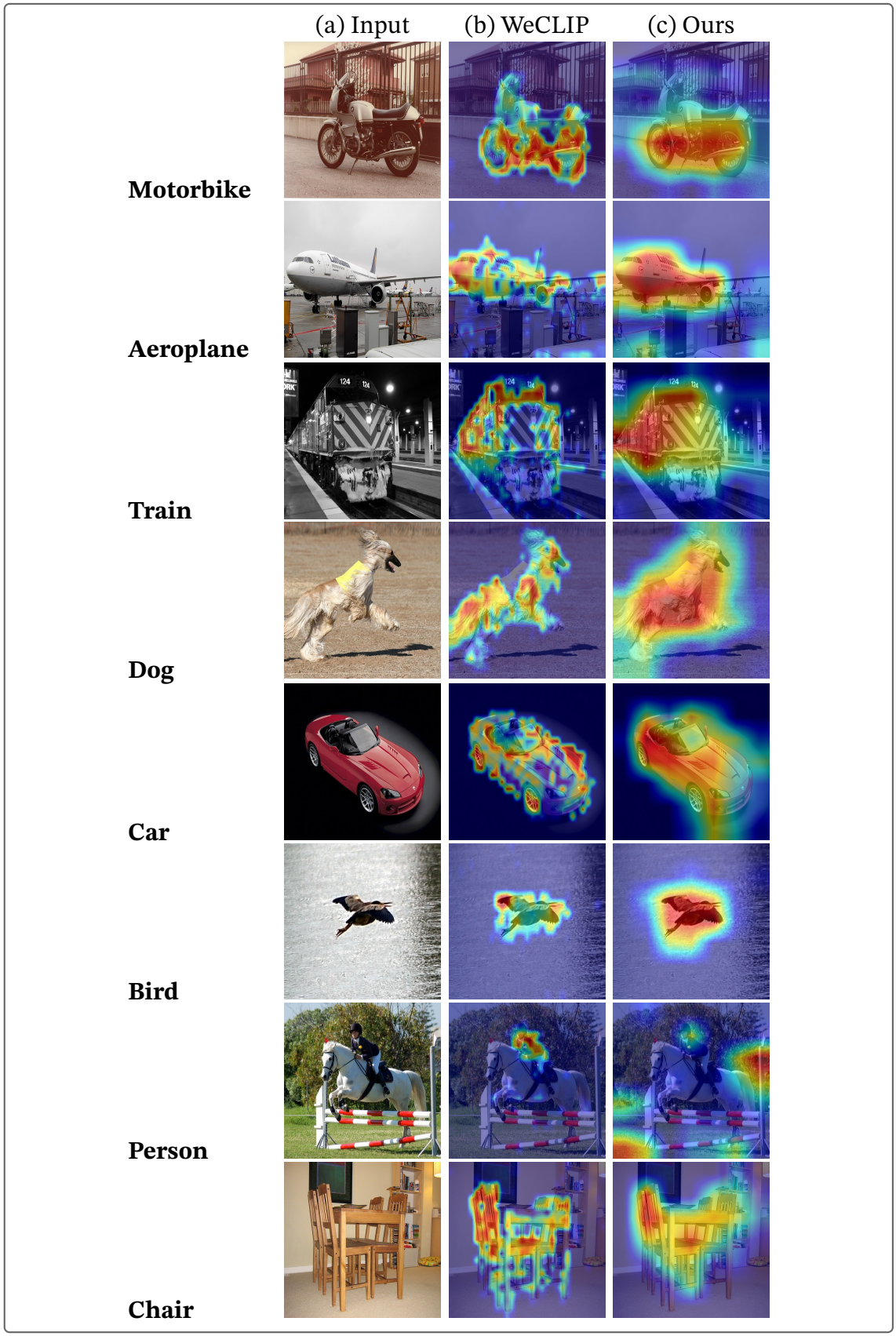


Figure 4.3: Qualitative comparison of CAMs between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *test* set.

4.2.3 Summary of Quantitative Analysis

Overall, the quantitative results reveal that our UniCL-AffSeg framework demonstrates reasonable segmentation ability under image- and language-level supervision, despite operating without pixel-level annotations. The model achieves comparable performance on both validation (50.3% mIoU) and test (50.8% mIoU) sets, indicating good generalization and stable behavior across unseen data.

A closer look at per-class IoUs shows that UniCL-AffSeg performs strongly on large, visually distinct categories such as *bus*, *cow*, and *sheep*, while struggling on small or underrepresented ones like *person*, *chair*, and *bicycle*. This disparity highlights the limitations of Swin-based affinity representations in capturing fine-grained or globally contextual features. When compared with state-of-the-art weakly supervised segmentation models, our approach trails methods like WeCLIP and ToCo, which rely on ViT-based attention mechanisms. The relatively lower mIoU suggests that while Swin Transformers provide strong local representations, their limited global context impedes effective region propagation during pseudo-label generation.

These findings collectively emphasize the need for integrating global reasoning components—such as cross-window attention or hybrid ViT-Swin architectures—to bridge the gap with ViT-based WSSS models and improve small-object segmentation consistency.

4.3 Qualitative Analysis

We present qualitative comparisons of our UniCL-AffSeg framework against the WeCLIP baseline and ground truth annotations on selected Pascal VOC 2012 images. Both class activation maps (CAMs) and pseudo-labels are visualized to highlight differences in object localization, boundary delineation, and overall mask quality.

Beyond side-by-side visuals, we analyze typical strengths and failure modes across diverse scenes: single-object vs multi-object, textured vs uniform backgrounds, small/occluded instances, and fine-structure categories (e.g., bicycle spokes, limbs). We reflect specifically on our outputs to identify where hierarchical features and affinity-based refinement help (improved coverage, cleaner boundaries) and where they struggle (background confusion in cluttered regions, very small objects). These observations complement the quantitative metrics by illustrating the practical behavior of the model under varied conditions.

4.3.1 Observations

Class Activation Maps (CAMs)

Comparison of UniCL-AffSeg CAMs with WeCLIP reveals several key differences:

- **Object Coverage and Completeness**: UniCL-AffSeg CAMs exhibit more contiguous and complete activation regions for many classes. For instance, in images such as *boat*, *aeroplane*, *train*, and *car* (see Figure 4.2 and Figure 4.3), our CAMs capture larger, more holistic object areas, reducing sparsity and better approximating full object extents.
- **Background Noise and Boundaries**: Our CAMs sometimes include extraneous background activations, producing noisy and diffuse maps, e.g., for *cat*, *bird*, and *bicycle* in the first rows of Figure 4.2. In contrast, WeCLIP CAMs show cleaner, more focused activations with sharper edges, minimizing false positives.
- **Handling of Complex or Small Objects**: For challenging cases such as *chair*, *potted plant*, and *person* in cluttered backgrounds, UniCL-AffSeg CAMs occasionally produce false positives or fragmented activations, as seen in the last rows of Figure 4.2 and Figure 4.3. WeCLIP CAMs, while less complete, tend to avoid such errors and maintain higher precision.
- **Consistency Across Datasets**: These trends hold across both the *val* and *test* sets, indicating that the differences in CAM quality are consistent and not dataset-specific.

In summary for CAMs, our UniCL-AffSeg tends to yield fuller object coverage by leveraging hierarchical features from the Swin backbone, which aggregate both local and global context. However, this broader sensitivity can introduce diffuse responses in cluttered areas; thus, the CAM stage benefits from careful thresholding and subsequent refinement to suppress background activations without eroding true positives.

Pseudo-Labels

The pseudo-labels generated by UniCL-AffSeg are compared with those from WeCLIP in Figure 4.5 and Figure 4.6. Key observations include:

- **Fine Detail Capture**: UniCL-AffSeg pseudo-labels more effectively capture intricate object details and shapes. For example, in Figure 4.5 and Figure 4.6, features such as *bicycle spokes*, *object contours*, and other finer structures are

better identified by our model, whereas WeCLIP often misses or oversimplifies them. Notably, some of these fine details are even absent in the ground truth, highlighting our method's ability to recover subtle features.

- **Boundary Delineation**: For classes such as *aeroplane* and *bird*, UniCL-AffSeg produces sharper and more accurate boundaries compared to WeCLIP.
- **Background Activation**: In some cases, our method incorrectly activates background regions and misclassifies them as foreground, for instance in the *person* and *chair* classes shown in Figure 4.7. This behavior is likely caused by noisy CAMs, leading to over-segmentation and false positives.
- **Consistency Across Datasets**: The observed improvements in pseudo-label quality are consistent across both the *val* and *test* sets, similar to the trends seen in the CAM analysis.

Taken together, the pseudo-labels produced by our pipeline typically align more closely with object extents and boundaries than the baseline, especially for categories requiring multi-scale context. The remaining errors largely stem from noisy CAM seeds in highly cluttered scenes, class co-occurrence with strong context priors (e.g., person–chair), and extreme scale variation. Post-processing choices (e.g., affinity strength, smoothing, and confidence thresholds) can shift the balance between over- and under-segmentation.

4.3.2 Summary of Qualitative Analysis

Overall, our qualitative study indicates the following:

- **Strengths**: Improved object coverage and continuity relative to the baseline; sharper boundaries after refinement; better separation in multi-object scenes; and resilience to moderate viewpoint/lighting changes.
- **Limitations**: Occasional background leakage in cluttered regions; under-coverage for very small or heavily occluded objects; and sensitivity to threshold/refinement hyperparameters in edge cases.
- **Implications**: High-quality pseudo-labels are attainable from image-level supervision when hierarchical features and affinity propagation are combined. Robustness further improves with adaptive thresholding and category-aware refinement.
- **Opportunities**: Future work could incorporate uncertainty-aware masks,

dynamic refinement strength per category/scene, and lightweight boundary-aware modules to further reduce leakage and missed fine structures.

4.4 Ablation Study

To evaluate the contribution of the Refinement Module (RFM) in our UniCL-AffSeg framework, we conducted an ablation study comparing the segmentation performance with and without the RFM. This analysis allows us to quantify how the RFM improves the initial Class Activation Maps (CAMs) generated by the Swin Transformer backbone.

Table 4.4: Per-Class IoU Comparison on PASCAL VOC: Without vs With RFM

Class	**Without RFM**	**With RFM**
background	0.796	0.782
aeroplane	0.606	0.566
bicycle	0.243	0.330
bird	0.565	0.610
boat	0.414	0.431
bottle	0.391	0.382
bus	0.639	0.678
car	0.417	0.495
cat	0.525	0.652
chair	0.242	0.261
cow	0.574	0.595
diningtable	0.319	0.420
dog	0.563	0.660
horse	0.562	0.609
motorbike	0.535	0.576
person	0.219	0.169
potted plant	0.283	0.287
sheep	0.596	0.620
sofa	0.373	0.465
train	0.534	0.558
tv/monitor	0.451	0.419
Mean IoU (mIoU)	0.469	0.503

4.4.1 Quantitative Study

Table 4.4 presents the per-class IoU and overall mean IoU (mIoU) for the PASCAL VOC dataset, comparing the baseline model without the Refinement Module (RFM) against the model with RFM applied. The inclusion of the RFM leads to a clear improvement in segmentation quality, increasing the overall mIoU from 0.469 to

0.503. This enhancement demonstrates the effectiveness of the module in refining spatial coherence and object completeness. Notably, classes that are large or visually distinctive, such as *bus*, *cat*, *dog*, and *sheep*, exhibit substantial gains in IoU, confirming the RFM's ability to enhance region coverage for well-represented categories. Moreover, smaller or structurally complex classes, including *bicycle*, *diningtable*, and *sofa*, also show improvement, suggesting that the RFM effectively sharpens boundaries and captures finer object details. However, the *person* class experiences a slight decline in IoU, likely due to inherent bias in the UniCL pretraining datasets (ImageNet-21K and YFCC-14M), where human-centric categories are underrepresented. Consequently, the RFM's refinement process becomes less effective for such classes, underscoring the importance of addressing pretraining bias in future work.

4.4.2 Qualitative Study

Figure 4.4 demonstrates the qualitative effect of the Refinement Module (RFM) on pseudo-label generation across diverse object categories. Without RFM, CAMs frequently appear fragmented and incomplete, particularly along fine structures such as bird wings or bicycle wheels. Incorporating RFM results in smoother, spatially coherent masks with improved object coverage and sharper boundaries, effectively reducing under-segmentation and background noise (e.g., in the airplane and dog examples).

However, in some cases—most notably for the person class—the initial CAMs exhibit high confidence in background regions, which RFM further propagates during affinity-based refinement. This leads to over-segmentation and explains the slight quantitative drop observed for the person category in Table 4.4. Despite this limitation, RFM generally enhances semantic completeness and boundary precision, validating its role in improving spatial consistency under weak supervision.

Overall, the ablation study confirms that the RFM effectively enhances segmentation quality by refining initial CAMs using affinity information and contextual cues, particularly for classes that are underrepresented or have complex shapes. These results establish a strong baseline for subsequent experiments and highlight the importance of the RFM in improving both boundary delineation and class-specific performance.

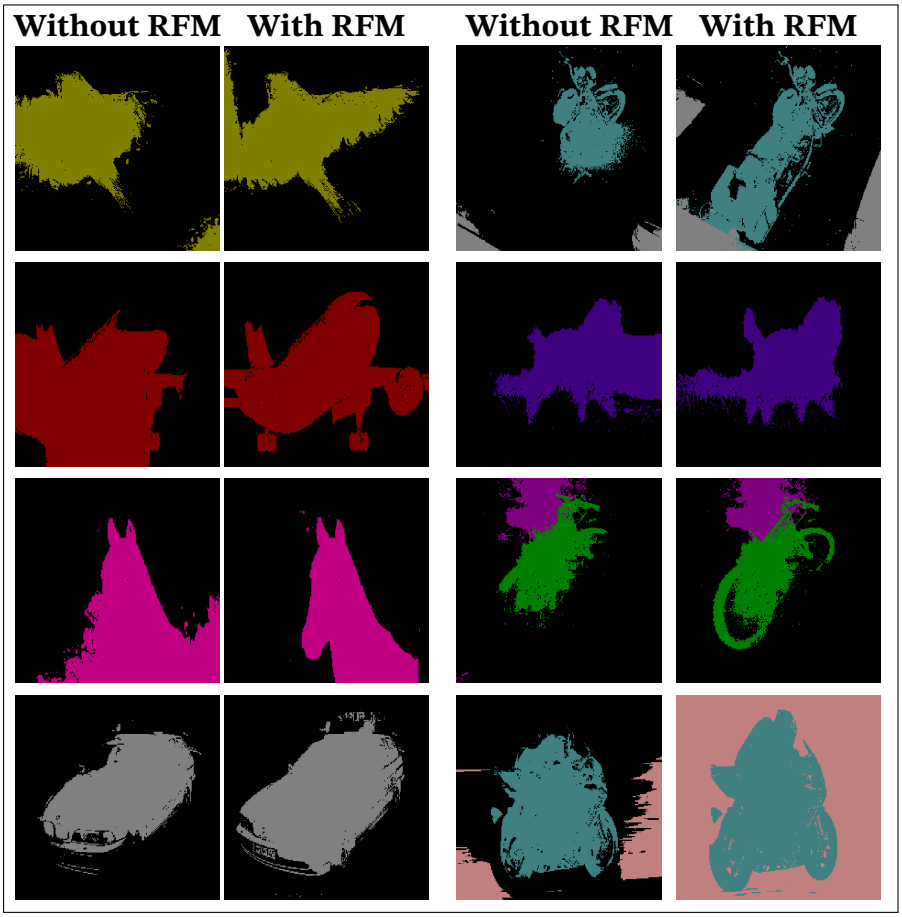


Figure 4.4: Qualitative ablation on the Pascal VOC dataset showing the effect of the Refinement Module (RFM) [61] using our method. Each pair compares pseudo-labels generated without (left) and with (right) RFM. The RFM enhances boundary precision and recovers missing object regions, leading to more complete and accurate segmentation masks.

4.5 Interpretation of Results

The quantitative and qualitative analyses together provide a comprehensive understanding of the strengths and limitations of our weakly supervised segmentation framework.

From the quantitative perspective, the model achieves a mean IoU of 50.3% on the validation set and 50.8% on the test set, indicating strong generalization despite using only weak supervision. The per-class IoU breakdown (Table 4.1) reveals that large and distinctive object categories—such as **bus**, **sheep**, **horse**, and **cat**—are segmented with high accuracy, while classes that are small, highly deformable, or poorly represented during pretraining (e.g., **person**, **chair**, and **potted plant**) show significantly weaker performance. This pattern aligns with the biases inherent in the pretraining datasets (ImageNet-21K and YFCC-14M), which lack dense, diverse coverage for these semantic categories.

Qualitative inspection further supports these findings. Visual examples show that predictions for animals and vehicles tend to produce coherent and complete masks with clear object boundaries, whereas human instances often exhibit fragmented or overly diffuse activations. In particular, the **person** class consistently fails due to its initial Class Activation Maps (CAMs) being dominated by high confidence in the **background** regions. Since the Refinement Module (RFM) operates via random walk propagation, these early misaligned activations are spread across spatially adjacent pixels, reinforcing incorrect background predictions. Moreover, the pixel-adaptive module, while effective in smoothing object interiors, tends to over-suppress fine structural details in thin or articulated objects such as limbs, further degrading segmentation quality for humans.

Overall, the results indicate that our model is highly effective in identifying large, contextually distinct objects but remains sensitive to CAM initialization errors and class imbalance during pretraining. The analysis suggests that improving CAM reliability—particularly for human-centric and small-object categories—through better initialization, adaptive refinement control, or scale-aware feature fusion could significantly enhance overall segmentation quality.

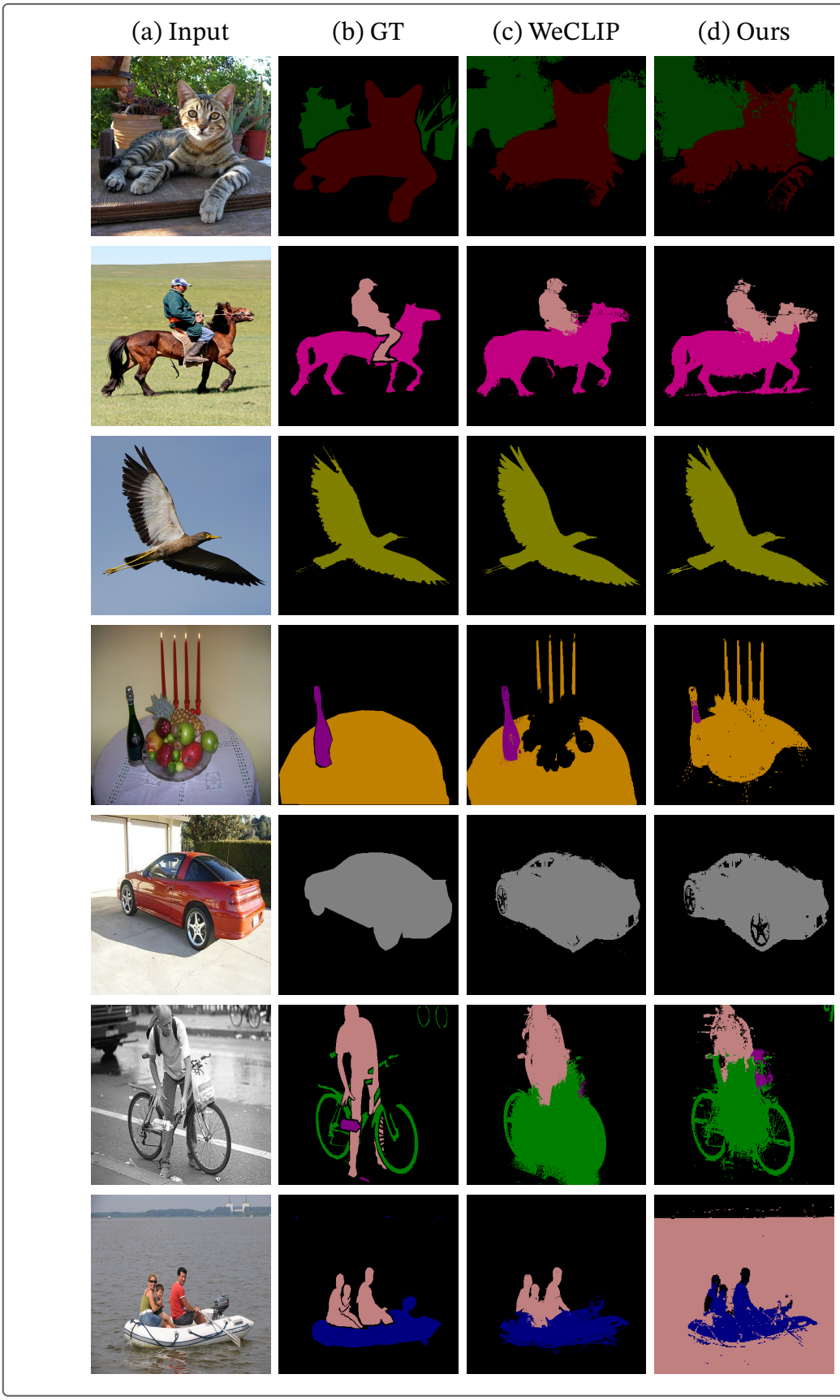


Figure 4.5: Qualitative comparison of pseudo-labels between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *val* set.

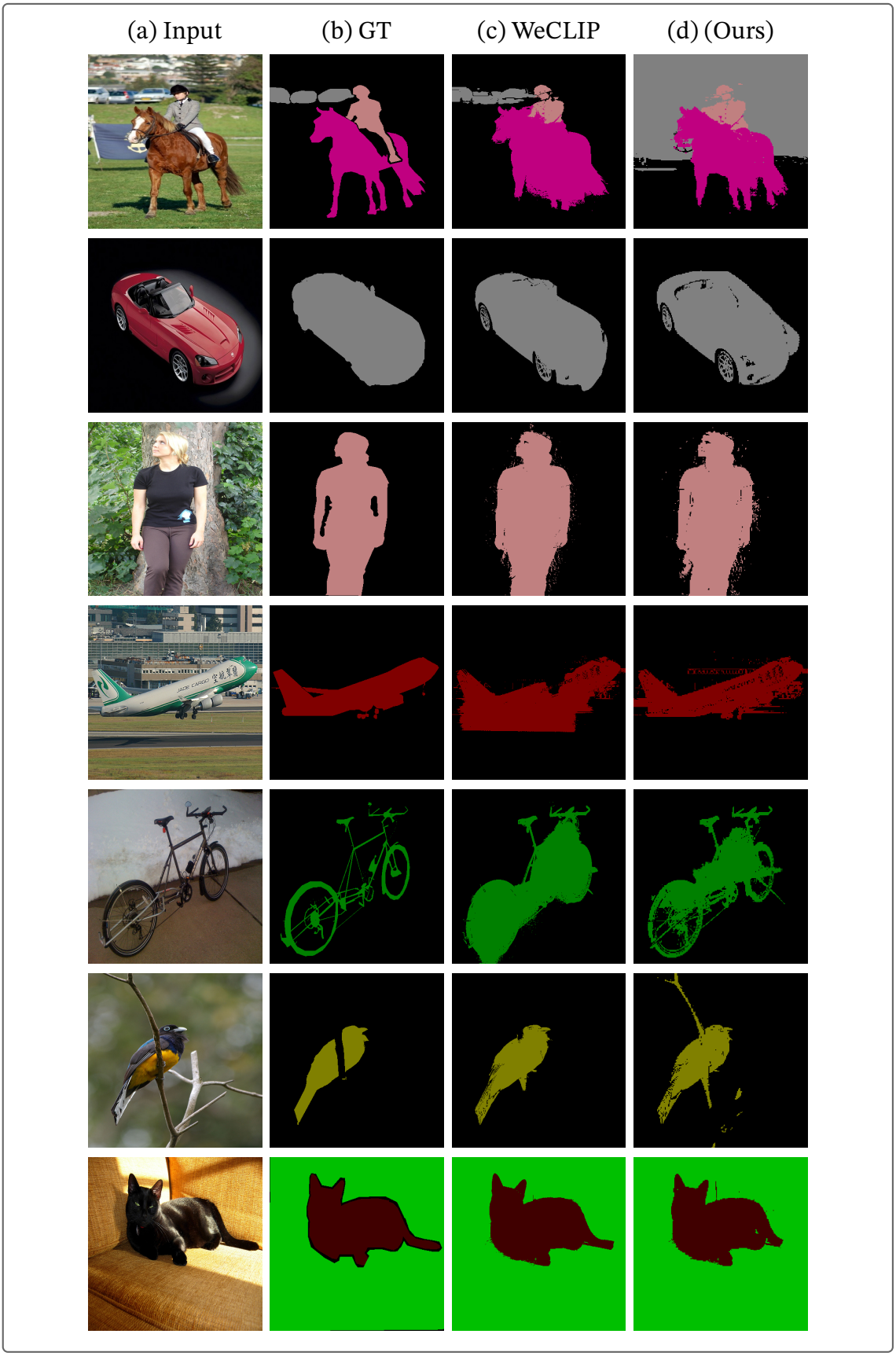


Figure 4.6: Qualitative comparison of pseudo-labels between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *test* set.

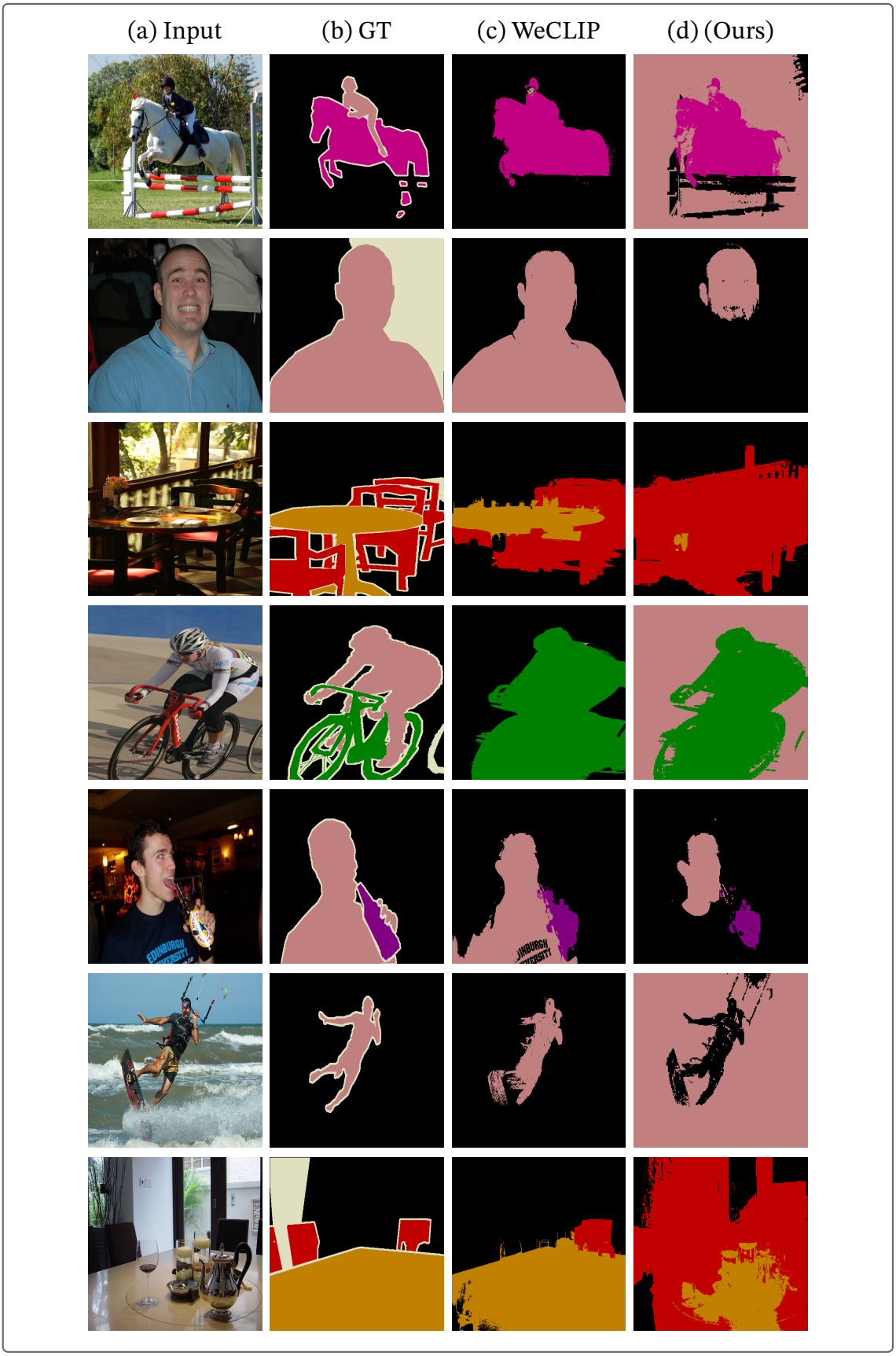


Figure 4.7: Qualitative comparison of pseudo-labels between WeCLIP and our UniCL-AffSeg on PASCAL VOC 2012 *val* and *test* set for person and chair classes.

4.6 Strengths of the Approach

The proposed UniCL-AffSeg framework demonstrates several strengths that collectively advance the state of weakly supervised semantic segmentation (WSSS). By combining multi-modal representation learning with hierarchical transformer-based feature extraction and affinity-driven refinement, UniCL-AffSeg effectively enhances both localization accuracy and boundary quality in segmentation masks. The following subsections outline these major strengths.

4.6.1 Multi-Modal Learning and Unified Objectives

Leveraging UniCL's unified contrastive and classification objectives, the framework aligns image and text representations in a shared embedding space. This integration enriches the semantic understanding of visual features without requiring pixel-level supervision. In practice, this results in more discriminative and spatially coherent Class Activation Maps (CAMs), as reflected in the model's improved localization performance on large and visually distinctive categories such as *bus*, *sheep*, and *horse*. The cross-modal synergy also enhances generalization across diverse scenes, reducing dependency on external localization cues like saliency maps.

4.6.2 Hierarchical Feature Extraction with Swin Transformer

The Swin Transformer backbone provides hierarchical multi-scale features that capture both local and global context—an essential quality for dense prediction tasks. Its shifted-window attention mechanism preserves fine-grained boundary details while maintaining computational efficiency. This hierarchical representation allows UniCL-AffSeg to delineate object boundaries more precisely and maintain structural continuity, particularly in multi-object scenes where traditional CAMs often suffer from fragmentation or missing parts.

4.6.3 Affinity-Based Refinement for Semantic Consistency

The affinity-based refinement module significantly enhances semantic consistency in pseudo-labels. By integrating encoder- and decoder-level affinity maps and filtering them through deviation scoring, the framework propagates relevant activations across object regions while suppressing background noise. This results in smoother, more complete segmentation masks with improved intra-object coherence. The approach effectively reduces over-segmentation and false activations

in cluttered scenes, contributing to the framework's robustness under purely weak supervision.

4.6.4 Efficiency and Scalability

Despite incorporating multi-modal and transformer-based components, UniCL-AffSeg remains computationally efficient. Swin's linear-complexity attention and UniCL's pre-trained representations enable scalable training on large datasets like PASCAL VOC 2012. Moreover, the Pixel-Adaptive Refinement (PAR) module introduces only minimal overhead while significantly improving boundary sharpness through adaptive color and spatial filtering. Together, these design choices ensure a favorable balance between segmentation accuracy and computational cost, making the framework suitable for broader real-world deployment.

4.6.5 Overall Impact

These strengths collectively enable UniCL-AffSeg to produce high-quality CAMs and pseudo-labels that bridge much of the gap between weakly supervised and fully supervised methods. The framework demonstrates that hierarchical transformers and multi-modal pretraining, when combined with affinity-based refinement, can deliver scalable and annotation-efficient segmentation without reliance on dense supervision.

4.7 Limitations and Challenges

Despite its conceptual alignment with recent single-stage approaches such as WeCLIP, our UniCL-AffSeg framework reveals several limitations that help explain its underperformance in practice. These challenges can be grouped into architectural constraints, pseudo-label generation issues, and broader training difficulties.

4.7.1 Architectural Constraints

Locality of Swin Attention. The Swin Transformer employs a window-based self-attention mechanism, which provides strong local feature representations but lacks a truly global receptive field. While beneficial for capturing fine-grained details, this design prevents the model from reasoning holistically about objects spread across distant regions. Consequently, the affinity maps produced by Swin

tend to be locally accurate but globally fragmented, limiting UniCL's ability to propagate semantic information across the entire object. By contrast, ViT backbones in CLIP capture long-range dependencies more naturally through global self-attention, yielding more coherent semantic maps.

Feature-Text Modality Mismatch. Although UniCL was designed to unify contrastive and classification-based objectives, its integration with Swin features exposes a modality gap. Swin feature maps are hierarchical and locally focused, whereas the ViT-based text encoder provides globally contextualized embeddings. This structural mismatch weakens image-text correspondence, which is particularly detrimental in weakly supervised settings where no pixel-level annotations exist to bridge the gap. The result is unreliable affinity learning and degraded segmentation quality.

Lack of Multi-Scale Feature Fusion. State-of-the-art WSSS methods often leverage multi-scale representations to capture both small details and large object contexts. While Swin naturally produces hierarchical feature maps at different resolutions, our current framework processes these scales in isolation rather than integrating them. This omission leads to two shortcomings: (i) small or thin object parts are often overlooked due to insufficient fine-scale emphasis, and (ii) large objects are inconsistently segmented because their spatially distributed regions cannot be jointly reasoned about. In contrast, ViT-based approaches such as WeCLIP benefit from globally uniform attention maps that inherently integrate multiple scales of context.

4.7.2 Pseudo-Label Generation Limitations

Weak and Inconsistent CAMs. The quality of the initial class activation maps (CAMs) is critical for downstream segmentation. In our framework, CAMs generated from Swin features tend to be weaker and more spatially inconsistent than those derived from ViT backbones. Unlike ViT attention maps, which implicitly encode global semantic cues, Swin affinity maps are more sensitive to local noise and lack interpretability. This often results in excessive false positives and incomplete object coverage, constraining the effectiveness of refinement steps.

Foreground-Background Separation. A persistent challenge in WSSS is distinguishing objects from background clutter. Since our pseudo-labels rely heavily on weak CAMs and affinity propagation, background regions are frequently

mislabeled as foreground, while parts of true objects remain unactivated. Although post-processing methods such as CRFs can partially alleviate this issue, they cannot fully compensate for noisy or incomplete initial cues.

4.7.3 Practical Training Challenges

Computational Limitations. Due to limited GPU memory, our framework requires gradient accumulation to achieve reasonable batch sizes. This increases training time and reduces the stability of optimization, as small effective batch sizes introduce noisy gradient estimates. The additional overhead slows down experimentation, making hyperparameter tuning less tractable.

Sensitivity to Hyperparameters. The UniCL-AffSeg framework exhibits high sensitivity to training schedules and refinement parameters. Factors such as learning rate decay, number of iterations, and CAM post-processing settings substantially influence final performance. This instability complicates adaptation to new datasets or domains and makes reproducibility more difficult compared to more stable ViT-based methods.

4.8 Key Findings

The analysis of UniCL-AffSeg in the context of weakly supervised semantic segmentation reveals several key findings, which are organized below for clarity:

1. Local vs. Global Feature Representation

- **Strength of Swin-based UniCL encoders:** These encoders provide robust local feature representations.
- **Limitation:** The windowed attention mechanism restricts global semantic reasoning, which is essential for generating coherent class activation maps (CAMs).
- **Implication:** Future WSSS frameworks may benefit from hybrid architectures that combine local affinity-based features with global attention cues, enabling more accurate propagation of semantic information across object regions.

2. Affinity Maps and Multi-Scale Feature Fusion

- **Affinity maps as an alternative:** Using affinity maps instead of attention maps is effective for capturing pixel-level relationships, especially with hierarchical Swin features.
- **Current limitation:** The approach does not fully leverage multi-scale feature fusion due to differing spatial resolutions of stage-wise feature maps.
- **Future directions:** Techniques such as upsampling, feature alignment, or transformer-based fusion modules could be explored to integrate multi-scale information, improving detection of small structures and segmentation coherence for large objects.

3. Quality of Initial CAMs

- **Bottleneck:** The quality of initial CAMs remains a limiting factor for downstream segmentation performance.
- **Potential improvements:** Incorporating stronger initialization strategies (e.g., contrastive pretraining on object parts) or leveraging pseudo-label refinement techniques could enhance initial activations and lead to more accurate final masks.

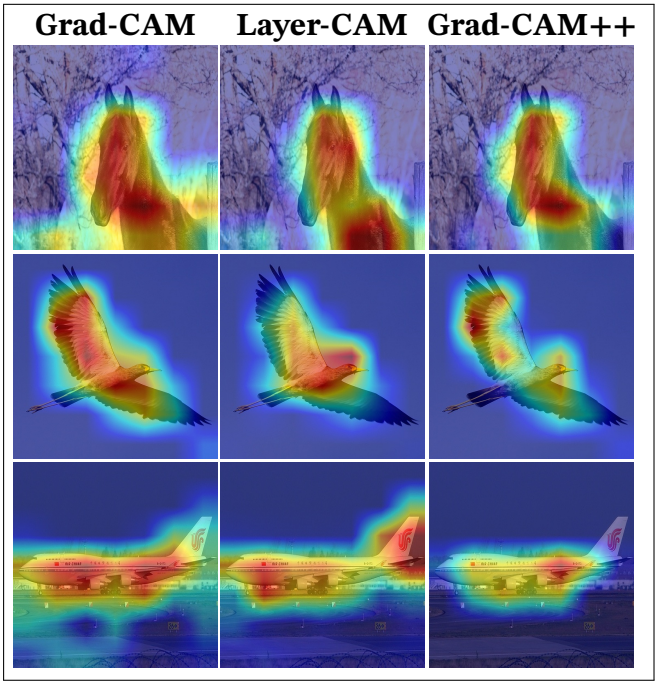


Figure 4.8: Comparison of CAM visualization methods on the Pascal VOC dataset for single class scenario. Columns show Grad-CAM, Layer-CAM, and Grad-CAM++ respectively. Each row corresponds to a different example image.

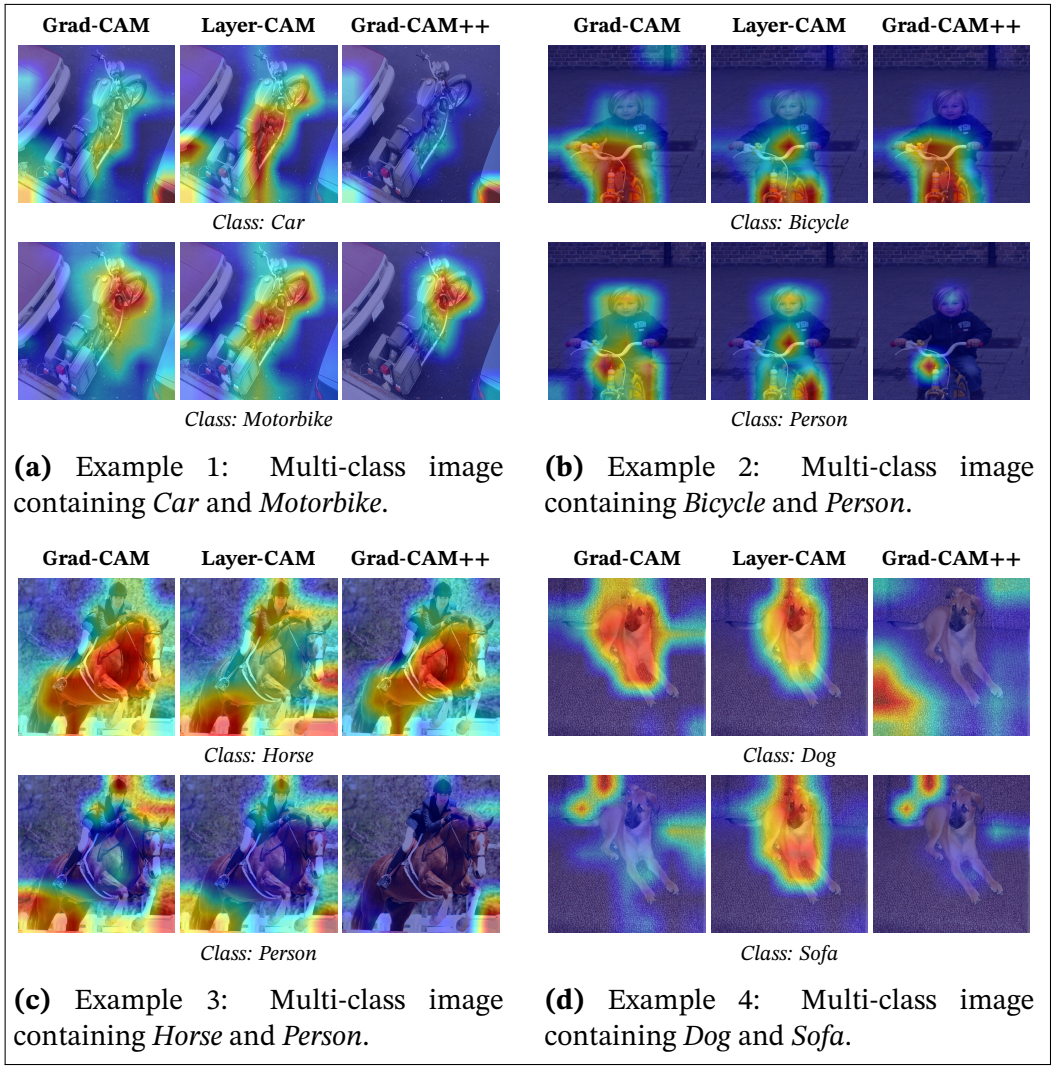
4. Broader Implications for WSSS Pipeline Design

- **Contrastive learning frameworks:** The results highlight the potential of frameworks like UniCL in weakly supervised settings.
- **Balance needed:** There is a need to balance local detail and global context for optimal performance.
- **General framework:** Addressing these challenges can improve segmentation performance and provide a more general approach for integrating vision-language pretraining with weak supervision in diverse visual recognition tasks.

4.9 Other Explored Approaches

During development, we investigated several alternative strategies that, while promising in theory, did not yield satisfactory results in practice. We summarize these negative results here both for completeness and to highlight potential pitfalls for future work.

Alternative CAM Generation Methods. We experimented with LayerCAM and Grad-CAM++, as replacements for our primary CAM extraction mechanism. Although these methods are widely used in weakly supervised settings, in our framework they couldn't produce better CAMs.
LayerCAM, which uses the element-wise product of feature maps and gradients, produced better CAMs than Grad-CAM in case of single class images, shown in Figure 4.8. However, in multi-class scenarios, it almost always highlighted only one dominant region for every class, which is quite interesting. This problem is shown in Figure 4.9(a), Figure 4.9(c) and Figure 4.9(d).
Grad-CAM++ weights gradients based on their positive influence. It often missed easier classes, such as the *Dog* class shown in Figure 4.9(d). It also failed to localize the *Person* class in multi-class images, as shown in Figure 4.9(b) and Figure 4.9(c). Overall, Grad-CAM++ produced more fragmented and less false positives compared to Grad-CAM.
Overall, Grad-CAM provided the most consistent and complete CAMs across both single and multi-class images, making it the most suitable choice for our framework. Figure 4.9 shows the comparison of all three methods on multiple multi-class images.



(a) Example 1: Multi-class image containing *Car* and *Motorbike*.

(b) Example 2: Multi-class image containing *Bicycle* and *Person*.

(c) Example 3: Multi-class image containing *Horse* and *Person*.

(d) Example 4: Multi-class image containing *Dog* and *Sofa*.

Figure 4.9: Comparison of Grad-CAM, Layer-CAM, and Grad-CAM++ visualizations for multiple multi-class images from the Pascal VOC dataset. Each subfigure shows CAMs for two distinct object classes within the same image.

Backbone Fine-tuning Variants. We also attempted to fine-tune the last one or two Swin Transformer blocks, both with and without an additional fully connected (FC) projection layer. These modifications degraded the quality of the generated CAMs, often resulting in unstable or inconsistent activations. This suggests that partial fine-tuning, without carefully designed regularization, may overfit to local patterns and disrupt the alignment with text embeddings.

Local-to-Global Attention Conversion. Finally, we explored replacing Swin's window-based local attention with a global attention mechanism to improve long-range reasoning. However, this approach led to a sparse and ineffective global attention matrix, which in turn degraded segmentation quality. This outcome underscores the difficulty of naively extending Swin to global attention without

appropriate architectural adaptations.

In summary, although these explorations did not improve performance, they provide valuable insights. They demonstrate the limitations of directly adopting generic CAM methods, highlight the fragility of partial backbone fine-tuning, and reveal the challenges of balancing local and global attention in window-based Transformers.

Chapter 5

Conclusion

This study presented UniCL-AffSeg, a weakly supervised semantic segmentation framework that integrates multi-modal contrastive learning with hierarchical transformer architectures to produce spatially coherent and semantically meaningful Class Activation Maps (CAMs). By leveraging UniCL's unified vision-language representations and the Swin Transformer's hierarchical design, the proposed approach enhances pseudo-label quality through affinity-based and pixel-adaptive refinement strategies.

The experimental analyses demonstrate that UniCL-AffSeg performs competitively under purely weak supervision, effectively capturing large and visually distinctive objects while revealing persistent challenges in segmenting small or fine-structured categories. These outcomes confirm the benefit of combining multi-modal pretraining with affinity-driven refinement, offering a scalable and annotation-efficient alternative to fully supervised methods. Beyond quantitative performance, the findings underscore the importance of improving CAM generation and addressing pretraining biases to further close the gap between weakly and fully supervised segmentation. The framework illustrates how vision-language alignment can guide more structured pixel-level understanding from limited supervision.

The following section outlines potential directions for future exploration—such as incorporating hybrid local-global reasoning, adaptive prompt learning, and alternative transformer backbones—which together provide a roadmap for advancing weakly supervised and multi-modal semantic segmentation research.

Future Work

So far, we have explored various models, architectures, and methods for refining CAMs, pseudo-labels, and segmentation maps; however, there remain several promising directions for future research and improvement. Building on recent advances in weakly supervised semantic segmentation, CLIP, and UniCL, these directions aim to enhance both model performance and generalization, while addressing limitations observed in current approaches.

Future work can focus on experimenting with diverse backbones and decoder architectures, such as ViT, ResNet variants, or Mix Transformer, to improve CAM and segmentation quality. Advanced feature aggregation techniques, including transformer-based fusion modules and multi-scale feature alignment, may enhance the integration of intermediate feature and attention maps. Refining CAMs and pseudo-labels using graph-based, CRF-based, affinity-based, uncertainty-aware, or region-based strategies can further improve semantic coherence. Additionally, exploring alternative segmentation heads, novel objective functions like region-based or contrastive losses, and prompt engineering for multi-modal models like CLIP or UniCL can strengthen learning and alignment, especially for rare or ambiguous classes. Leveraging self-supervised or semi-supervised pretraining and combining local-global reasoning through hybrid architectures, such as Swin Transformer and ViT, may further boost feature representation and semantic propagation. Finally, evaluating models for cross-dataset generalization can improve adaptability across datasets with different distributions or label sets.

Collectively, these future directions provide a comprehensive roadmap for advancing weakly supervised semantic segmentation and multi-modal learning. They highlight opportunities to improve architectural design, feature aggregation, refinement strategies, and model generalization, paving the way for more accurate, robust, and scalable segmentation systems.

This chapter presents all the references cited throughout the thesis. The list is compiled in accordance with the IEEE citation style and includes all works that informed the research, methodology, and discussion.

References

[1] S. Abnar and W. Zuidema, *Quantifying attention flow in transformers*, 2020. arXiv: 2005.00928 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2005.00928.

[2] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994. DOI: 10.1109/34.295913.

[3] J. Ahn, S. Cho, and S. Kwak, *Weakly supervised learning of instance segmentation with inter-pixel relations*, 2019. arXiv: 1904.05044 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1904.05044.

[4] J. Ahn and S. Kwak, *Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation*, 2018. arXiv: 1803.10464 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1803.10464.

[5] N. Araslanov and S. Roth, *Single-stage semantic segmentation from image labels*, 2020. arXiv: 2005.08104 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2005.08104.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[7] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Mar. 2018. DOI: 10.1109/wacv.2018.00097. [Online]. Available: http://dx.doi.org/10.1109/WACV.2018.00097.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

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

[11] 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)*, 2018, pp. 801–818.

[12] L. Chen, C. Lei, R. Li, S. Li, Z. Zhang, and L. Zhang, "Fpr: False positive rectification for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1108–1118.

[13] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. DOI: 10.1109/CVPR.2016.350. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/Cordts_The_Cityscapes_Dataset_CVPR_2016_paper.html.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," Jun. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[15] A. Dosovitskiy et al., *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2010.11929.

[16] Y. Du, Z. Fu, Q. Liu, and Y. Wang, *Weakly supervised semantic segmentation by pixel-to-prototype contrast*, 2022. arXiv: 2110.07110 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2110.07110.

[17] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, Jun. 2010. DOI: 10.1007/s11263-009-0275-4.

[18] W. Gao et al., *Ts-cam: Token semantic coupled attention map for weakly supervised object localization*, 2021. arXiv: 2103.14862 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2103.14862.

[19] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011.

[20] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014–7023. DOI: 10.1109/CVPR.2018.00733.

[21] P.-T. Jiang, Y. Yang, Q. Hou, and Y. Wei, "L2g: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[22] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "Layercam: Exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5875–5888, 2021. DOI: 10.1109/TIP.2021.3089943.

[23] A. Kolesnikov and C. H. Lampert, *Seed, expand and constrain: Three principles for weakly-supervised image segmentation*, 2016. arXiv: 1603.06098 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1603.06098.

[24] A. Kolesnikov and C. H. Lampert, *Seed, expand and constrain: Three principles for weakly-supervised image segmentation*, 2016. arXiv: 1603.06098 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1603.06098.

[25] P. Krähenbühl and V. Koltun, *Efficient inference in fully connected crfs with gaussian edge potentials*, 2012. arXiv: 1210.5644 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1210.5644.

[26] J. R. Lee, S. Kim, I. Park, T. Eo, and D. Hwang, "Relevance-cam: Your model already knows where to look," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 944–14 953.

[27] J. Lee, J. Choi, J. Mok, and S. Yoon, "Reducing information bottleneck for weakly supervised semantic segmentation," *Advances in neural information processing systems*, vol. 34, pp. 27 408–27 421, 2021.

[28] J. Lee, S. J. Oh, S. Yun, J. Choe, E. Kim, and S. Yoon, "Weakly supervised semantic segmentation using out-of-distribution data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 897–16 906.

[29] J. Li, Z. Jie, X. Wang, X. Wei, and L. Ma, "Expansion and shrinkage of localization for weakly-supervised semantic segmentation," *Advances in neural information processing systems*, vol. 35, pp. 16 037–16 051, 2022.

[30] T.-Y. Lin et al., *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1405.0312.

[31] Y. Lin et al., *Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation*, 2023. arXiv: 2212.09506 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2212.09506.

[32] Z. Liu et al., *Swin transformer: Hierarchical vision transformer using shifted windows*, 2021. arXiv: 2103.14030 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2103.14030.

[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[34] J. Pan et al., "Learning self-supervised low-rank network for single-stage weakly and semi-supervised semantic segmentation," *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1181–1195, 2022.

[35] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, *Weakly- and semi-supervised learning of a dcnn for semantic image segmentation*, 2015. arXiv: 1502.02734 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1502.02734.

[36] Z. Peng, G. Wang, L. Xie, D. Jiang, W. Shen, and Q. Tian, "Usage: A unified seed area generation paradigm for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 624–634.

[37] P. O. Pinheiro and R. Collobert, *From image-level to pixel-level labeling with convolutional networks*, 2015. arXiv: 1411.6228 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1411.6228.

[38] A. Radford et al., *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2103.00020.

[39] S. Rong, B. Tu, Z. Wang, and J. Li, "Boundary-enhanced co-training for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 574–19 584.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference,*

Munich, Germany, October 5-9, 2015, proceedings, part III 18, Springer, 2015, pp. 234–241.

[41] L. Ru, B. Du, Y. Zhan, and C. Wu, "Weakly-supervised semantic segmentation with visual words learning and hybrid pooling," *arXiv preprint arXiv:2202.04812*, 2022.

[42] L. Ru, Y. Zhan, B. Yu, and B. Du, *Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers*, 2022. arXiv: 2203.02664 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2203.02664.

[43] L. Ru, H. Zheng, Y. Zhan, and B. Du, *Token contrast for weakly-supervised semantic segmentation*, 2023. arXiv: 2303.01267 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2303.01267.

[44] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Oct. 2019, ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. [Online]. Available: http://dx.doi.org/10.1007/s11263-019-01228-7.

[45] R. Sinkhorn, "A relationship between arbitrary positive matrices and stochastic matrices," *Canadian Journal of Mathematics*, vol. 18, pp. 303–306, 1966. [Online]. Available: https://api.semanticscholar.org/CorpusID:123663969.

[46] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, *Segmenter: Transformer for semantic segmentation*, 2021. arXiv: 2105.05633 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2105.05633.

[47] G. Sun, W. Wang, J. Dai, and L. Van Gool, "Weakly supervised semantic segmentation with generative attention spread and region refinement," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 11 329–11 339.

[48] W. Sun, J. Zhang, Z. Liu, Y. Zhong, and N. Barnes, *Getam: Gradient-weighted element-wise transformer attention map for weakly-supervised semantic segmentation*, 2022. arXiv: 2112.02841 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2112.02841.

[49] P. Tokmakov, K. Alahari, and C. Schmid, *Weakly-supervised semantic segmentation using motion cues*, 2017. arXiv: 1603.07188 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1603.07188.

[50] C. Wang, R. Xu, S. Xu, W. Meng, and X. Zhang, "Treating pseudo-labels generation as image matting for weakly supervised semantic segmentation," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 755–765. DOI: 10.1109/ICCV51070.2023.00076.

[51] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, *Object region mining with adversarial erasing: A simple classification to semantic segmentation approach*, 2018. arXiv: 1703.08448 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1703.08448.

[52] T. Wu, G. Gao, J. Huang, X. Wei, X. Wei, and C. H. Liu, "Adaptive spatial-bce loss for weakly supervised semantic segmentation," in *European Conference on Computer Vision*, Springer, 2022, pp. 199–216.

[53] Z. Wu, C. Shen, and A. van den Hengel, *Wider or deeper: Revisiting the resnet model for visual recognition*, 2016. arXiv: 1611.10080 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1611.10080.

[54] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in neural information processing systems*, vol. 34, pp. 12 077–12 090, 2021.

[55] J. Xie, X. Hou, K. Ye, and L. Shen, "Clims: Cross language image matching for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4483–4492.

[56] L. Xu, W. Ouyang, M. Bennamoun, F. Boussaid, and D. Xu, "Multi-class token transformer for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4310–4319.

[57] R. Xu, C. Wang, J. Sun, S. Xu, W. Meng, and X. Zhang, "Self correspondence distillation for end-to-end weakly-supervised semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 3045–3053.

[58] J. Yang et al., *Unified contrastive learning in image-text-label space*, 2022. arXiv: 2204.03610 [cs.CV].

[59] S.-H. Yoon, H. Kweon, J. Cho, S. Kim, and K.-J. Yoon, "Adversarial erasing framework via triplet with gated pyramid pooling layer for weakly supervised semantic segmentation," in *European conference on computer vision*, Springer, 2022, pp. 326–344.

[60] B. Zhang, J. Xiao, Y. Wei, M. Sun, and K. Huang, *Reliability does matter: An end-to-end weakly supervised semantic segmentation approach*, 2019. arXiv: 1911.08039 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1911.08039.

[61] B. Zhang, S. Yu, Y. Wei, Y. Zhao, and J. Xiao, *Frozen clip: A strong backbone for weakly supervised semantic segmentation*, 2024. arXiv: 2406.11189 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2406.11189.

[62] F. Zhang, C. Gu, C. Zhang, and Y. Dai, "Complementary patch for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 7242–7251.

[63] X. Zhang et al., *Adaptive affinity loss and erroneous pseudo-label refinement for weakly supervised semantic segmentation*, 2021. arXiv: 2108.01344 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2108.01344.

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

[65] S. Zheng et al., *Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers*, 2021. arXiv: 2012.15840 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2012.15840.

[66] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Learning deep features for discriminative localization*, 2015. arXiv: 1512.04150 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1512.04150.

[67] B. Zhou et al., *Semantic understanding of scenes through the ade20k dataset*, 2018. arXiv: 1608.05442 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1608.05442.

[68] T. Zhou, M. Zhang, F. Zhao, and J. Li, "Regional semantic contrast and aggregation for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4299–4309.