CamGuide:
An Efficient Method for Small Weakly Supervised Models

João Lucas Aparecido Rocha Paes, Wesley Nunes Gonçalves

[1] ***Abstract*—Weakly Supervised Semantic Segmentation (WSSS) aims to reduce the annotation cost associated with pixel-level semantic segmentation by relying only on image-level labels. However, most existing approaches depend on large architectures with high representational capacity and substantial computational requirements. This work investigates the potential of lightweight architectures for WSSS through a multi-stage training strategy based on structural distillation, stochastic semantic constraints, and heuristic refinement. The method combines teacher-guided learning, heuristics and morphological operations to improve localization quality and spatial consistency. Experiments conducted on the PASCAL VOC 2012[8] dataset demonstrate that the proposed method achieves 64.89% mIoU on the training split, 61.63% mIoU on the validation and 54.13% on the test split using a lightweight ResNet18 backbone. The results suggest that small networks remain a viable research direction for WSSS and may serve as efficient pseudo-label generators for subsequent semantic segmentation pipelines.**

***Index Terms*—Deep Learning, Semantic Segmentation, Weakly Supervised, Small Nets.**

I. Introduction

SEMANTIC segmentation has become increasingly important in applications such as autonomous robotics, medical systems, industrial inspection, and intelligent surveillance[21, 28, 29]. This relevance stems from its ability to perform fine-grained pixel-level analysis, enabling highly precise scene understanding and decision-making. However, achieving such performance requires high-quality pixel-wise annotations, which represent a major challenge in the training process [19].

In classification tasks, whether single-label or multi-label, annotations only indicate whether a class is present or absent within an image. However, in semantic segmentation, a similar process occurs at the pixel level. Consequently, the annotation cost is substantially smaller than that of semantic segmentation. In contrast, segmentation datasets require annotators, reviewers, annotation guidelines, and domain knowledge to ensure consistency and quality. Even with these resources, the generated masks may still contain inaccuracies and inconsistencies. Therefore, if a semantic segmentation model could be trained using only image-level classification labels while maintaining competitive performance, the cost, complexity, and development time associated with segmentation systems would be significantly reduced.

Nowadays, Weakly Supervised Semantic Segmentation (WSSS) remains an open and challenging research problem. Recent state-of-the-art approaches have increasingly incorporated additional low-cost sources of supervision to compensate for the limitations imposed by image-level labels alone. Examples include CLIP-based prompt learning [17], prototype-based methods [30], and other foundation-model-driven techniques.

Most contemporary approaches rely on large-scale architectures with substantial representational capacity and require considerable computational resources as substantiated in [18]. As a consequence, reproducibility is often reduced and practical deployment becomes more challenging. In contrast, the use of lightweight architectures for WSSS remains comparatively unexplored.

This work revisits classical and foundational WSSS methods to investigate the challenges associated with lightweight networks. In particular, we focus on two recurrent limitations observed in small models: fragmented activations and weak spatial consistency.

This method uses a student-teacher model, in which a larger, more powerful model, called the 'teacher', teaches the student network. This strategy enables the student network to learn its own internal representation while preserving the semantic organization induced by the teacher. We argue that such structural guidance is particularly beneficial for lightweight architectures where representational capacity is limited and spatial consistency is often compromised [4, 32].

To address this problem, we propose a multi-stage training strategy based on knowledge distillation [11] and stochastic masking [26], where pseudo-labels generated by a teacher network guide the student model during optimization. Unlike conventional distillation approaches, which typically transfer knowledge by forcing a student model to mimic the output distribution or intermediate representations of a larger teacher model, the proposed framework allows the student to learn its own representation while respecting its architectural limitations. In addition, heuristic refinement techniques, including Dense Conditional Random Fields (dCRF) [14], Random Walk (RW) propagation [15], morphological operations, and other auxiliary methods, are employed to improve the quality of the generated activation maps. Through this combination, we investigate a more heuristic and lightweight approach for obtaining competitive results in classical WSSS scenarios.

II. Methodology

A. Motivation

In weakly supervised semantic segmentation, limited-capacity networks may have reduced ability to aggregate long-range spatial and semantic context across the feature hierarchy. This limitation is related to the effective receptive field, which often occupies only a fraction of the theoretical receptive field, making it harder for shallow models to capture large object

[1] AI was used to improve the grammar and coherence of this document.

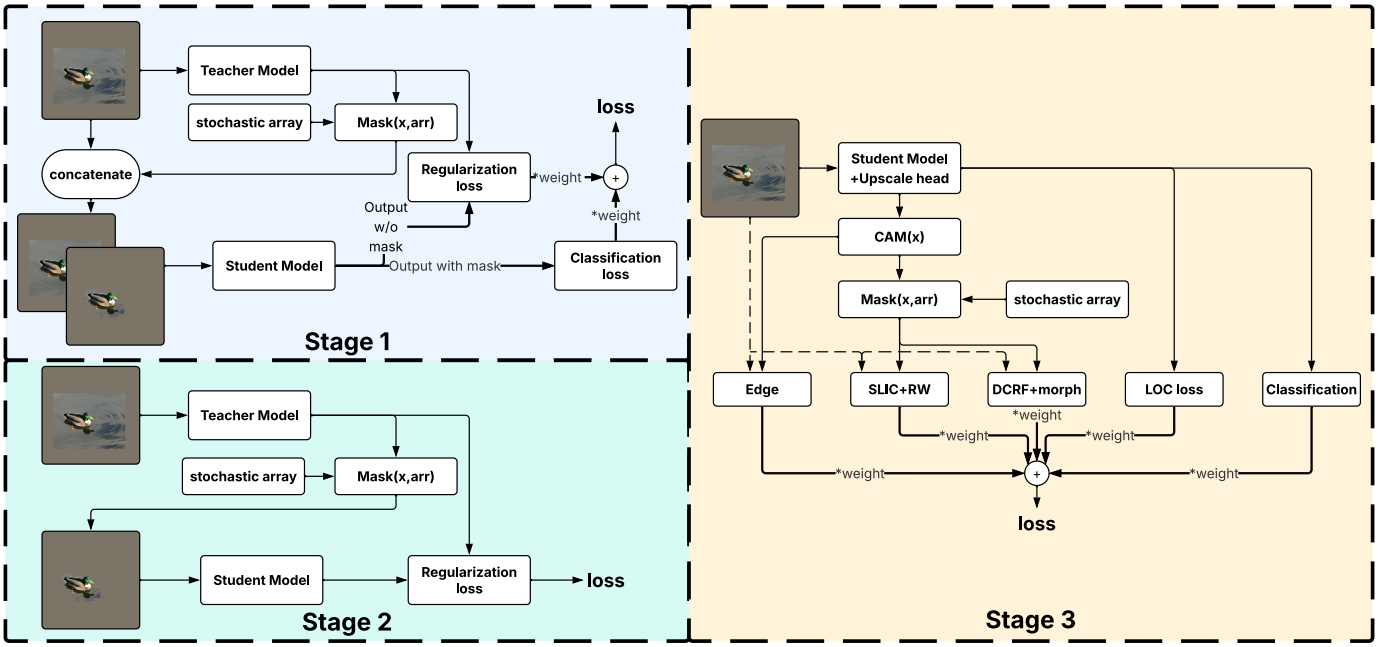


Fig. 1: CamGuide Methodology overview

regions and global spatial dependencies [20, 32]. In CAM-based WSSS, this issue becomes particularly critical because pseudo-label generation depends on classification-driven activations as localization cues. However, CAMs are known to focus primarily on the most discriminative object parts rather than the complete object extent, partly due to biases introduced by global average pooling and the classification objective [3, 27]. Consequently, attempts to expand CAM responses toward complete object regions may still produce incomplete localization maps or introduce noisy activations that spill over into the background, affecting both convolutional and transformer-based backbones [16, 22]. Moreover, without explicit spatial constraints, expanded activations may fail to preserve object structure and align with actual object boundaries [5].

Despite these limitations, small networks are not inherently constrained to under perform relative to larger models. Instead of relying solely on discriminative keypoints, our method encourages the student network to capture broader morphological structures by leveraging guidance from a teacher model. Specifically, we employ the teacher's regularization signal to stabilize the learning dynamics of the student, while a multi-label loss reinforces the activation of multiple relevant regions. This combination enables the student model to learn a structured anchor from the teacher, mitigating instability and improving spatial coherence.

As a result, the proposed method enhances the quality of CAMs by filtering background noise and promoting more consistent activation masks, ultimately enabling small networks to approximate the representational behavior of deeper architectures.

The general structure can be seen in Fig. 1.

B. CAM Method

As our baseline, we adopt the Class Activation Map (CAM) formulation proposed in Puzzle-CAM [24], ensuring a fair comparison under identical localization settings. During inference, we follow their class-wise activation strategy to generate class-specific localization maps. Where:

$$features = model(image) \tag{1}$$

Where $features \in \mathbb{R}^{B\times K\times H'\times W'}$ where K is the number of classes and H', W' are spatial dimensions after downsampling, B is the batch size and $F_T(X)$ is the teacher forward.

$$feats_{amax} = max_{j,i}(\epsilon, features_{j,i} + \epsilon) \tag{2}$$

The epsilon is an insignificant value and $features_{j,i}$ represents the model output.

$$CAM_{k,j,i} = \frac{ReLU(\ features_{k,j,i} - \epsilon)}{feats_{amax}} \tag{3}$$

C. Pretrain

Our pipeline consists of two networks: a deep teacher model and a lightweight student model. The teacher network is trained on the dataset using standard classification supervision to obtain a well-optimized representation. This stage allows the integration of different pretraining strategies, as long as they follow the same classification objective.

For fair comparison, we also evaluate a baseline configuration using standard classification training without additional modifications.

D. Train

The training procedure consists of a three-stage optimization strategy designed to progressively transfer knowledge from the teacher network to the student network and graduate the model. The first stage focuses on guiding the student to align with the teacher's predictive behavior, while the second stage refines the learned representations to improve spatial consistency and localization quality in small networks. The third stage focuses on using heuristics to reduce the model's error. During student training, the teacher network remains frozen.

E. Definitions

1) Teacher Forward: Given an input batch $x \in \mathbb{R}^{B\times C\times H\times W}$, the teacher network produces feature maps where:

$$F_T(X) \in \mathbb{R}^{B\times K\times H'\times W'} \tag{4}$$

Otherwise, the teacher branch is frozen, in other words:

$$\omega_T \leftarrow frozen \tag{5}$$

where ω_T denotes the teacher weights.

2) Stochastic thresholding mask: We define a stochastic threshold vector:

$$\tau_b \sim \mathcal{U}(\beta,\delta), \quad 0<\beta<\delta<1, \quad \tau \in \mathbb{R}^{B\times k} \tag{6}$$

Where τ_b is a random value from a uniform distribution in $[\beta,\delta]$. Each sample threshold is applied independently to the teacher feature maps and produces:

$$Mask_T = F_T(X) > \tau \tag{7}$$

Where the mask is used to filter the input image:

$$x_{cropped} = x \odot Mask_T \tag{8}$$

x is the original image.

3) Defines logits: To calculate the logits, class-wise, uses the average over width and height. I.e:

$$logits = \frac{1}{H'\times W'}\sum_{i=1}^{H'}\sum_{j=1}^{W'} feature[i,j] \tag{9}$$

F. Stage 1

The first stage is responsible for initializing the student network with semantically meaningful and class-discriminative representations. At this point, the objective is not yet to refine the final segmentation masks, but to guide the lightweight student model toward stable class activation responses.

To achieve this, Stage 1 combines two complementary training signals. First, an image-level classification objective encourages the student to identify the semantic classes present in the input image. This term is necessary because the available supervision is given only at image level, and therefore the model must learn discriminative correlations between visual patterns and class labels. Second, a teacher-student regularization term is used to anchor the student representation to the semantic structure produced by the teacher network.

Given an input image (x), the teacher network produces a semantic reference representation. From this representation, a stochastic thresholding mask is generated and applied to the input image, producing a masked view ($x_{cropped}$). The student network then receives both the original and the masked views, allowing the model to learn class evidence from the complete image while also being regularized under partial semantic suppression.

1) Student forward: Given the input pair ($[x, x_{cropped}]$), where ($x, x_{cropped} \in \mathbb{R}^{B\times C\times H\times W}$), the student network produces class-wise convolutional feature maps for both views:

$$F_S([x, x_{cropped}]) \in \mathbb{R}^{2B\times K\times H'\times W'}. \tag{10}$$

Equivalently, this output can be decomposed into the feature maps obtained from the original image and from the masked image:

$$F_S([x, x_{cropped}]) = [F_S(x), F_S(x_{cropped})], \tag{11}$$

where:

$$F_S(x), F_S(x_{cropped}) \in \mathbb{R}^{B\times K\times H'\times W'}. \tag{12}$$

In this stage, ($F_S(x_{cropped})$) is used to compute the image-level classification logits, following Equation 9, while ($F_S(x)$) is used in the teacher-student regularization objective defined later. This separation forces the student to learn discriminative class evidence from teacher-selected semantic regions, while the original view is used to preserve teacher-guided semantic structure through regularization.

G. stage 2

Stage 2 preserves the same teacher-student structure used in Stage 1, but changes both the input condition and the optimization objective. In this stage, the teacher receives the original image x and produces the reference representation $F_T(x)$, while the student receives the masked image x_{masked}, generated from the stochastic teacher mask, and produces $F_S(x_{masked})$.

Unlike Stage 1, no classification loss is used in this stage. The objective is to force the student to preserve the semantic response of the original image even when part of the visual evidence is suppressed by the mask. Therefore, Stage 2 acts as a masked feature-consistency refinement step.

H. Stage 3

Stage 3 is designed as the final refinement stage of the proposed training pipeline. At this point, the student network has already learned image-level semantic discriminators from Stage 1 and has been further regularized under stochastic masked inputs in Stage 2. Therefore, the objective of this stage is no longer to transfer the teacher representation directly, but to improve the spatial quality, boundary coherence, and local consistency of the activation maps produced by the student model.

In this stage, the ResNet-18 backbone is extended with the proposed refinement head. The output of the backbone is first

subjected to a dropout operation before being forwarded to the refinement module. This dropout prevents the refinement head from relying exclusively on the most discriminative classifier responses and encourages the model to exploit complementary spatial information from earlier feature maps. The refinement head then combines the coarse class activation maps with intermediate backbone features and image-guided structural information, producing refined class-specific activation maps with higher spatial resolution.

The primary semantic supervision remains the image-level classification objective. This term preserves the discriminative capability of the model and prevents the refinement process from drifting toward classes that are not present in the image. However, classification supervision alone does not explicitly enforce spatial coherence. For this reason, Stage 3 introduces auxiliary heuristic supervision signals designed to regularize the generated activation maps.

All auxiliary terms operate over the refined feature maps produced by the new module, as defined in Equation 10, after CAM normalization according to Equation 3. These normalized maps are used to generate or compare against heuristic targets based on Dense Conditional Random Fields, Random Walk propagation over SLIC superpixels, edge consistency, and local window consistency. The dCRF branch encourages alignment with image appearance and object boundaries, while the Random Walk branch promotes semantic propagation across neighboring superpixels, reducing fragmented activations. The edge-consistency term penalizes semantic transitions that are not supported by image structure, and the local-consistency term suppresses isolated high-confidence responses by encouraging agreement within local neighborhoods.

Unlike the previous stages, Stage 3 can be interpreted as a graduation step from direct teacher guidance. The model is no longer optimized to simply match the teacher representation. Instead, it uses its own predictions, refined by image-guided and graph-based heuristics, to improve the quality of the generated pseudo-labels. In this sense, the stage shifts the training objective from teacher-student alignment to spatial refinement of the student's class activation maps.

Following the heuristic refinement step, the dCRF-derived targets are post-processed using morphological closing in order to reduce fragmented multi-focal regions associated with the same object. This operation acts as a structural correction mechanism over the pseudo-labels, improving target stability before they are used as supervision. The complete optimization objective for this stage, including the classification and auxiliary terms, is defined later in the training objectives subsection.

I. Refinement Components

1) Heuristics refinement stage: At this stage, the student network is expected to "graduate" from the teacher guidance and refine its predictions using auxiliary heuristic signals. The heuristic stage focuses on aligning the generated masks with the structural information present in the input image, improving the spatial consistency of the pseudo-labels through image-guided refinement strategies.

However, because the refinement process remains dependent on the model predictions, systematic artifacts and error propagation may still emerge during optimization. To mitigate these effects, we introduce an additional morphological refinement stage operating exclusively over the predicted masks. Unlike the heuristic refinement, this stage does not directly use image information, but instead acts as a structural correction mechanism designed to compensate persistent failure behaviors produced by the network and the heuristic targets.

The refinement pipeline combines three heuristic strategies: Dense Conditional Random Fields (dCRF) [14], edge-consistency supervision, and random-walk graph [15] propagation. In particular, the edge-consistency heuristic constrains semantic transitions in the CAM according to image gradient support extracted with the Scharr operator [25], encouraging alignment between activation boundaries and visually consistent image structures.

After heuristic generation, classical morphological operations, such as erosion, dilation, opening, and closing, are applied to stabilize the generated targets and suppress undesirable prediction patterns, including excessive false-positive activations and fragmented semantic regions. The selected operations and structural parameters were empirically defined according to optimization stability, qualitative mask behavior, and validation performance during refinement.

2) Random Walk Graph propagation: The Random Walk propagation is performed over a superpixel graph constructed using SLIC segmentation [1]. Given an image partitioned into superpixels $s \in S$, we define the unary representation for class k as the mean activation inside each superpixel:

$$u_{k,s} = \frac{1}{|s|} \sum_{(i,j)\in s} target[i,j] \tag{13}$$

A weighted adjacency graph is then constructed using 4-neighborhood connectivity between superpixels, where edge weights are defined based on color similarity:

$$w_{transaction} = e^{-\frac{||\mu_{local} - \mu_{neighbor}||}{2\sigma^2_{color}}} \tag{14}$$

and σ_{color} is a fixed hyperparameter controlling sensitivity to appearance differences. The transition matrix is obtained via row normalization:

$$P_{transaction} = \frac{w_{transaction}}{\sum_{i\in neighborhood} w_i} \tag{15}$$

Random walk diffusion is then applied iteratively in the superpixel space:

$$u_{t+1} = P_{transaction} * u_t \tag{16}$$

3) Edge consistency pipeline: Before anything else, this heuristic rests on a single structural assumption that is essential for coherent supervision: not every image edge is a class boundary, but every class boundary should be supported by image structure. In other words, we do not require the CAM to reproduce all photometric edges (textures, shadows, clutter), yet we penalize semantic discontinuities that appear where the image is locally smooth or weakly structured. Symmetric

metrics such as cosine similarity or ℓ_1 distance between edge maps would treat false positives in the image and false positives in the CAM symmetrically; our training objective is deliberately asymmetric to encode this implication.

The procedure is a lightweight edge-alignment pipeline. Given an input image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ and class activation maps $M \in \mathbb{R}^{C\times H'\times W'}$ for the image-level active classes, both signals are low-pass filtered before gradient extraction. The image is blurred with a Gaussian kernel of size 9×9 and $\sigma = 3$, which attenuates high-frequency texture and reduces spurious edge responses. The CAMs are blurred with a 5×5 Gaussian filter ($\sigma = 1.5$), which slightly spreads activations and stabilizes per-class edge estimation without destroying localization. Gradient magnitude is then computed with a Scharr operator [25] on each channel; for the image, the channel-wise gradient norms are averaged, whereas for the CAM they are kept per class. Both edge-strength maps are normalized by their global maximum over spatial locations (and, for the CAM, per channel), so that comparisons are scale-invariant across samples.

From the blurred image gradients we build an image edge-support map $S^{\text{img}} \in [0,1]^{1\times H\times W}$, followed by a 3×3 morphological dilation to widen thin structures and improve tolerance to small misalignment. The CAM yields a per-class semantic edge map $E^{\text{cam}} \in [0,1]^{C\times H_m\times W_m}$. Because the CAM may live on a coarser grid, S^{img} is resized to the CAM resolution via adaptive average pooling, producing $S^{\text{img}}\downarrow$. Only pixels belonging to classes present in the image-level label vector are considered, through a binary mask $y_{b,k}$ broadcast over space.

4) Model upscale module: In the upscaling module, the inputs are the classifier output, the ResNet-18 stage-1 output, and the image. Let the image be $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, the classifier output be $\mathbf{x} \in \mathbb{R}^{K\times \frac{H}{16}\times \frac{W}{16}}$, and the stage-1 feature map be $\mathbf{x}_{\text{stg1}} \in \mathbb{R}^{C_1\times \frac{H}{4}\times \frac{W}{4}}$. The module has two parts: (i) fusion-upscale, in which $\mathbf{x}$ and $\mathbf{x}_{\text{stg1}}$ are merged to produce class maps at stage-1 resolution, and (ii) gated refinement, in which an edge-aware guided filtering branch and a gated residual branch further refine the maps.

First, the coarse class map is bilinearly upsampled to stage-1 resolution:

$$\mathbf{x}_{\uparrow} = \text{Upsample}(\mathbf{x}) \in \mathbb{R}^{K\times \frac{H}{4}\times \frac{W}{4}}. \tag{17}$$

Then, fusion is performed by concatenating $\mathbf{x}_{\uparrow}$ and $\mathbf{x}_{\text{stg1}}$, followed by convolutional projection:

$$\mathbf{h}_0 = \phi([\mathbf{x}_{\uparrow}, \mathbf{x}_{\text{stg1}}]), \qquad \mathbf{z}_0 = \mathbf{x}_{\uparrow} + \psi(\mathbf{h}_0), \tag{18}$$

where $\phi(\cdot)$ denotes the fusion block (Conv-BN-ReLU with depthwise mixing) and $\psi(\cdot)$ is a 1×1 projection to K channels.

In the gated [6] refinement stage, one refinement step is:

$$\mathbf{g} = \text{GF}(\mathbf{z}_0, \mathbf{I}_{\downarrow}), \tag{19}$$

where GF is the guided filter [10] and $\mathbf{I}_{\downarrow} \in \mathbb{R}^{3\times \frac{H}{4}\times \frac{W}{4}}$ is the downsized image guide. The guided filter is an edge-aware filtering operator that locally models the output as a linear transformation of a guidance image. By using the downsized image $\mathbf{I}_{\downarrow}$ as guide, the refinement process incorporates low-level visual structure, encouraging smooth responses within homogeneous regions while preserving discontinuities aligned with image boundaries

A residual update branch is computed from $\mathbf{g}$:

$$\mathbf{r} = f_{\text{res}}(\mathbf{g}), \tag{20}$$

And a gate map is computed as:

$$\mathbf{a} = \sigma(r), \qquad \mathbf{a} \in [0,1]^{K\times \frac{H}{4}\times \frac{W}{4}}. \tag{21}$$

Where σ is the sigmoide function. Finally, the refined output is:

$$\mathbf{z}_1 = \mathbf{z}_0 + \mathbf{a} \odot \mathbf{r}. \tag{22}$$

This operation corresponds to an image-guided gated residual refinement, implemented as specified in the accompanying figure 2.

5) Window local consistency formulation: The Window Local Consistency is inspired by affinity-based regularization methods [2]. Its objective is to encourage local semantic coherence by enforcing agreement between each pixel and the semantic distribution of its surrounding neighborhood. The underlying assumption is that nearby pixels belonging to the same object tend to share similar semantic responses, resulting in locally smooth activation maps.

Unlike conventional affinity losses that explicitly model pairwise relationships between neighboring pixels, the proposed formulation compares each pixel with the average semantic distribution within a local window. Consequently, the loss acts as a spatial regularizer that suppresses isolated activations and reduces high-frequency semantic noise. Although this assumption may not hold near object boundaries, the contribution of this loss is controlled by the weighting coefficient λ_{loc}, preventing excessive smoothing.

J. Training Losses

1) Classification loss: From the stage 1 and the stage 3, the classification loss was the MultiLabelMarginLoss defined as:

$$\mathcal{L}_{cls} = \sum_{i,j} \frac{max(0, 1 - (logits[y[j]] - logits[i]))}{logits.size(0)} \tag{23}$$

Where y is the label defined as $y \in [0,1]^{B\times K}$.

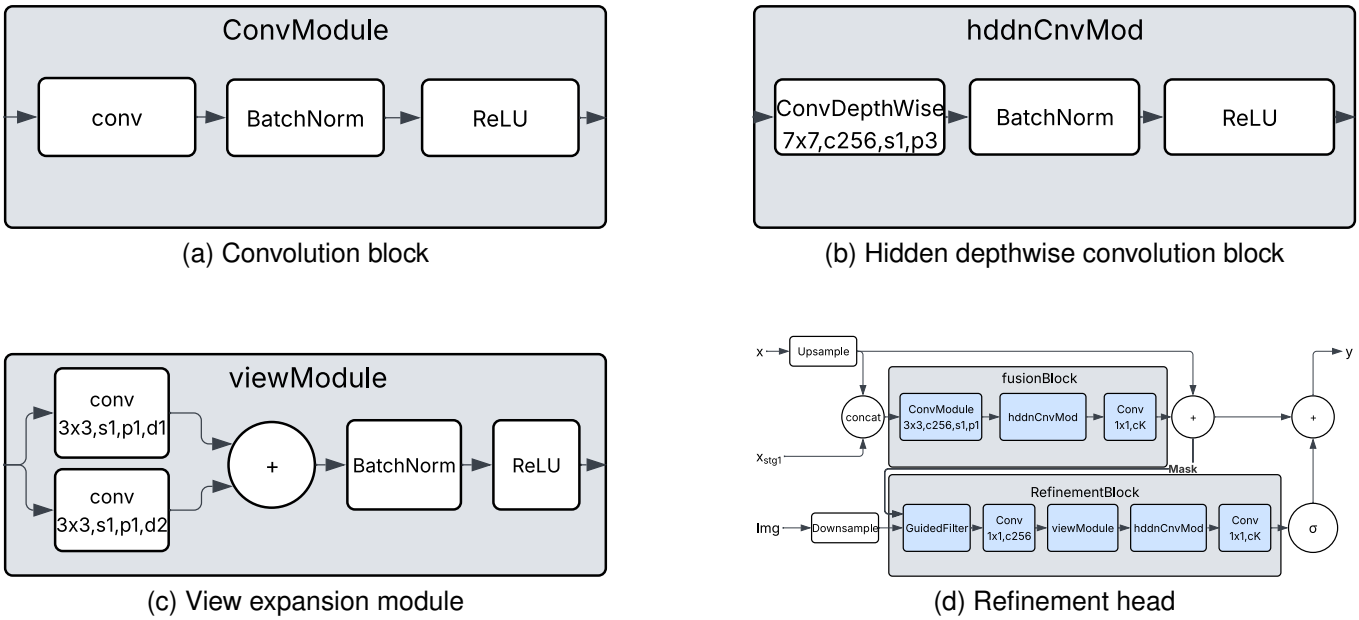
2) Regularization loss: The teacher-student regularization term is implemented as an L1 distance between teacher and student feature maps. However, the input condition differs across stages.

In Stage 1, the regularization is applied between the teacher representation from the original image and the student representation from the original view:

$$\mathcal{L}_{reg}^{(1)} = \|F_T(x) - F_S(x)\|_1. \tag{24}$$

In Stage 2, the regularization is applied as a masked consistency objective, where the teacher receives the original image and the student receives the masked image:

$$\mathcal{L}_{reg}^{(2)} = \|F_T(x) - F_S(x_{masked})\|_1. \tag{25}$$



(a) Convolution block (b) Hidden depthwise convolution block

(c) View expansion module (d) Refinement head

Fig. 2: Architectural modules used in the proposed refinement pipeline.

3) Heuristic algorithms loss: Two heuristic supervision losses are used. The standard structure used was:

$$target = morph(algorithm(mask_{stochastic})) \tag{26}$$

where the algorithm can be any non-derivable method to improve CAM response and generate an pseudo-label, where this algorithm can be supported by an morphological operation to produce the target that spread guidance signal to model response as:

$$\mathcal{L}_{example} = CE(CAM_{student}, target) \tag{27}$$

The $CAM_{student}$ is produced using 3 from the student features.

The first loss is the DCRF, where the morphological operation was the closure:

$$\mathcal{L}_{crf} = CE(CAM_{student}, closure(dcrf(mask))) \tag{28}$$

The second was the SLIC + RW algorithm without morphological operation:

$$\mathcal{L}_{rw} = CE(CAM_{student}, SLIC\_RW(mask)) \tag{29}$$

4) Edge consistency loss: The core violation variable measures how much a CAM edge exceeds the local image support, but only where the image provides weak support. Let τ denote the edge-support threshold (in our run, $\tau = 0.1$), and let $\delta = 0.05$ be a small margin that avoids penalizing negligible mismatches. For each active class c and pixel (i, j),

$$u_{b,k,i,j} = \mathrm{ReLU}\left(E^{\mathrm{cam}}_{b,k,i,j} - \left(S^{\mathrm{img}}_{\downarrow,b,i,j} + \delta\right)\right) \cdot \mathbb{1}\left[S^{\mathrm{img}}_{\downarrow,b,i,j} < \tau\right] \cdot y_{b,k} \tag{30}$$

Thus u is zero whenever the CAM edge lies within the tolerated band above the image support, and it is also zero wherever the image already exhibits sufficiently strong edge structure ($S^{\mathrm{img}}_{\downarrow} \geq \tau$), An increase in τ implies that this low-pass filter will impose a more stringent attenuation on the resulting loss. This is the operational form of the implication "class-edge $\Rightarrow$ image-edge support": CAM-only edges in smooth or weakly supported regions are penalized, while strong image edges without a matching CAM edge are not punished.

The denominator for normalization is the number of supervised pixels that satisfy both conditions—active class and weak image support:

$$\mathcal{N} = \sum_{b,k,i,j} y_{b,k}, \mathbb{1}\left[S^{\mathrm{img}}_{\downarrow,b,i,j} < \tau\right] \tag{31}$$

The loss function comprises two complementary components constructed from u. The first component applies a focal-style modulation with exponent $\gamma = 2$. The parameter γ serves as a hyperparameter that primarily controls the model's sensitivity to large violations: contributions associated with small values of u are attenuated more rapidly than those corresponding to large values of u.

$$\mathcal{L}_{\mathrm{focal}} = \frac{1}{\mathcal{N}} \sum_{b,k,i,j} u^{\gamma}_{b,k,i,j}. \tag{32}$$

The second term performs hard mining on the strongest CAM-edge responses. Let $\tau_2 = 0.35$ be a high-confidence threshold on E^{cam}, In this context, an increase in τ_2 exerts a permissive direct effect on this filter. Unlike the focal term, the hard component is linear in u but restricted to pixels where the CAM edge is pronounced:

$$\mathcal{L}_{\text{hard}} = \frac{1}{\mathcal{N}} \sum_{b,k,i,j} u_{b,k,i,j}, \mathbb{1}\left[E^{\text{cam}}_{b,k,i,j} > \tau_2\right] \tag{33}$$

Basically, there are two principal approaches for penalizing high confidence when it is associated with a large divergence from the image's morphological characteristics. The full edge-consistency objective is:

$$\mathcal{L}_{\text{edge}} = \mathcal{L}_{\text{focal}} + \lambda_{\text{hard}}.\mathcal{L}_{\text{hard}}, \qquad \lambda_{\text{hard}} = 2.0, \tag{34}$$

5) Window local consistency loss: First, the feature responses are normalized across the semantic dimension to obtain a probability distribution for each pixel:

$$K_{k,i,j} = \frac{features_{k,i,j}}{\sum_{k'} features_{k',i,j} + \epsilon}. \tag{35}$$

This normalization ensures that

$$\sum_k K_{k,i,j} = 1, \tag{36}$$

allowing each pixel to be interpreted as a categorical probability distribution over the semantic classes.

Next, the average semantic distribution within a local window $\Omega(i,j)$ is computed as

$$\bar{K}_{k,i,j} = \frac{1}{|\Omega|} \sum_{(u,v)\in\Omega(i,j)} K_{k,u,v}. \tag{37}$$

The local agreement score is then defined as

$$S_{i,j} = \sum_k K_{k,i,j} \bar{K}_{k,i,j}, \tag{38}$$

which corresponds to the inner product between the semantic distribution of the central pixel and the average distribution of its neighborhood. Higher values indicate stronger agreement with the local context.

Finally, the loss penalizes disagreements between a pixel and its surrounding neighborhood while assigning larger weights to highly confident predictions:

$$\mathcal{L}_{loc} = \frac{1}{HW} \sum_i \sum_j (1 - S_{i,j}) \cdot \max_k K_{k,i,j}. \tag{39}$$

As a result, highly confident predictions that are inconsistent with the local semantic context receive stronger penalties, encouraging spatially coherent activation maps while preserving discriminative responses.

6) Final loss for Stage 1: In Stage 1, the classification and regularization terms are applied to different student views. The classification loss is computed from the cropped view, since this view contains the semantic regions selected by the teacher-guided stochastic mask. The regularization term is computed using the student response from the original image and the teacher-guided reference representation.

The final objective for Stage 1 is therefore defined as:

$$loss = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{reg}\mathcal{L}^{(1)}_{reg} \tag{40}$$

where λ_{cls} and λ_{reg} control the relative contribution of the classification and regularization terms.

Teacher	resnet50
Student	resnet18
pre-train	classification Imagenet1k[23]
Batch Size	32
AMP	not used
Image Size	512
Optimizer	Adam
Learning Rate	0.1
Weight Decay	1e-4
Scheduler	Reduce On Plateau
Params Scheduler	patience 500; factor 0.5
Image Normalization	ImageNet values
λ_{cls}	0.35
λ_{reg}	0.65
Epochs	15
Stochastic threshold min	0.1
Stochastic threshold max	0.45

TABLE I: Stage 1 parameters

7) Final loss for stage 2: The final loss for Stage 2 is therefore given by:

$$loss_{stage2} = \mathcal{L}^{(2)}_{reg}. \tag{41}$$

8) Final loss for stage 3: The final loss function is defined as follows

$$loss = \mathcal{L}_{edge}.\lambda_{edge} + \mathcal{L}_{crf}.\lambda_{crf} + \mathcal{L}_{rw}.\lambda_{rw} + \mathcal{L}_{cls}.\lambda_{cls} + \mathcal{L}_{loc}.\lambda_{loc} \tag{42}$$

III. Results

A. Implementation Details

The computer used in this train was two 4060ti 16GB, OS Ubuntu 24.04.3 LTS and CUDA 13.0. The dataset used was PASCAL VOC 2012 [7]. The method used was based in [24], so the train split was 10,582 images [9], 1,449 for validation split, these two was affected by augmentations as re-scale [320, 640] and 512x512 crop (online method). As our method doesn't has any method of consistency by scale and view, the inference method that we used adopted multi-scale, horizontal-flip and as the PASCAL VOC has an natural imbalance between the classes in this inference method we used class-wise threshold.

1) Model Selection: To get the well-optimized representation from the train in stage 1 and 2, it's necessary to use classification methods of evaluation on train split.

B. Parameters

All parameters in Stage 2 are identical to those used in Stage 1, except for:

λ_{cls}	0
λ_{reg}	1

TABLE II: Stage 2 parameters

Model	resnet18
Batch Size	32
AMP	float16
Image Size	512
Optimizer	SGD + Nesterov Momentum 0.9
Learning Rate	0.009
Weight Decay	1e-4
Scheduler	Polinomial
Image Normalization	ImageNet values
Guided Filter Radius kernel size	4
λ_{cls}	1.00
λ_{crf}	0.80
λ_{edge}	0.70
λ_{rw}	0.40
λ_{loc}	0.45
Closure kernel size	15
Window Local Consistency	7
dCRF parameter	iterations 5;
RW-SLIC Parameters	segments 1100; iterations 12
dCRF each n iterations	4
Dropout Head Classifier	0.50
Epochs	10
Stochastic threshold min	0.05
Stochastic threshold max	0.30

TABLE III: Stage 3 parameters

C. Results

Our main metric was the mean Intersection-Over-Union (mIoU) [IV,V,VI]:

Backbone	Method	Param.(M)	threshold	mIoU(%)
Resnet18	Our+RW+dCRF	15.41	Global	61.46
Resnet18	Our+RW+dCRF	15.41	Class-wise	**64.89**

TABLE IV: Final mIoU results

The final performance on the training split achieved using the AffinityNet, RandomWalk, and dCRF pipeline was 64.89%, representing an improvement of more than 20 mIoU points compared to the baseline ResNet-18 model trained solely with classification signal:

Backbone	Method	Param.(M)	threshold	mIoU(%)
Resnet50	Classification only	23.56	Global	47.44
Resnet18	Classification only	14.00	Global	45.01
Resnet18	Our*	14.00	Global	47.59
Resnet18	Our*	14.00	Class-wise	48.97
Resnet18	Our**	15.41	Global	48.22
Resnet18	Our**	15.41	Class-wise	50.00
Resnet18	Our†	15.41	Global	49.14
Resnet18	Our†	15.41	Class-wise	50.96
Resnet18	Our⋆	15.41	Global	49.34
Resnet18	Our⋆	15.41	Class-wise	51.13
Resnet18	Our⋆+RW	15.41	Global	60.98
Resnet18	Our⋆+RW	15.41	Class-wise	64.31
Resnet50	PC[24]+RW+DCRF	23.56	Global	**64.70**

TABLE V: **Train** split. *: stage 1 and 2. **: all stages without morphological operations. †: all stages with morphological operations. ⋆: all stages with morphological operations and local consistency. RW: Random Walk with AffinityNet. PC: PuzzleCam

Our unrefined model surpasses the teacher baseline by 1.90% when using a global average threshold. When employing class-wise thresholds, the performance improvement increases to 3.59% over the teacher baseline and remains only 0.4% below the ResNet-50 results reported in the PuzzleCam study [24] without post-processing.

The Random Walk with dCRF configuration attains a performance of 64.89%, which is 0.19% higher than the result reported for PuzzleCam, while relying on a model that is 34.6% smaller (8.15M parameters).

Backbone	Method	Param.(M)	threshold	mIoU(%)
Resnet50	No one	23.56	global	47.11
Resnet18	No one	14.00	global	43.59
Resnet18	Our*	14.00	global	45.48
Resnet18	Our*	14.00	Class-wise	46.77
Resnet18	Our†	15.41	Global	47.46
Resnet18	Our†	15.41	Class-wise	49.09
Resnet18	Our†+RW	15.41	Global	57.13
Resnet18	Our†+RW	15.41	Class-wise	61.13
Resnet18	Our†+RW+dCRF	15.41	Global	57.55
Resnet18	Our†+RW+dCRF	15.41	Class-wise	**61.63**

TABLE VI: mIoU results on **Validation** split. *: stage 1 and 2. †: All stages with morphological operations and local consistency.

Backbone	Method	Val	Test
Resnet18	CamGuide	61.6	54.1
Wide-ResNet38	AffinityNet[2]	61.7	63.2
ResNet101	DSRG[13]	61.4	63.2
ResNet101	SeeNet[12]	63.1	62.8
Wide-ResNet38	SEAM[31]	64.5	65.7
ResNest269	PuzzleCAM[24]	71.9	72.2

TABLE VII: Classical methods (mIoU)

Evaluation on the validation split showed that the proposed approach improved class-wise performance by 1.98% compared to the baseline, achieving a final score of 61.6% after Random Walk and dCRF post-processing.

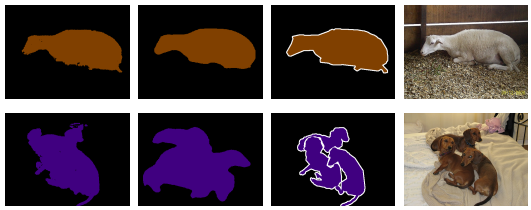


Fig. 3: **Qualitative segmentation results** on PASCAL VOC 2012. *First:* Prediction of the CAM from resnet18+RW+dCRF class-wise threshold. *Second:* Prediction of the segmentation model using the pseudo-labels from PuzzleCAM. *Third:* Ground Truth. *Fourth:* Original images.

The model, as we can see, showed high discernment between classes, but experienced difficulties in maintaining spatially consistent pixel-wise logits, which impaired mask propagation into object regions with weak semantic evidence.

IV. Discussion

A. Distillation Alignment

The primary contribution of Stages 1 and 2 lies in the proposed distillation strategy based on stochastic semantic constraints. Unlike conventional knowledge distillation approaches, which directly transfer semantic representations from the teacher network to the student network, the proposed method uses teacher-guided stochastic masks to constrain the

student's learning process. The masked input encourages the student to focus on semantically relevant regions identified by the teacher, while the original image simultaneously forces the model to learn how to suppress irrelevant content and background activations. This dual-view strategy allows the student network to learn its own internal representation while preserving the semantic structure induced by the teacher.

The stochastic thresholding mechanism plays a fundamental role in this process. Since no pixel-level annotations are available, selecting a fixed threshold would introduce a strong prior regarding which teacher activations should be considered correct. Such assumptions could propagate systematic localization errors throughout the entire training pipeline. Instead, stochastic threshold sampling exposes the student to multiple semantic views of the same image, reducing the dependence on a single threshold configuration and improving robustness. Stage 2 can be interpreted as a refinement phase of Stage 1, where the objective is to reinforce semantically relevant regions while progressively encouraging the student to reproduce the concentrated object responses generated by the teacher. As demonstrated in the experimental results, this strategy alone improved the baseline performance from 45.01% to 47.59% mIoU, indicating that structural guidance from the teacher contributes positively to the localization capabilities of lightweight architectures.

B. Refinement Stage

A major limitation of the distillation stages is that the student network may inherit localization errors and biases present in the teacher model. Although the classification objective improves semantic discrimination, it does not explicitly enforce spatial coherence. Consequently, due to the limited representational capacity of lightweight architectures, the model tends to converge toward highly discriminative but spatially incomplete object regions. The refinement stage was introduced to address this limitation through the incorporation of heuristic supervision mechanisms capable of expanding and regularizing the generated activation maps.

However, heuristic supervision introduces an additional challenge: erroneous predictions may be reinforced and propagated during optimization. To mitigate this effect, Dense Conditional Random Fields (dCRF) were adopted as the primary auxiliary refinement mechanism due to their strong ability to recover object boundaries and improve mask consistency. Furthermore, dCRF supervision was applied sparsely during training, using a reduced number of iterations and periodic updates, with the objective of encouraging the network to learn the underlying object morphology while minimizing excessive error propagation. The classifier dropout also played an important role, preventing the refinement head from relying exclusively on the classifier output and forcing it to learn complementary spatial representations through the proposed upscaling and refinement modules. The effectiveness of this stage is reflected in the experimental results, where the complete refinement strategy achieved a performance improvement of 1.98% over the teacher baseline on the validation set.

C. New Module Impact

One of the main limitations of lightweight classification backbones is the low spatial resolution of the generated activation maps. In the case of ResNet18, the classifier output is produced at a downsampling ratio of 16. Consequently, an input image of size 512×512 generates an activation map of only 32×32, resulting in a significant loss of fine-grained spatial information. This limitation is one of the reasons why architectures such as ResNet38d became popular in the WSSS literature, as they provide richer intermediate representations and improved localization capabilities.

The proposed refinement module was designed to mitigate this limitation by recovering high-resolution information from earlier stages of the backbone. The fusion block combines the classifier output with the Stage-1 feature maps, allowing the model to reintroduce spatial details that were lost during the successive downsampling operations. Furthermore, the Guided Filter branch incorporates structural information directly from the input image, reinforcing object boundaries and improving spatial coherence in the generated activation maps.

Despite introducing only 1.41 million additional parameters, the proposed module provides an effective mechanism for recovering spatial information in lightweight architectures. During the early stages of development, the refinement block was initially designed as a recurrent architecture, where multiple refinement iterations would progressively improve the activation maps. However, this approach was eventually abandoned due to optimization instability and increased memory consumption, while providing only marginal performance improvements. As a result, the final design adopted a single-pass refinement strategy, achieving a more favorable trade-off between computational cost, stability, and segmentation performance.

D. Heuristic Losses Impact

The primary auxiliary supervision signal employed in the proposed framework is the dCRF-derived target. Unlike the other heuristic mechanisms, dCRF simultaneously exploits both appearance similarity and semantic information, allowing it to model relationships between pixels beyond local neighborhoods. This characteristic makes it particularly effective at refining weak responses and recovering fine object boundaries. Although heuristic supervision may propagate prediction errors, the main advantage of dCRF lies in its ability to correct imprecise object contours and expand under-segmented regions. As a consequence, the generated pseudo-labels become spatially more consistent, encouraging the student network to produce smoother and more complete object responses.

However, dCRF presents an important limitation: its propagation mechanism is highly dependent on color similarity. Consequently, regions belonging to the same object but exhibiting significant appearance variations may remain disconnected. To mitigate this issue, Random Walk (RW) propagation over SLIC superpixels was incorporated. The objective of this component is to establish semantic connections between fragmented activation regions and propagate information across

visually distinct areas. For example, an object partially illuminated by sunlight and partially covered by shadow may be segmented into separate appearance regions. In such cases, dCRF alone may fail to propagate activations between these regions, whereas the graph-based propagation performed by RW allows semantic information to diffuse through neighboring superpixels, reducing fragmentation and improving object coverage.

The Edge Consistency Loss was introduced to enforce structural coherence between the generated activation maps and the image content. Its objective is to discourage semantic transitions that are not supported by image boundaries, encouraging the model to align object responses with meaningful structural information. Since not all image edges correspond to semantic boundaries and not all semantic boundaries are perfectly represented by image gradients, this supervision is intentionally constrained by the weighting coefficient λ_{edge} to avoid excessive regularization and preserve the flexibility of the learning process.

The Window Local Consistency Loss originated as a lightweight alternative to traditional pairwise affinity regularization methods. Initial experiments with local affinity and pairwise consistency formulations introduced considerable computational overhead and implementation complexity while providing limited practical benefits. Consequently, a simplified formulation was adopted. The proposed loss encourages neighboring pixels to maintain similar semantic distributions, reinforcing local spatial coherence and suppressing isolated activations. Conceptually, its behavior resembles a soft morphological closure operation applied in feature space, promoting uniform expansion and encouraging the network to generate semantically coherent activation regions with higher confidence.

E. Morphological Post-processing Impact

The morphological closing operation produced an improvement of approximately 1% mIoU, highlighting its effectiveness as a post-processing refinement mechanism. This gain can be attributed to its ability to compensate for one of the main limitations of the dCRF-based supervision. Due to the conservative dCRF configuration adopted in this work, which uses a limited number of iterations to reduce error propagation, the generated masks frequently contain small disconnected regions and void areas inside object structures. Such artifacts are particularly common when semantically related regions exhibit significant appearance differences.

The closing operation acts as a structural correction mechanism by connecting fragmented activation regions and filling small gaps that remain unresolved after dCRF refinement. As a result, disconnected semantic islands are merged into more coherent object regions, improving mask completeness and spatial consistency. These findings suggest that lightweight morphological operations can effectively complement heuristic supervision, providing a low-cost mechanism for mitigating under-segmentation artifacts without introducing additional trainable parameters

F. Final Results (Our + RW + dCRF)

The final configuration, combining the proposed method with Random Walk and dCRF refinement, achieved 64.89% mIoU on the training split and 61.63% mIoU on the validation split and 54.13% on the test split. These results demonstrate that competitive performance can be achieved even under constrained computational resources and limited representational capacity. Although the proposed approach does not reach the performance of recent state-of-the-art methods, many of these approaches rely on substantially larger architectures, foundation models, or computational requirements that are often impractical for reproduction in resource-constrained environments.

From this perspective, the obtained results suggest that lightweight architectures remain a viable research direction for WSSS. The performance achieved by the proposed framework indicates that a significant portion of the localization gap can be mitigated through structural guidance, heuristic refinement, and spatial consistency regularization, without requiring large-scale backbones or external supervision sources. Furthermore, the relatively small performance gap compared to classical WSSS approaches suggests that there is still considerable opportunity for further exploration of efficient and resource-aware methods in this field.

Overall, the results support the hypothesis that improving semantic structure and spatial coherence may be as important as increasing representational capacity when designing lightweight WSSS systems. This finding reinforces the potential of efficient architectures as a practical alternative for scenarios where computational resources are limited.

G. Limitations and Complexity

Although the proposed method achieved competitive results, the final performance depends on a combination of multiple refinement stages, auxiliary losses, and heuristic supervision mechanisms. Consequently, the framework introduces additional complexity when compared to conventional classification-based WSSS approaches. From a practical perspective, this characteristic may limit its adoption in scenarios where simplicity and ease of deployment are prioritized.

However, the primary objective of this work was not to propose a production-ready solution, but rather to investigate the representational potential of lightweight architectures in the WSSS setting. The obtained results suggest that small networks possess a considerably higher localization capability than is commonly assumed, provided that appropriate structural guidance and refinement strategies are employed. In this sense, the proposed framework should be interpreted as evidence that lightweight architectures remain a promising research direction within WSSS.

Another important limitation concerns the experimental scope of the evaluation. All experiments were conducted exclusively on the PASCAL VOC 2012 dataset, which contains a relatively small number of semantic classes when compared to more challenging benchmarks. Consequently, the generalization capability of the proposed method on larger and more diverse datasets remains uncertain. As the number of classes

increases, the complexity of the localization problem may also increase, potentially reducing the effectiveness of the proposed refinement strategies.

Finally, although the method demonstrates that competitive performance can be achieved using a lightweight backbone, additional investigations are required to determine whether the proposed structural guidance mechanisms scale effectively to larger architectures and more complex datasets. These questions remain open for future work and represent promising directions for further research.

V. Conclusion

This work investigated the limitations and potential of lightweight architectures in the context of Weakly Supervised Semantic Segmentation. The proposed framework was specifically designed to address some of the most common challenges observed in small networks, including fragmented activations, weak spatial consistency, and limited representational capacity. The experimental results demonstrate that, when appropriately guided through structural distillation and heuristic refinement, lightweight models are capable of achieving competitive localization performance, particularly in the training environment. Nevertheless, the results also reveal clear limitations. Although the proposed method significantly improved the quality of the generated activation maps, the model exhibited a larger generalization gap than deeper architectures, as reflected by the difference between training and validation performance. This behavior suggests that lightweight networks may struggle to preserve the same level of semantic consistency and robustness achieved by larger backbones. However, these observations also suggest an alternative perspective on the role of lightweight architectures within WSSS. Rather than viewing them exclusively as end-to-end segmentation solutions, small networks may be more effectively employed as efficient pseudo-label generators. Their reduced computational requirements allow the incorporation of multiple refinement strategies and auxiliary supervision mechanisms while maintaining a relatively low memory footprint. In this context, the objective shifts from directly producing the final segmentation output to generating high-quality localization cues that can subsequently supervise dedicated semantic segmentation networks. Given that fully supervised semantic segmentation already provides highly effective and mature solutions, this direction may represent a promising alternative for lightweight WSSS research. Instead of focusing exclusively on closing the performance gap with larger architectures, future work may investigate how lightweight networks can be optimized for pseudo-label generation, serving as efficient localization modules within larger segmentation pipelines. Such an approach may offer a practical path toward achieving competitive results while maintaining substantially lower computational requirements.

References

[1] Radhakrishna Achanta et al. “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282. DOI: 10.1109/TPAMI.2012.120.

[2] Jiwoon Ahn and Suha Kwak. “Learning Pixel-level Semantic Affinity with Image-level Supervision for Weakly Supervised Semantic Segmentation”. In: *CoRR* abs/1803.10464 (2018). arXiv: 1803.10464. URL: http://arxiv.org/abs/1803.10464.

[3] Wonho Bae, Junhyug Noh, and Gunhee Kim. “Rethinking Class Activation Mapping for Weakly Supervised Object Localization”. In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV*. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 618–634. ISBN: 978-3-030-58554-9. DOI: 10.1007/978-3-030-58555-6_37. URL: https://doi.org/10.1007/978-3-030-58555-6_37.

[4] Tao Chen et al. *Spatial Structure Constraints for Weakly Supervised Semantic Segmentation*. 2024. arXiv: 2401.11122 [cs.CV]. URL: https://arxiv.org/abs/2401.11122.

[5] Tao Chen et al. “Spatial Structure Constraints for Weakly Supervised Semantic Segmentation”. In: *Trans. Img. Proc.* 33 (Jan. 2024), pp. 1136–1148. ISSN: 1057-7149. DOI: 10.1109/TIP.2024.3359041. URL: https://doi.org/10.1109/TIP.2024.3359041.

[6] Yann N. Dauphin et al. “Language Modeling with Gated Convolutional Networks”. In: *CoRR* abs/1612.08083 (2016). arXiv: 1612.08083. URL: http://arxiv.org/abs/1612.08083.

[7] M. Everingham et al. “The PASCAL Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338. DOI: 10.1007/s11263-009-0275-4.

[8] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[9] Bharath Hariharan et al. “Semantic Contours from Inverse Detectors”. In: *International Conference on Computer Vision (ICCV)*. 2011.

[10] Kaiming He, Jian Sun, and Xiaoou Tang. “Guided Image Filtering”. In: *European Conference on Computer Vision (ECCV)*. 2010, pp. 1–14.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML]. URL: https://arxiv.org/abs/1503.02531.

[12] Qibin Hou et al. “Self-Erasing Network for Integral Object Attention”. In: *NeurIPS*. 2018.

[13] Zilong Huang et al. “Weakly-Supervised Semantic Segmentation Network with Deep Seeded Region Growing”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7014–7023.

[14] Philipp Krähenbühl and Vladlen Koltun. “Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials”. In: *CoRR* abs/1210.5644 (2012). arXiv: 1210.5644. URL: http://arxiv.org/abs/1210.5644.

[15] Lovász László, L. Lov, and Of Erdos. “Random Walks on Graphs: A Survey”. In: Jan. 1996, pp. 1–46.

[16] Jinlong Li et al. "Expansion and Shrinkage of Localization for Weakly-Supervised Semantic Segmentation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 16037–16051. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/66738d21d3cddb8717ca52deff5a5546-Paper-Conference.pdf.

[17] Ci-Siang Lin et al. "SemPLeS: Semantic Prompt Learning for Weakly-Supervised Semantic Segmentation". In: *arXiv preprint arXiv:2401.11791* (2024).

[18] Yuqi Lin et al. *CLIP is Also an Efficient Segmenter: A Text-Driven Approach for Weakly Supervised Semantic Segmentation*. 2023. arXiv: 2212.09506 [cs.CV]. URL: https://arxiv.org/abs/2212.09506.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[20] Wenjie Luo et al. "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". In: *CoRR* abs/1701.04128 (2017). arXiv: 1701.04128. URL: http://arxiv.org/abs/1701.04128.

[21] Ilias Papadeas et al. "Real-Time Semantic Image Segmentation with Deep Learning for Autonomous Driving: A Survey". In: *Applied Sciences* 11.19 (2021). ISSN: 2076-3417. DOI: 10.3390/app11198802. URL: https://www.mdpi.com/2076-3417/11/19/8802.

[22] Lixiang Ru et al. "Token Contrast for Weakly-Supervised Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 3093–3102.

[23] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *CoRR* abs/1409.0575 (2014). arXiv: 1409.0575. URL: http://arxiv.org/abs/1409.0575.

[24] In-Jae Yu Sanghyun Jo. "Puzzle-CAM: Improved Localization Via Matching Partial And Full Features". In: *2021 IEEE International Conference on Image Processing (ICIP)* (Sept. 2021), pp. 639–643. DOI: 10.1109/icip42928.2021.9506058. URL: https://doi.org/10.1109/icip42928.2021.9506058.

[25] Hanno Scharr. "Optimale Operatoren in der digitalen Bildverarbeitung". German. Doctoral dissertation. Ruprecht-Karls-Universität Heidelberg, 2000.

[26] Krishna Kumar Singh and Yong Jae Lee. *Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-supervised Object and Action Localization*. 2017. arXiv: 1704.04232 [cs.CV]. URL: https://arxiv.org/abs/1704.04232.

[27] Weixuan Sun, Jing Zhang, and Nick Barnes. "Inferring the Class Conditional Response Map for Weakly Supervised Semantic Segmentation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2022, pp. 2878–2887.

[28] Hans Thisanke et al. *Semantic Segmentation using Vision Transformers: A survey*. 2023. arXiv: 2305.03273 [cs.CV]. URL: https://arxiv.org/abs/2305.03273.

[29] Maria Tzelepi and Anastasios Tefas. "Semantic Scene Segmentation for Robotics Applications". In: *CoRR* abs/2108.11128 (2021). arXiv: 2108.11128. URL: https://arxiv.org/abs/2108.11128.

[30] Jian Wang et al. "POT: Prototypical Optimal Transport for Weakly Supervised Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2025, pp. 15055–15064.

[31] Yude Wang et al. "Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[32] Changqian Yu et al. "BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation". In: *CoRR* abs/1808.00897 (2018). arXiv: 1808.00897. URL: http://arxiv.org/abs/1808.00897.