International Journal of Computer Vision

ISIM: Iterative Self-Improved Model for Weakly Supervised Segmentation

--Manuscript Draft--

Manuscript Number:	VISI-D-21-00725
Full Title:	ISIM: Iterative Self-Improved Model for Weakly Supervised Segmentation
Article Type:	Manuscript
Keywords:	Weakly Supervised Semantic Segmentation, Self SupervisedLearning, Class Activation Map, Pascal VOC 2012
Corresponding Author:	Cenk Bircanoğlu Bahçeşehir Üniversitesi: Bahcesehir Universitesi Istanbul, TURKEY
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Bahçeşehir Üniversitesi: Bahcesehir Universitesi
Corresponding Author's Secondary Institution:	
First Author:	Cenk Bircanoğlu
First Author Secondary Information:	
Order of Authors:	Cenk Bircanoğlu
	Nafiz Arica
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	Weakly Supervised Semantic Segmentation (WSSS) is a challenging task aiming to learn the segmentation labels from class-level labels. In the literature, exploiting the information obtained from Class Activation Maps (CAMs) is widely used for WSSS studies. However, as CAMs are obtained from a classification network, they are interested in the most discriminative parts of the objects, producing non-complete prior information for segmentation tasks. In this study, to obtain more coherent CAMs with segmentation labels, we propose a framework that employs an iterative approach in a modified encoder-decoder-based segmentation model, which simultaneously supports classification and segmentation tasks. As there are no ground-truth segmentation labels given, the same model also generates the pseudo-segmentation labels with the help of dense Conditional Random Fields (dCRF). As a result, the proposed framework becomes an iterative self-improved model. The experiments performed with DeepLabv3 and UNet models show a significant gain on the Pascal VOC12 dataset, and the DeepLabv3 application increases the current state-of-the-art metric by \%2.5. The implementation associated with the experiments can be found: https://github.com/cenkbircanoglu/isim}.

ISIM: Iterative Self-Improved Model for Weakly Supervised Segmentation

Cenk Bircanoglu[1,2*] and Nafiz Arica[2†]

[1*]Cognition, Adevinta, 24 rue des Jeûneurs, Paris, 75002, France.
[2]Computer Engineering Department, Bahcesehir University, Besiktas, Istanbul, 34353, Turkey.

*Corresponding author(s). E-mail(s):
cenk.bircanoglu@adevinta.com;
cenk.bircanoglu@bahcesehir.edu.tr;
Contributing authors: nafiz.arica@eng.bau.edu.tr;
†These authors contributed equally to this work.

Abstract

Weakly Supervised Semantic Segmentation (WSSS) is a challenging task aiming to learn the segmentation labels from class-level labels. In the literature, exploiting the information obtained from Class Activation Maps (CAMs) is widely used for WSSS studies. However, as CAMs are obtained from a classification network, they are interested in the most discriminative parts of the objects, producing non-complete prior information for segmentation tasks. In this study, to obtain more coherent CAMs with segmentation labels, we propose a framework that employs an iterative approach in a modified encoder-decoder-based segmentation model, which simultaneously supports classification and segmentation tasks. As there are no ground-truth segmentation labels given, the same model also generates the pseudo-segmentation labels with the help of dense Conditional Random Fields (dCRF). As a result, the proposed framework becomes an iterative self-improved model. The experiments performed with DeepLabv3 and UNet models show a significant gain on the Pascal VOC12 dataset, and the DeepLabv3 application increases the current state-of-the-art metric by %2.5. The implementation associated with the experiments can be found: https://github.com/cenkbircanoglu/isim.

Keywords: Weakly Supervised Semantic Segmentation, Self Supervised Learning, Class Activation Map, Pascal VOC 2012

1 Introduction

In the last decade, researchers have achieved outstanding results on the semantic segmentation problem, one of the fundamental tasks of Computer Vision. These achievements have been mainly reached within the fully supervised setting where the training dataset contains pixel-level class labels. However, accessing that kind of dataset for a specific problem is a compelling task, and it is well-known that collecting pixel-level class labels is costly and time-consuming. Consequently, this issue diverted the attention of the researchers working on semantic segmentation tasks to apply another approach called Weakly Supervised Semantic Segmentation (WSSS), where weaker labels that are less costly to collect, such as class labels, bounding boxes, scribbles, are employed to obtain pixel-level class labels.

Semantic Segmentation tasks in both fully supervised and weakly supervised settings have several challenges, such as cluttering of the background, occlusions, and intra-class variations. In addition to these challenges, there is another significant challenge in WSSS called the supervision gap. The supervision gap happens when the provided labels contain less information than the desired labels. For example, class labels contain information about the existence of the object classes in the image, but they do not hold any information about its shape, size, color, or any other attributes. In addition, there is no knowledge on if one or multiple instances of a class exist in an image. Therefore, this gap significantly increases the complexity of the segmentation task and has a notable impact on substantial performance differences between WSSS and Fully Supervised Semantic Segmentation (FSSS) solutions.

Most of the advanced studies in WSSS have employed a method called Class Activation Maps (CAMs) to obtain better segmentation labels in various ways. However, there is a significant issue with CAMs; they are interested in the most discriminative parts of the objects in the image. When CAMs are utilized to target segmentation labels, focusing on the most discriminating regions naturally concludes reaching non-complete and incorrect information about the objects and background. Most of the time, some parts of an object may be declared as background, or the other class instances may be missed in CAMs. Therefore, it is more reasonable to expect non-perfect results for the segmentation tasks with CAMs and accept that they require some modifications to reach more coherent results.

The above observations make us focus on boosting the CAMs results with the implicit information pieces of the images to narrow the performance gap between FSSS and WSSS methods. In this study, we propose a general framework having an iterative approach to reveal these implicit information pieces of the images in a multi-task learning setting. The primary task of the framework is to train against the classification and segmentation labels simultaneously within the same network by using only the class labels and images as a source. The training objective against the segmentation label is scraping the implicit information about the objects from the images with the help of segmentation loss.

The proposed framework is built on an encoder-decoder-based segmentation model that is modified to support learning classification and segmentation simultaneously. As there are no ground-truth-values for the segmentation task, the same model also generates the segmentation labels. After giving the task of generation of the segmentation labels, fundamentally, the proposed framework becomes a self-improved encoder-decoder model supporting multi-task learning. Both encoder and decoder parts of the network are an instance of Fully Convolutional Networks (FCN). The overall network learns the semantic segmentation labels while the CAMs are produced as outcomes of the training of the encoder part and classification branch. The intuition behind this idea is that the encoder should carry the latent features of the objects to make the decoder learn segmentation labels while optimizing itself in class labels training.

The main objective of the proposed framework can be summarized as pixel-wise propagation of the CAMs by adding a pixel-wise loss to the training phase and using the image content. Additionally, to reinforce this propagation, dense Conditional Random Fields (denseCRF, dCRF) are employed in the production phase of pixel-level labels from CAMs.

There are three main contributions of this study,

- The proposed method is a general framework that can employ any encoder-decoder-based segmentation model aiming to achieve CAMs which is more coherent with segmentation results.
- To the best of our knowledge, it is the first iterative self-improved segmentation architecture with CAMs and dCRF.
- It is illustrated with the experiments on PASCAL VOC 2012 that our framework with a version of DeepLab achieves the state-of-the-art performance with only image-level annotations.

The paper is organized as follows: after giving a detailed overview of related works on WSSS in section 2, the motivation of the study and proposed architecture are explained in the third section. Section 4 describes the implementation details, experimental results. Finally, in Section 5, we conclude the paper by discussing the proposed approach with experimental results.

2 Related Works

Segmentation is an essential ingredient of many systems built on image understanding. It plays a significant role in various applications from different domains such as autonomous driving, analysis of biomedical images, video surveillance, augmented reality, and fashion. Image segmentation algorithms designed to support these applications have led to numerous methods for an extensive history of image segmentation. Even though there are many studies in the literature using traditional Computer Vision and Machine Learning methods, such as thresholding [1], clustering [2], Conditional and Markov Random Fields [3], and others [4–6], our focus in this study is on the latter approaches that apply Deep Learning methods.

In recent years, FSSS applications have become state-of-the-art on the popular benchmark datasets using Deep Learning methods [7–13]. The main objective of FSSS is assigning the correct class label to each pixel of the image by observing the pixel-level ground truths. To fulfill this objective, the authors [14] introduced the Fully Convolutional Network (FCN) to obtain low-dimensional segmentation masks by converting the fully connected layers to convolutional layers in CNN architecture. Conditional Random Fields (CRFs) were added as a post-processing step by executing it on the results of FCN to enhance their performances [15]. Noh et al. [16] proposed a new model which consists of two parts, encoder, and decoder to obtain relatively high-dimensional segmentation masks. In the network, an instance of FCN takes charge of the encoding part, and the decoder part employs the transposed convolution layers. This study enlarges the dimension of the segmentation result without using interpolation directly on the output of an FCN. One representative example of the encoder-decoder-based methods called UNet [17] employed specific connections between the encoder and decoder blocks and symmetric expanding paths. In addition to these, there are two popular families in FSSS called DeepLab [13, 18, 19] and R-CNN based models [20]. DeepLab introduced the usage of the dilated convolutions in FCN. R-CNN based models introduced Region Proposal Network, which proposes the candidate regions and extracts Region of Interest (RoI) and RoIPool layer, which computes the features from the proposals. There are also other methods, which are popular because of their performance, elegance, or simplicity [21–25].

Likewise, the objective in WSSS is the designation of each pixel of the images with class labels. However, WSSS applications aim to fulfill this objective by learning from image-level ground-truths instead of pixel-level ground-truths. This critical difference significantly affects the literature of WSSS as it requires applying different techniques than FSSS.

In recent years, various studies have been published on achieving segmentation labels by employing weaker labels, and these studies can be broadly categorized into four main approaches [26];

- Expectation-Maximization (EM)
- Multiple Instance Learning (MIL)
- Self-Supervised Learning (SSL)
- Object Proposal Class Inference (OPCI)

In the following sub-sections, we introduce the above categories by giving the details of decisive studies in each of them. However, the studies on natural scene images take more of our attention even though there are WSSS applications [27–34] in other domains such as histopathology and satellite images. They are not considered in scope due to the differences in the characteristics of the solutions and datasets.

2.1 Expectation-Maximization

Theoretically, Expectation-Maximization is an iterative approach that contains two main tasks: optimizing a latent distribution across the image and

learning the segmentation masks from that latent distribution. In practice, the researchers follow these steps,

1. Generate the segmentation labels from images and class labels with a prior assumption
2. Train a Fully Convolutional Network to learn segmentation labels
3. Regenerate the segmentation labels from images, class labels, and the features learned by FCN
4. Iterate over steps 2 and 3

One of the early studies of this category, called CCNN [35], converted the original problem into a biconvex optimization problem and solved it by optimizing the convex latent distribution of fixed FCN outputs with several constraints while training the FCN against the fixed latent distribution. One representative study for this category called EM-Adapt [36] sets the expectation by adding a pixel-level bias to the FCN according to the given class labels, and in the maximization phase, it handles the expectation by optimizing the outputs of the FCN to target these pixel-level labels.

2.2 Multiple Instance Learning

Multiple Instance Learning (MIL) is one of the approaches of the supervised learning framework that deals with learning from the labels containing incomplete information. In practice, this method solves the WSSS task by having a model that aims to learn the class labels of the image and then assign each pixel with one of the given class labels for a given image and its class labels. In general, the WSSS application of MIL comes across as the training of a Convolutional Neural Network (CNN) with image-level loss and inferring the image locations according to the class level prediction. Class Activation Map (CAM), which is extensively used in the literature for several reasons and ways by other studies like this study, is an early approach of attention mechanism on CNN. Mainly, it was proposed to understand, explain, and explore CNN architectures better by achieving the activated areas of the image according to the model's predictions. In CAM study [37], there are two constraints, (i) CNN architecture should be an instance of a Fully Convolutional Network (FCN) (ii) There should be a Global Pooling layer before the classifier layer, which is a convolutional layer. Within these constraints, the calculation of the CAMs can be accessed by the dot product between the last two convolutional layers. As the calculation of CAMs requires some modifications in the well-known CNN architectures, the training step is mandatory for this initial study. There are other studies to extend the work of CAMs to improve the results or ease the process. For example, in Grad-CAM [38], the authors used guided backpropagation to reach CAMs without changing any part of the original network and training the network. WILDCAT [39] offered a more complex pooling layer to enrich the activated areas by following mainly the idea of the original CAM study [37]. The authors [40] converted the VGG-16 [41] network to an FCN by replacing the Dense layers with a 1x1 Convolutional layer also employed Global Average Pooling as the pooling layer and trained the modified network

to learn class-level annotations with a special MIL loss offered. In the end, the top predictions at each location were upsampled bilinearly to achieve the segmentation results. The authors [42] utilized a specific method called guided backpropagation to reach the pixel-level top-class predictions. First, with the help of guided-backpropagation, authors created the coarse class activations for multiple convolution layers of the network and aggregated them after unifying into one scale. Additionally, the process ended with the employing of dense-CRF. The authors [43] trained an FCN by targeting the class labels to generate prior foreground/background masks from the intermediate convolution layers; then, the authors used these masks to learn segmentation masks.

2.3 Self-Supervised Learning

Lately, Self-Supervised Learning has become the most popular approach with its diverse and performant examples for WSSS. The applications in this category mostly contain two steps; one to solve pretext tasks to use the outputs of it in the other task, the other one is a more complex downstream task which is most of the time the actual task. Heaps of times, the pipeline starts with learning pseudo semantic segmentation labels from the given class labels with one network and continues with a segmentation model to target the generated pseudo labels. In most studies, the first network aims to produce CAMs, and then the obtained CAMs are used in the training phase of the segmentation network.

In the literature, there are a vast amount of examples in this category as they reach good performances. SEC [44], which can be described as the best matching study to the given pipeline above, trained a VGG-16 on class labels to produce pseudo-pixel-level labels with the outcome of CAM and dense-CRF [45]. After the training of the first model, DeepLab-LargeFOV [15] was employed against the produced labels with a constraint loss as a segmentation model. Another study called MDC [46] followed the same steps with SEC with one modification; they assigned multi-dilated convolutional layers to the pseudo-ground-truths and generated the CAMs from these multi-dilated layers. AE-PSL [47] is another study that follows SEC's steps; with a significant novelty and a better performance, an adversarial erasing mechanism was integrated into the model to erase the activated regions of the previous steps to encourage the network to learn less discriminative parts. FickleNet [48] employed Grad-CAM instead of CAM to achieve a better performer segmentation model and added a center-fixed spatial dropout to the later convolutional layers to produce the pseudo-ground-truth labels to train an FCN against them. However, Grad-CAM also suffered from the same issue as it focuses on discriminative regions of the objects and the inputs of the second model are not precise.

Later, the researchers started to focus on how to take the CAM results as seed points to propagate from them. The idea behind this is that, hypothetically, adjacent pixels of the activated ones have more possibility to be a part of the same object than the others. Therefore, propagating from CAMs

to the adjacent regions can be efficient in producing better pseudo-ground-truths. With this in mind, the authors of DSRG [49] proposed a region-growing approach from CAMs to produce pixel-level labels. In the literature, there is one important study called Pixel-level Semantic Affinity (PSA) [50], which dramatically influences other studies, including ours, with its proposed architecture. The pipeline of PSA has three steps to follow and starts with a CNN training to generate class activations as in SEC. In the second step of the pipeline, instead of using the CAMs as ground-truth segmentation labels, the CAMs were used as seed points, and one more model was added to the pipeline, which performs a random walk to propagate from the seeds to achieve the pseudo-ground-truth for training a segmentation model. IRNet [51] can be called a version of PSA, which also targets a more complex task called instance-level semantic segmentation, with additional pixel-wise clustering to the random walk process. It performs the random walk from low-displacement field centroids in the CAM seeds until the class boundaries to produce pseudo-ground truths. Recently there are more advanced studies influenced by PSA were published. One of them, PuzzleCAM [52], improved the overall performance significantly with obtaining CAMs on image patches. The network to obtain the CAMs transformed to a siamese network without changing its aim. One branch of this siamese network was kept the same, but the other was organized to create tiles from the input image and merge the CAMs of these tiles. In addition to these, a reconstruction loss was proposed to regulate the results of both branches. There are also some studies to improve CAMs using saliency maps as additional information, such as [53, 54]. Even though most of the studies focus on improving the first model, CAMs, in the pipeline offered in PSA, one study [55] focused on the feature propagation frameworks and examined the Graph Convolutional Network (GCN) instead of AffinityNet to propagate from CAMs. To expand object activation regions like the others, DRS [56] used a different approach than others, and it suppressed the attention on discriminative regions and spread the attention to adjacent non-discriminative regions by generating dense localization maps.

2.4 Object Proposal Class Inference

Most of the applications in this category first employ a low-level feature extractor, mainly with one of the Traditional Computer Vision methods, and use the features to generate the pseudo-ground-truths. In the applications, researchers assembly methods from MIL and SSL to solve the WSSS problem. The authors [57] aggregate the superpixels and the features extracted from a CNN to generate the pseudo-ground-truths to train an FCN against them. In PRM [58], the low-level object proposals were secured by employing Multi-scale Combinatorial Grouping (MCG) [59], and an FCN was trained with the loss they proposed called peak stimulation loss. As the next step, they applied peak backpropagation to convert Class Response Map to Peak Response Map for each peak. By performing non-maximum suppression on the class labels, the object proposal extracted from PRM peaks. SPML [60] employed a Contrastive Learning

approach with its four types of relationships between pixels and segments in the feature space, capturing low-level image similarity, semantic annotation, co-occurrence, and feature affinity to access the segmentation masks.

3 Methodology

In most WSSS approaches, CAMs are used to obtain pseudo-segmentation labels as it is or as prior information for the subsequent stages. We also utilize CAMs [37] and propose a generic framework, which iteratively improves the semantic segmentation model in an encoder-decoder-based architecture. This new framework contains elements from all the WSSS categories mentioned in section 2. The proposed method is an example of EM due to its iterative approach, and the overall pipeline is an application of SSL. To generate the pseudo-segmentation labels, CAMs, which is an application of MIL approach, are employed. As CAMs are improved with the help of dCRF, it also contains the elements from OPCI.

In this section, after revealing our motivation, the proposed framework is described in detail.

3.1 Motivation

The utilization of CAMs to attain pseudo-segmentation labels is highly popular in the literature, and they are widely used in WSSS applications. These applications achieved relatively great results on the benchmark datasets; however, they have two main issues that can be improved from our perspective.

The first issue appears as a result of the standard process of CAMs generation. The typical procedure of obtaining CAMs contains a classification model optimized with a classification loss that mainly focuses on the most discriminative parts of the objects in the image that causes to ignore the less discriminating parts. Ignoring less discriminative regions or concentrating on some parts of the objects is reasonable for a classification model as its primary interest is the information of the object's existence in the image. However, when it comes to the segmentation models, every single pixel is equally essential. As a result of this contradiction between the nature of classification and segmentation, employing standard CAMs procedure to generate pseudo segmentation labels in WSSS applications can perform well; however, it can not achieve the optimal solution by itself. In light of this information, injecting a pixel-level loss to the training phase may help to achieve more coherent CAMs hypothetically by trying to equalize the importance of the pixels and learn from the image content. This pixel-level loss will help activate the less discriminative regions of the objects in the CAMs.

The other downside of using CAMs appears in the transition step of creating the pseudo-segmentation labels from them by applying a threshold. This thresholding process on the obtained CAMs has vital importance as the decision about each pixel belonging to an object or background is done here. This transition process mainly results in having non-complete, non-precise

results. The most identified problem is the wrong categorization of a pixel of the foreground object as a background pixel. Therefore, working on these pseudo-segmentation labels as ground truth significantly affects the next steps. Moreover, in previous applications, there is no link between the threshold value and the training steps of the classification network. Therefore, we propose an iterative process with a pixel-level loss by setting this threshold at the beginning of the training and implicitly optimizing CAMs and the classification network according to this threshold value.

These observations above bring us to the point that the CAM-based WSSS model should aim to learn from the whole body of objects and be boosted with a generated pixel-level loss. We think that it can be achieved by iteratively improving the CAMs with the help of a pixel-level segmentation loss using pseudo-ground-truths at each step of the training phase. In addition, deciding the threshold value at the beginning of the process and optimizing the model, CAMs can be crucial to have more coherent pseudo-segmentation labels. With this motivation, we propose a framework based on an encoder-decoder segmentation model that generates its segmentation labels from the classification branch with the help of denseCRF and thresholding process and learns these labels with the decoder part while learning the classification labels simultaneously and iterating over this process multiple times to improve itself at each iteration.

3.2 Proposed Framework

We follow the general pipeline of the PSA method [50] previously proposed and used by many other studies such as PuzzleCAM, CDA, and SEAM [52, 61, 62]. This section first presents the PSA pipeline and then describes the modifications in the pipeline as a proposed framework in detail.

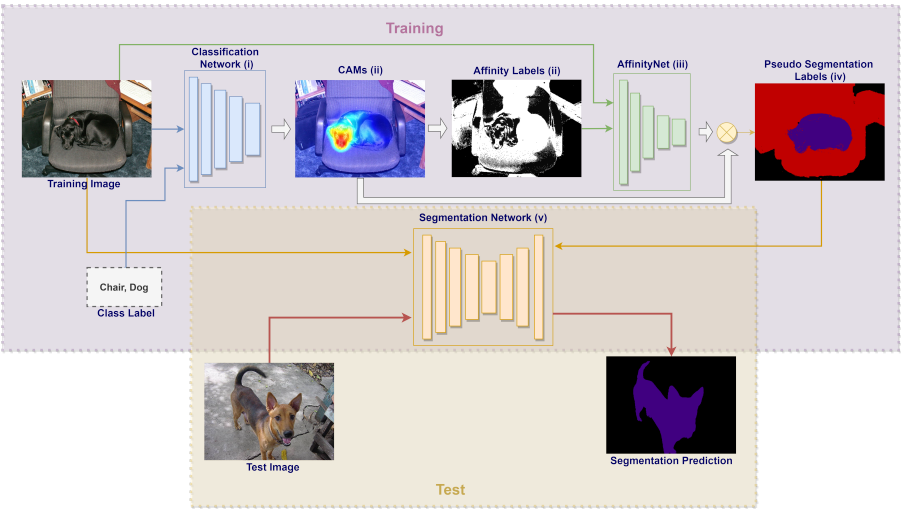

Fig. 1: End-to-End Pipeline of PSA

The PSA pipeline contains several steps, visualized in Figure 1 in the training phase. It is a representative application of a two-stage framework with a Self-Supervised Learning approach to achieve a semantic segmentation model using class labels. Generation of the pseudo-ground-truths in the first stage and targeting them in the second stage are the two stages of Self-Supervised approaches. In the first stage, after obtaining the CAMs, PSA offers a new model called AffinityNet, which propagates CAMs by using class-agnostic similarities to generate segmentation labels. In the second stage, the segmentation network is trained by taking these labels as pseudo-ground-truths. Briefly, PSA contains five steps in the training phase to fulfill this two-stage framework: (i) training of the classification model to obtain CAMs, (ii) generation of pixel-level affinity labels from CAMs with applying a threshold and denseCRF, (iii) training AffinityNet to learn class agnostic pixel-similarities targeting pixel-level affinity labels, (iv) generation of the segmentation labels from the combination of CAMs and predictions of AffinityNet, (v) training of the segmentation model using segmentation labels. In the beginning, the classification model takes the training images with their class labels and produces CAMs. Then these CAMs are processed with threshold and denseCRF to reach affinity labels. AffinityNet is trained with the images and their affinity labels, and then segmentation labels are generated by applying a random walk algorithm on CAMs based on the affinity matrix produced from AffinityNet. Finally, the segmentation network is trained by taking those segmentation labels as pseudo-ground-truths. In the test phase, Segmentation Network is used to obtain segmentation results in the test images.

In PSA-based approaches, adjustments are primarily made in the first step to improve its overall performance by reaching more accurate CAMs and affinity labels. As the problem of the CAMs mainly comes from the loss function and employing a classification network to obtain them, the most straightforward alternative is employing a segmentation network with a segmentation loss. However, as there are no ground truths for the segmentation task, this requires extra supplementary updates.

This study proposes an iterative approach to train a model that concurrently targets both the classification and segmentation tasks while generating pseudo-segmentation labels. The proposed model is an encoder-decoder-based segmentation model, which enhances the pseudo-segmentation labels and CAMs at each iteration.

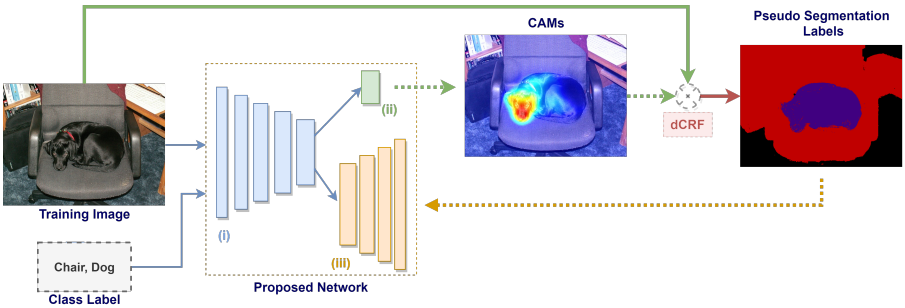

Fig. 2: Proposed Architecture: (i) Encoder network, (ii) Classification Branch, (iii) Decoder network

In the proposed encoder-decoder-based architecture, the tasks for WSSS learning are shared by different parts of the model. The encoder network learns the class-level annotations while discovering the latent features in CAMs to support the segmentation task. The decoder network aims the pixel-level annotations. The training steps of the proposed architecture are given in the pseudo-code Algorithm-1 and visualized in Figure 2. The overall training phase is an iterative approach containing three main steps. Even though it can be divided into three steps to make it easy to explain and formalize, in practice, the training is done in one step without stopping or starting it. Given the training set of images and their class labels, the first step is to train the encoder network, from which we can extract CAM of each training image. This step can be considered as the initialization of the whole model. In the second step, pseudo-segmentation labels are produced with the help of the denseCRF algorithm. In the last step, the whole model is retrained by taking the pseudo-segmentation labels as ground truth. Two different methods are employed for the pseudo-segmentation label generation and encoder-decoder training steps. In the first one, iteration continues until convergence, and in the second one,

fixed-size iterations are used. The performances of these two methods are given in the experimental results.

Algorithm 1 Training steps of the Proposed Architecture

```
procedure TRAIN(images, class_labels)
    x ← images
    z ← class_labels
    y ← null
    Model ← SegmentationModelwithPretrainedImageNet
    Model = TrainEncoder(Model, x, z)
    for i = 1 to N do
        y = GeneratePseudoGroundTruths(Model, x, z)
        Model = TrainEncoderDecoder(Model, x, y, z)
    end for
    return Model
end procedure
```

The only architectural change in the encoder-decoder-based segmentation model is that a new classification branch, marked as (ii) in Figure 2, is added after the encoder network with Global Average Pooling (GAP) and $(1x1)$ Convolutional layer to support classification and to produce CAMs. The proposed architecture is generic to practice with any encoder-decoder-based segmentation model.

In the training procedure, we utilize denseCRF to refine CAMs and practice the outputs as pseudo-ground-truths for the segmentation task. The utilization of denseCRF over CAMs and accessing the segmentation labels have two implicit benefits. First, to improve the CAMs, the power of denseCRF is embedded in the encoder network. Second, the information pieces of the images become more extractable for the model. After generating and using segmentation labels in the model, the problem becomes multi-task learning. With these changes, the encoder network is simultaneously optimized to learn the discriminative regions and the other regions related to the object with the help of classification and segmentation loss, respectively. Hypothetically, learning other features should positively affect the CAMs, and to challenge this hypothesis, we extensively operate experiments on the proposed framework.

However, in practice, the proposed framework has two potential issues which can affect the experiments significantly and make the training unstable. The first one is observed when each pixel of the pseudo segmentation masks is marked as the background. The second issue is that classification or segmentation loss suppresses the other one, and as a consequence, both or one of them starts to diverge. The first issue is suspected to be seen mostly when the classification network is not confident enough to distinguish between the classes. In addition that, discriminating between the classes is not enough as there is a threshold applied to CAMs. The model should have a higher confidence score

in the relevant pixels than the threshold value. Also, it is reported in Section 4 that the classification models are not 100% accurate on predictions. Therefore, there is a possibility of inconsistency in the pseudo-ground-truths, which will affect the overall training process. The decoder part of the model becomes confused when there are all background pixels as targets for the segmentation task. As a result, it will affect the encoder part and may make the classification loss diverge. The function given in Eq 2 is proposed to ease this issue. The aim of the proposed function is that when there is a segmentation mask that involves only background pixels, ignore that mask and do not calculate any loss for that specific image.

Supposed that y_i is the threshold applied CAM for the instance i and it is defined as,

$$y_i = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \tag{1}$$

the formulation of the segmentation loss for one instance becomes as below with our basic modifications,

$$\text{ModifiedCrossEntropyLoss}(\hat{y}, y) = \text{CrossEntropyLoss}(\hat{y} \cdot r_y, y) \tag{2}$$

where r_y is defined as ,

$$r_y = \begin{cases} 0, & \text{if } \sum_{j=0,k=0} a_{jk} = 0 \\ 1, & \text{else} \sum_{j=0,k=0} a_{jk} > 0 \end{cases} \tag{3}$$

On the second issue, instead of generating the pseudo-ground-truths in each epoch, we create them with a different frequency to give some room to the network to learn class and segmentation labels. On this point, we investigate and report the outcomes in the following sections.

To challenge the proposed framework and process, we concentrate on two common and well-known segmentation models, one from the DeepLab family, DeepLabv3, and the other is the UNet model. The original architectures of DeepLabv3 and UNet are modified according to the recommended method. The modified architectures are called DeepLabCAM and UNetCAM to refer in our study.

3.2.1 DeepLabCAM

DeepLab is one of the most popular image segmentation approaches in the literature. The networks belonging to this family achieved marvelous results on various tasks in fully supervised settings. In this study, one version of DeepLab called DeepLabV3 is preferred as a segmentation network to transform for a strong reason as the primary network. The literature shows that the dimension of the CAMs has an essential role in the results. Most of the applications in

the literature adopt a network from the ResNet family to obtain the CAMs. When the original ResNet is applied for that task, the output dimension of the CAMs becomes $16x16$, and with experiments, it is shown that using $32x32$ dimensioned CAMs perform better. Moreover, the dimension is doubled by changing the dilation rate of the last block of the ResNet family. In addition, one of the critical features of the DeepLab family is denoted as using the dilated convolutional layers. With these in mind, we add the classification branch after the fourth convolutional block of the encoder part of the DeepLabv3, as shown in Figure 3.

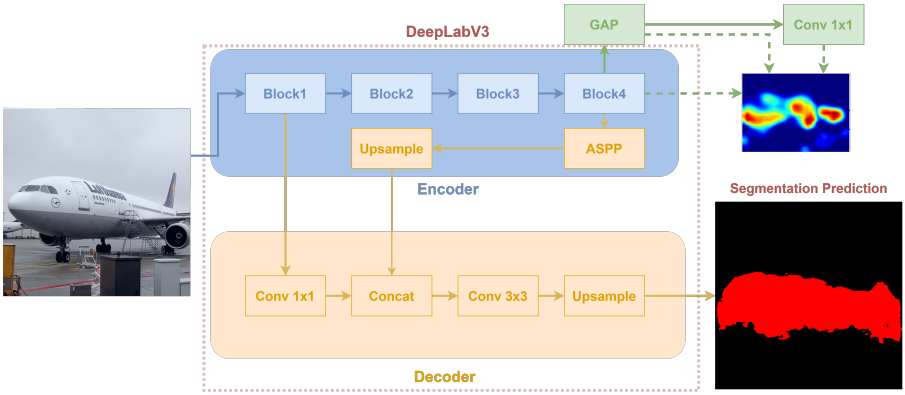

Fig. 3: DeepLabCAM Architecture

3.2.2 UNetCAM

UNet is also one of the most popular architectures in segmentation studies with unique characteristics and features. Two different forms of the UNet are modified, and they are executed in the experiments to prove once more the hypothesis that adding segmentation loss affects the CAMs positively.

In both UNetCam versions, ResNet50 is employed as the encoder part, and the decoder part is implemented concerning the constraints in the original UNet architecture. In UNet, there are skip connections between the encoder and the decoder blocks, and these skip connections are kept as they are. The difference between the UNetCAM versions comes from the implementation details of the upsampling layers. The first version contains non-learning upsampling layers, and the second one has learnable deconvolution layers. Also, in both networks, the classification branch is injected after the encoder network as proposed.

In the following section, the implementation details and the experimentation settings are given. The experimentation settings are organized to challenge the proposed framework from multiple perspectives such as genericity, stability, performance.

4 Experiments

The proposed framework is assessed under numerous settings from four different perspectives. First of all, to reveal its primary effect on the performance metric, the proposed framework is compared extensively with its original form, CAMs. As the second, two different segmentation models are transformed according to the proposed method, and they are executed to support the genericity of the approach. As a third perspective, we study the effects of the pseudo-segmentation regeneration frequency as the proposed training procedure possesses iterations over the network. The fourth collection of experiments are arranged related to the impact of the size of the images as image resolution has an apparent effect on the segmentation part.

After investigations introduced above are concluded on the proposed method, another experimentation set is practiced to challenge the other conjecture improving CAMs is the key to enhancing WSSS results. In that direction, the pipeline mentioned previously is applied by employing the CAM results of our proposed framework. All the details of the experiments and the results are given in the following parts of the section.

Additionally, another comparison is made between the end-to-end pipeline with the proposed framework and the state-of-the-art methods in the literature.

All the experiments are performed on PASCAL VOC 2012 dataset [63] with 20 class annotations and additional background annotation. The splits of the official dataset have 1464, 1449, and 1456 images for training, validation, and testing, respectively. We obey the conventional experimental protocol to make fair comparisons and take the additional annotations from SBD [64] to build an augmented training set. There are 10582 images in this extended training set, and it is named the trainaug set in this study. In addition to the images, we only use image-level classification labels during training phases.

Furthermore, segmentation results are assessed on the mean intersection over union (mIoU) metric. In addition to metrics, visualizations of the pseudo-segmentation labels are represented to investigate the effects of the proposed method. The dataset does not contain the ground-truths values for the test split, and to evaluate our results, we use the evaluation server maintained by the owners of the dataset. This evaluation server calculates the metrics on the test split, and they are used as it is. Even though there are ground-truth values for the validation dataset, the same method is preferred among the validation split, aiming for consistency in the results.

Additionally, all the experiments are implemented in PyTorch [65], and a computer that contains 4 RTX 2080 GPU, is used to operate the end-to-end pipeline.

4.1 Experimentations on the Proposed Framework

The proposed framework is challenged from several perspectives. Before presenting the experimental results, we introduce all the crucial functions and

parameters applied in these experiments by grouping them as preprocessing, generating pseudo-ground-truths, and training.

Preprocessing: In the training phase, the images are resized according to these rules as follows,

- If the longer side of the image is larger than 640, it is scaled down to 640.
- If the longer side of the image is smaller than 320, it is scaled up to 320.

Also, random horizontal flip and random cropping are applied to the images with the given resolution value, which is 320×320 if not otherwise declared, and images are normalized between 0 and 1. Also, the exact same cropping and scaling operations are followed for the pseudo-segmentation labels. In the CAMs prediction phase, four different scaled versions and their horizontally flipped images are used separately to generate CAMs, and the outputs are concatenated into one. 1, 0.5, 1.5, and 2 values are used as scales.

Although there are possible other data augmentation options, we employ the same steps with the previous studies to make the comparison fair.

Generation of Pseudo-ground-truths: Three steps are applied to generate pseudo-ground-truths, respectively (i) Producing CAMs as a result of encoder and classification branch, (ii) Applying a threshold value to decide which pixel involves foreground and background, and (iii) Executing dCRF on the thresholded CAMs. The threshold value is chosen as 0.3, although choosing another value for this threshold does not have a significant effect if they are not too big or too small, as the network feeds itself with the outcomes and reaches the balance with the result of the iterations. Also, this behavior can be pointed out as a significant difference from the previous studies as it is a new addition to the CAM process. Additionally, Unary potentials are used in dCRF calculations.

Training Process: To support the classification and the segmentation simultaneously, the model employs two different loss functions. The multi-label classification task calculates the loss with the Multi-Label Soft Margin loss function, and modified pixel-wise Cross-Entropy loss, which is described in Section 3.2, is adopted for segmentation tasks. Additionally, Poly optimizer, an extension of Stochastic Gradient Descent (SGD), is employed with momentum value 0.9 with different initial learning rates for the encoder, decoder, and classifier layers as 0.1, 0.01, 1, respectively.

The training is performed for 50 epochs in the default settings, and the first five epochs are only trained among classification labels. After the fifth epoch, the network is trained with both class and segmentation labels for each epoch. In default settings, segmentation labels are regenerated every ten epochs after the fifth epoch.

4.1.1 CAM vs. DeepLabCAM

To prove our hypothesis and evaluate our framework, an extensive set of experimental configurations is designed. First of all, to perceive the effects of the

proposed method, several variations of it and the corresponding implementation of CAM formed. In this direction, seven different backbones are employed for CAM and DeepLabCAM methods to compare them.

In the first experiment set, the classifier backbones of the DeepLabCAM and CAM are adjusted as the versions of the ResNet and ResNeSt network family. Dilated convolution is used in the last blocks of all networks to double the dimension of the CAMs.

Before presenting the segmentation results, in Table 1, classification scores are shared. Even though previous WSSS studies ignored these metrics, they are essential metrics as the accuracy of the classification model affects the CAM performance directly. If the model does not correctly predict the class, there will be no information about it to learn in the following steps of the pipeline. As shown from Table 1, models containing ResNeSt based backbones perform better in the classification task and, most probably, obtain better CAMs.

Table 1: Classification Accuracy

	ResNet			ResNeSt			
	50	101	152	50	101	200	269
train	95.86	96.35	96.58	97.46	97.96	98.49	0.9861
val	93.52	93.33	93.95	95.44	96.38	96.2	96.31

Table 2: mIoU on Pseudo Segmentation Labels

	ResNet			ResNeSt			
	50	101	152	50	101	200	269
CAM[1]	48.93	50.33	50.83	53.49	**57.78**	57.33	54.75
DeepLabCAM[1]	54.89	57.11	58.35	56.91	**63.09**	62.48	61.11

This table contains the mIoU metric for the trainaug split of PASCAL VOC 2012 dataset

The results of the CAM and DeepLabCAM versions are reported in Table 2. Moreover, significant performance improvement is apparent for each backbone from that table as the gap is more than 3% even in the closest form, rising higher than 7%.

The proposed method is proved to perform better than CAMs, with results presented in Table 2, at least for the modified versions of the DeepLabv3. The next set of experiments are done with another segmentation model called UNet.

4.1.2 CAM vs. UNetCAM

In the second set of configurations, another well-known segmentation model with different characteristics and internals than DeepLab is hired called UNet.There are two different versions of the UNet composed in this experiment set, and they are named UNetCAM Learnable and UNetCAM Non-Learnable, as described in Section 3.2.2. In the architecture of UNetCAM Learnable, the upsampling part of the decoder is constituted of transposed convolution layers, and for the other one, upsampling is done with bilinear interpolation. ResNet50 is selected to be the encoder part of the UNetCAM.

Table 3: CAM vs UNetCAM

	CAM	UNetCAM Learnable	UNetCAM Non-Learnable
train	48.93	54.30	53.75
val	47.59	53.01	52.41

All the experiments are done by using trainaug split

Table 3 presents the results of both UNetCAM networks and CAM, and it shows that UNetCAM enhances the results by at least 3%, which supports the experimentation results on DeepLabCAM.

Table 4: CAM vs UNetCAM

	CAM	UNetCAM Learnable	UNetCAM Non-Learnable
train	48.14	52.75	51.90
val	46.60	51.32	50.56

All the experiments are done by using train split

To go a little further, we change the data source with the current settings and use the train split instead of trainaug. Table 4 proves that the UNetCAM is performing better even with fewer training images.

One another exciting examination is arranged to interpret the effects of the usage of dCRF. For this purpose, in Table 5, the metrics are calculated after dCRF is applied to the CAMs. We are investigating if our proposed method embeds the dCRF to the network and the further application of the dCRF becomes somehow useless. The embedding idea matches with the results of Table 5, but it also shows that dCRF still has room to enhance the UNetCAM results.

Table 5: CAM + dCRF vs UNetCAM + dCRF

	CAM	UNetCAM Learnable	UNetCAM Non-Learnable
train	53.18	**59.90**	58.74
val	50.72	**57.11**	57.01

All the experiments are done by using trainaug split

4.1.3 Investigation on the Iteration Procedure

The conception of an iterative self-improved training procedure has its challenges, such as stabilization and optimization. In our proposal, the classification performance directly impacts the segmentation network as it uses the classification results. Furthermore, the segmentation network implicitly affects the features of the classifier network. If things go wrong for one task, they will go the same in the other, and the worsening will continue growing. The problem with optimization is that the overall network can be stuck in early steps without improving its performance.

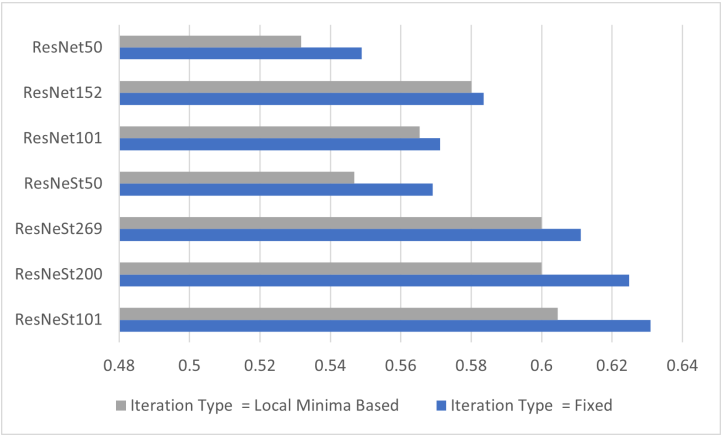

Fig. 4: mIoU According to Iteration Procedure

Intuitively, these two dilemmas may happen in two different ways, (i) when the classifier and the segmentation part could not find the room to optimize itself, (ii) when one of the tasks dominates the other in the training phase. To examine these potential problems, we set experimentations by changing the frequency of pseudo-ground-truth creation. The experiments focus on two main configurations; the first is waiting until the losses converge, and the second is fixing the generation frequency manually at the beginning of the training procedure.

Figure 4 represents the experimental results on the frequency of the pseudo-ground-truth generation, and from these results, it is clear that fixing the frequency performs better. Intuitively, this makes sense as this is an instance

of the expectation-maximization problem, and giving room to each task and training both in relaxed conditions performs better than strict rules.

4.1.4 Investigation on Image Resolutions

There are several differences between the applications of the segmentation and classification problems. One of them is that their input resolution preferences are different. Most of the classification architecture in their original forms accept the images which resolutions vary between 224 to 299. However, it increases to 512 or higher for the popular segmentation architectures with viable reasons. As the proposed network supports segmentation and classification simultaneously, studying the different input sizes makes sense to reveal the framework's stability and robustness.

In this part of the study, the experimentation reveals the effects of the input resolutions on the proposed method, and four input sizes are selected to run tests as 224x224, 320x320, 448x448, and 512x512.

Table 6: Experiments with Different Input Sizes

	256×256	320×320	448×448	512×512
CAM	57.52	57.78	55.34	54.94
DeepLabCAM	62.86	63.09	57.94	56.26

This table contains the mIoU metric between predicted segmentation labels and original segmentation labels

According to the reported Table 6, the proposed architecture performs better than the original CAM for each input size. In addition to that, when the input size is 320x320, it reaches its best.

These four experiments prove that the proposed network performs better than CAMs when employed as segmentation labels for the dataset. It engages the presented network results as affinity labels and executes the pipeline to achieve the final segmentation model.

4.1.5 Qualitative Results

Qualitative investigations can be done on the visualizations in Figure 5 to understand the effects of the segmentation loss and iterations more clearly. The first column of the figure shows the original image, and the other columns from left to right contain the pseudo-segmentation labels generated after epochs 5, 15, 25, 35, and 45, respectively. For the images in the first and the last rows, it is evident that the latter pseudo masks are more accurate than the previous ones. However, in the fourth row, it first begins to perform better, and in the last column, some pixels that are the part of the background start to be marked as a plane, which is evidence of becoming poorer. Even though it is hard to say the effects of the framework are in the positive or negative direction, it is

obvious from the images, one thing is clear, the proposed model has effects on the qualitative results.

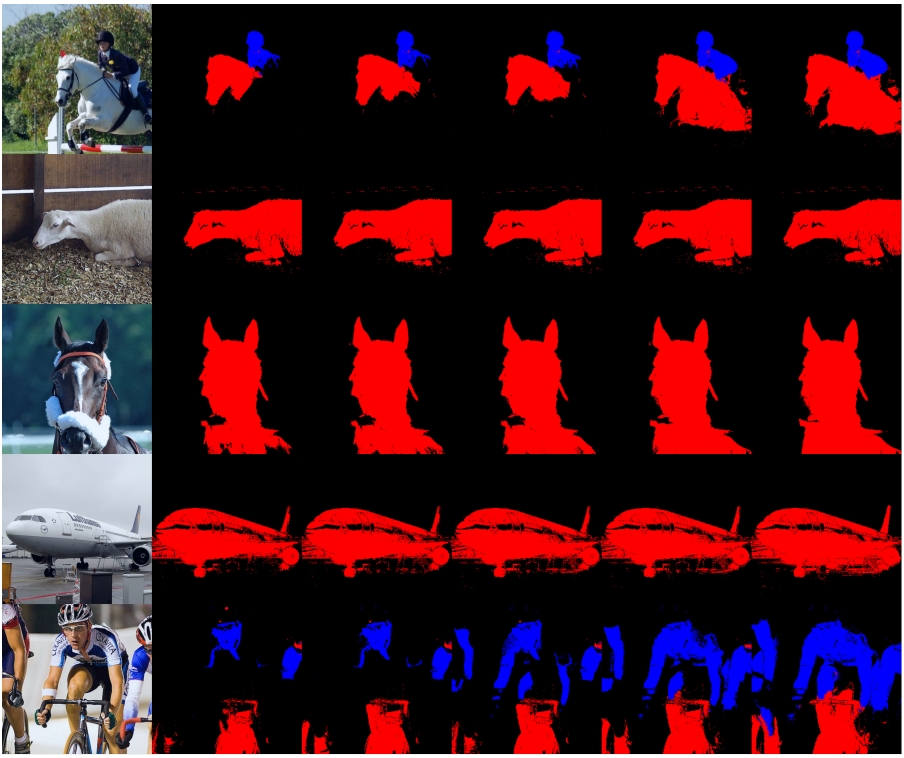

Fig. 5: Visualization of Pseudo Segmentation Labels

4.1.6 AffinityNet Training

AffinityNet is the network proposed in PSA to learn class-agnostic pixel-wise affinities. The model is trained on the images with a target called the affinity labels, generated from the denseCRF algorithm on CAMs of the DeepLab-CAM network. There are two threshold values to apply on CAMs to generate affinity labels more confidently, one for the background and the other for the foreground, with the values 0.1 and 0.3, respectively. The foreground threshold has the same value as the threshold used in DeepLabCAM to optimize according to it. We followed the same steps to train and evaluate the model as performed in the original study. In addition to the overall procedure, we only adjust the backbone instead of ResNet-38 in our experiments, and we use deeper versions and versions of ResNeSt architecture. As the optimizer, the Poly optimizer is chosen. Moreover, AffinityNet trained for three epochs with batch size 16 on 512×512 resolution images.

In the end-to-end pipeline, the backbones are held in common generally. It means that we operate the same backbone in DeepLabCAM, AffinityNet, and Segmentation model unless otherwise declared.

Table 7: mIoU on Pseudo Segmentation Labels

	ResNet			ResNeSt			
	50	101	152	50	101	200	269
CAM[1]	48.93	50.33	50.83	53.49	**57.78**	57.33	54.75
DeepLabCAM[1]	54.89	57.11	58.35	56.91	**63.09**	62.48	61.11
DeepLabCAM+RW[1]	74.25	76.40	78.54	75.65	76.78	77.47	**79.10**
DeepLabCAM+RW+CRF[1]	74.83	77.09	79.41	75.97	77.29	77.93	**79.64**

This table contains the mIoU metric between CAMs and original segmentation labels for the trainaug split of Pascal VOC2012

Table 7 presents the mIoU values of several backbones with different settings. The first two rows contain the results of CAM and DeepLabCAM, respectively. The third row shows the improvements done on DeepLabCAM results by applying Random Walk with AffinityNet predictions, and in addition to these, the fourth row also has one more step as employing denseCRF. The results in table 7 prove that AffinityNet performs better when it has more accurate affinity labels and also deeper networks achieve better performances.

The comparison between our proposed method with AffinityNet and previous studies with AffinityNet is handled in Section 4.1.8.

4.1.7 Segmentation model training

According to the previous studies, three different configurations are organized on the version of the DeepLab family to make fair comparisons. In the first set of experiments, DeepLabv2 architecture with the ResNet-101 backbone is exercised. The second set of experiments is done by upgrading the DeepLab version to 3 and keeping the same backbone. The last experiments are performed with again DeepLabv3 but with various backbones. For all of them, the input image resolution is 512×512, and the training is stopped after 50 epochs.

Table 8: mIoU on Segmentation Labels with DeepLabv3

	ResNet			ResNeSt			
	50	101	152	50	101	200	269
Val	67.87	70.51	72.28	69.77	73.52	**74.90**	74.76
Test	67.89	71.45	72.41	69.88	73.25	**74.98**	74.90

This table contains the mIoU metric between predicted segmentation labels and original segmentation labels

The results of the segmentation model with different backbones are reported in Table 8. In this Table, the backbone of all three models, DeepLab-CAM, AffinityNet, and DeepLabv3, is kept precisely the same in the end-to-end pipeline.

Table 9: mIoU on Segmentation Labels with DeepLabv3 (ResNet101)

	ResNet			ResNeSt			
	50	101	152	50	101	200	269
Val	69.09	70.51	71.54	70.39	73.55	**73.70**	73.37
Test	69.46	71.45	71.62	70.43	73.39	**74.50**	73.50

This table contains the mIoU metric between predicted segmentation labels and original segmentation labels

In Table 9, the backbone of the last segmentation model is kept the same as ResNet-101 as it is widely used in the previous experiments.

Table 10: mIoU on Segmentation Labels with DeepLabv2 (ResNet101)

	ResNet-50	ResNet-101	ResNeSt-200
Val	68.64	70.60	**72.60**
Test	68.83	70.38	**72.94**

This table contains the mIoU metric between predicted segmentation labels and original segmentation labels

In Table 10, to compare our results with the previous studies, we train the DeepLabv2 segmentation model with the pseudo-segmentation labels obtained from ResNet-50, ResNet-101 backbones.

4.1.8 Comparison with SOTA

In Table 11, we compare our best results with the previous studies, which are state-of-the-art at their time.

Table 11 can be interpreted like this, in overall performance, the best model is our proposed model with ResNeSt-269 backbone without using any additional data or saliency map with DeepLabv3 with a margin over %2.5. When the backbone is kept the same between the studies as ResNet-101, the best performance is observed in the EPS study, which uses extra saliency map data in the training phase. When there is a restriction to not use any additional data in the training phase, our model becomes the second-best one with %70.38 mIoU with a small margin on SPML which also does not use any additional information. In addition to these, EPS and DRS studies use saliency map data as the input in the other results in the table.

Table 11: Comparison of the results with literature

Method	Backbone	CAM[1]	Pseudo Masks[1]	Seg Masks[2]	Seg Masks[3]
PSA	ResNet-38	48	59,7	61,7	63,7
IRNet	ResNet-50	48,3	65,9	63,5	64,8
FickleNet	ResNet-101	-	-	64.9	65.3
ICD	ResNet-101	-	-	64.1	64.3
SEAM	ResNet-38	55.4	63.4	64.5	65.7
CDA	ResNet-38	58.4	66.4	66.1	66.8
PuzzleCAM	ResNeSt-101	61.85	72.46	66.9	67.7
WSGCN	ResNet-101	-	-	68.7	69.3
Ours[4]	ResNet-101	57,11	77.09	70.61	70.38
DRS v1[4]	ResNet-101	-	-	70.4	70.7
EPS[4]	ResNet-101	69.4	71.6	70.9	70.8
DRS v2[4]	ResNet-101	-	-	71.2	71.4
Ours[5]	ResNet-101	57,11	77.09	70.51	71.45
SPML[4]	ResNet-101	-	-	69.5	71.6
EPS[7]	ResNet-101	69.4	71.6	71.0	71.8
PuzzleCAM	ResNeSt-269	62.45	74.67	71.9	72.2
Ours[5]	ResNeSt-200	62.48	77.93	**72.60**	**72.94**
Ours[6]	ResNeSt-200	62.48	77.93	**74.90**	**74.98**

This table contains the mIoU metric between predicted segmentation labels and original segmentation labels

[1]The mIoU metric calculated on train split

[2]The mIoU metric calculated on val split

[3]The mIoU metric calculated on test split

[4]DeepLabv2 with backbone ResNet-101 used as Segmentation Model

[5]DeepLabv3 with backbone ResNet-101 used as Segmentation Model

[6]DeepLabv3 with backbone ResNeSt-200 used as Segmentation Model

[7]DeepLabv1 with backbone ResNet-101 used as Segmentation Model

5 Conclusion

In this study, we propose a novel framework for WSSS. It employs an iterative self-improved approach in an encoder-decoder-based segmentation model to obtain coherent CAMs with segmentation labels. The proposed framework contains various elements from the available approaches in the literature. Therefore, it tries to combine the strengths of those methods.

The extensive experiments show that the proposed framework is challenged and proven to improve the CAMs with the quantitative and qualitative results. The iterative approach is optimized with the help of a simple modification in the well-known loss function, Cross-Entropy, to achieve the goal. In addition to these, a specific implementation of the proposed framework achieves the state-of-the-art on Pascal VOC 2012 dataset.

As a possible future work, in theory, there is room to improve the loss function to stabilize more the training phase of the proposed framework. And also, combining AffinityNet with the proposed network will have a significant effect on the results as AffinityNet learns from pixel-level information from the

images, and adding more pixel-level loss to the proposed network may improve the results.

Supplementary information.

References

[1] Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics **9**(1), 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076

[2] Dhanachandra, N., Manglem, K., Chanu, Y.J.: Image segmentation using k -means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science **54**, 764–771 (2015). https://doi.org/10.1016/j.procs.2015.06.090. Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India

[3] Plath, N., Toussaint, M., Nakajima, S.: Multi-class image segmentation using conditional random fields and global classification. In: Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09, pp. 817–824. Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1553374.1553479. https://doi.org/10.1145/1553374.1553479

[4] Neubert, P., Protzel, P.: Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. 2014 22nd International Conference on Pattern Recognition, 996–1001 (2014)

[5] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**, 321–331 (2004)

[6] Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) Computer Vision – ECCV 2008, pp. 705–718. Springer, Berlin, Heidelberg (2008)

[7] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. arXiv preprint arXiv:2012.15840 (2020)

[8] Wu, T., Tang, S., Zhang, R., Cao, J., Zhang, Y.: Cgnet: A light-weight context guided network for semantic segmentation. IEEE Transactions on Image Processing **30**, 1169–1179 (2020)

Table 12: Class Based Results for the Best Model

background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
93.07	89.51	37.46	84.24	63.12	72.56	91.70	87.14	92.16	37.49	85.29
dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv monitor	
66.95	88.32	85.73	84.89	83.48	60.10	85.77	57.01	74.88	53.77	

[9] Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9799–9808 (2020)

[10] Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., Hu, H.: Disentangled Non-Local Neural Networks (2020)

[11] He, J., Deng, Z., Qiao, Y.: Dynamic multi-scale filters for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

[12] He, J., Deng, Z., Zhou, L., Wang, Y., Qiao, Y.: Adaptive pyramid context network for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

[13] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)

[14] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CoRR **abs/1411.4038** (2014) https://arxiv.org/abs/1411.4038

[15] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs (2016)

[16] Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. CoRR **abs/1505.04366** (2015) https://arxiv.org/abs/1505.04366

[17] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR **abs/1505.04597** (2015) https://arxiv.org/abs/1505.04597

[18] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. CoRR **abs/1606.00915** (2016) https://arxiv.org/abs/1606.00915

[19] Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR **abs/1706.05587** (2017) https://arxiv.org/abs/1706.05587

[20] He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. CoRR **abs/1703.06870** (2017) https://arxiv.org/abs/1703.06870

[21] Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. CoRR **abs/1612.03144** (2016) https://arxiv.org/abs/1612.03144

[22] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. CoRR **abs/1612.01105** (2016) https://arxiv.org/abs/1612.01105

[23] Ghiasi, G., Fowlkes, C.C.: Laplacian reconstruction and refinement for semantic segmentation. CoRR **abs/1605.02264** (2016) https://arxiv.org/abs/1605.02264

[24] Visin, F., Kastner, K., Courville, A.C., Bengio, Y., Matteucci, M., Cho, K.: Reseg: A recurrent neural network for object segmentation. CoRR **abs/1511.07053** (2015) https://arxiv.org/abs/1511.07053

[25] Chen, L., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. CoRR **abs/1511.03339** (2015) https://arxiv.org/abs/1511.03339

[26] Chan, L., Hosseini, M.S., Plataniotis, K.N.: A comprehensive analysis of weakly-supervised semantic segmentation in different image domains. CoRR **abs/1912.11186** (2019) https://arxiv.org/abs/1912.11186

[27] Aresta, G., Araújo, T., Kwok, S., Chennamsetty, S.S., P., M.S.K., Varghese, A., Marami, B., Prastawa, M., Chan, M., Donovan, M.J., Fernandez, G., Zeineh, J., Kohl, M., Walz, C., Ludwig, F., Braunewell, S., Baust, M., Vu, Q.D., To, M.N.N., Kim, E., Kwak, J.T., Galal, S., Sanchez-Freire, V., Brancati, N., Frucci, M., Riccio, D., Wang, Y., Sun, L., Ma, K., Fang, J., Koné, I., Boulmane, L., Campilho, A., Eloy, C., Polónia, A., Aguiar, P.: BACH: grand challenge on breast cancer histology images. CoRR **abs/1808.04277** (2018) https://arxiv.org/abs/1808.04277

[28] Chan, L., Hosseini, M., Rowsell, C., Plataniotis, K., Damaskinos, S.: Histosegnet: Semantic segmentation of histological tissue type in whole slide images. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10661–10670 (2019). https://doi.org/10.1109/ICCV.2019.01076

[29] Hosseini, M.S., Chan, L., Tse, G., Tang, M., Deng, J., Norouzi, S., Rowsell, C., Plataniotis, K.N., Damaskinos, S.: Atlas of digital pathology: A generalized hierarchical histological tissue type-annotated database for deep learning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11739–11748 (2019). https://doi.org/10.1109/CVPR.2019.01202

[30] Lin, H., Chen, H., Dou, Q., Wang, L., Qin, J., Heng, P.: Scannet: A fast and dense scanning framework for metastatic breast cancer detection

from whole-slide images. CoRR **abs/1707.09597** (2017) https://arxiv.org/abs/1707.09597

[31] Nivaggioli, A., Randrianarivo, H.: Weakly supervised semantic segmentation of satellite images. CoRR **abs/1904.03983** (2019) https://arxiv.org/abs/1904.03983

[32] Robinson, C., Hou, L., Malkin, K., Soobitsky, R., Czawlytko, J., Dilkina, B., Jojic, N.: Large scale high-resolution land cover mapping with multi-resolution data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019). http://openaccess.thecvf.com/content_CVPR_2019/html/Robinson_Large_Scale_High-Resolution_Land_Cover_Mapping_With_Multi-Resolution_Data_CVPR_2019_paper.html

[33] Seferbekov, S.S., Iglovikov, V.I., Buslaev, A.V., Shvets, A.A.: Feature pyramid network for multi-class land segmentation. CoRR **abs/1806.03510** (2018) https://arxiv.org/abs/1806.03510

[34] Wang, X., Chen, H., Gan, C., Lin, H., Dou, Q., Huang, Q., Cai, M., Heng, P.: Weakly supervised learning for whole slide lung cancer image classification. (2018)

[35] Pathak, D., Krähenbühl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. CoRR **abs/1506.03648** (2015) https://arxiv.org/abs/1506.03648

[36] Papandreou, G., Chen, L., Murphy, K., Yuille, A.L.: Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. CoRR **abs/1502.02734** (2015) https://arxiv.org/abs/1502.02734

[37] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. CoRR **abs/1512.04150** (2015) https://arxiv.org/abs/1512.04150

[38] Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. CoRR **abs/1610.02391** (2016) https://arxiv.org/abs/1610.02391

[39] Durand, T., Mordan, T., Thome, N., Cord, M.: Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5957–5966 (2017). https://doi.org/10.1109/CVPR.2017.631

[40] Pathak, D., Shelhamer, E., Long, J., Darrell, T.: Fully Convolutional

Multi-Class Multiple Instance Learning (2015)

[41] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2015)

[42] Shimoda, W., Yanai, K.: Distinct class-specific saliency maps for weakly supervised semantic segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016, pp. 218–234. Springer, Cham (2016)

[43] Saleh, F., Akbarian, M.S.A., Salzmann, M., Petersson, L., Gould, S., Alvarez, J.M.: Built-in foreground/background prior for weakly-supervised semantic segmentation. CoRR **abs/1609.00446** (2016) https://arxiv.org/abs/1609.00446

[44] Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation. CoRR **abs/1603.06098** (2016) https://arxiv.org/abs/1603.06098

[45] Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. CoRR **abs/1210.5644** (2012) https://arxiv.org/abs/1210.5644

[46] Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., Huang, T.S.: Revisiting dilated convolution: A simple approach for weakly- and semi- supervised semantic segmentation. CoRR **abs/1805.04574** (2018) https://arxiv.org/abs/1805.04574

[47] Wei, Y., Feng, J., Liang, X., Cheng, M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. CoRR **abs/1703.08448** (2017) https://arxiv.org/abs/1703.08448

[48] Lee, J., Kim, E., Lee, S., Lee, J., Yoon, S.: Ficklenet: Weakly and semi-supervised semantic image segmentation\\using stochastic inference. CoRR **abs/1902.10421** (2019) https://arxiv.org/abs/1902.10421

[49] Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7014–7023 (2018)

[50] Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. CoRR **abs/1803.10464** (2018) https://arxiv.org/abs/1803.10464

[51] Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance

segmentation with inter-pixel relations. CoRR **abs/1904.05044** (2019) https://arxiv.org/abs/1904.05044

[52] Jo, S., Yu, I.-J.: Puzzle-CAM: Improved localization via matching partial and full features (2021)

[53] Lee, S., Lee, M., Lee, J., Shim, H.: Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. CoRR **abs/2105.08965** (2021) https://arxiv.org/abs/2105.08965

[54] Yao, Q., Gong, X.: Saliency guided self-attention network for weakly-supervised semantic segmentation. CoRR **abs/1910.05475** (2019) https://arxiv.org/abs/1910.05475

[55] Pan, S., Lu, C., Lee, S., Peng, W.: Weakly-supervised image semantic segmentation using graph convolutional networks. CoRR **abs/2103.16762** (2021) https://arxiv.org/abs/2103.16762

[56] Kim, B., Han, S., Kim, J.: Discriminative region suppression for weakly-supervised semantic segmentation. CoRR **abs/2103.07246** (2021) https://arxiv.org/abs/2103.07246

[57] Kwak, S., Hong, S., Han, B.: Weakly supervised semantic segmentation using superpixel pooling network. In: AAAI (2017)

[58] Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., Jiao, J.: Weakly supervised instance segmentation using class peak response. CoRR **abs/1804.00880** (2018) https://arxiv.org/abs/1804.00880

[59] Pont-Tuset, J., Arbelaez, P., Barron, J.T., Marqués, F., Malik, J.: Multi-scale combinatorial grouping for image segmentation and object proposal generation. CoRR **abs/1503.00848** (2015) https://arxiv.org/abs/1503.00848

[60] Ke, T., Hwang, J., Yu, S.X.: Universal weakly supervised segmentation by pixel-to-segment contrastive learning. CoRR **abs/2105.00957** (2021) https://arxiv.org/abs/2105.00957

[61] Su, Y., Sun, R., Lin, G., Wu, Q.: Context decoupling augmentation for weakly supervised semantic segmentation. CoRR **abs/2103.01795** (2021) https://arxiv.org/abs/2103.01795

[62] Wang, Y., Zhang, J., Kan, M., Shan, S., Chen, X.: Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. CoRR **abs/2004.04581** (2020) https://arxiv.org/abs/2004.04581

[63] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.:

The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010)

[64] Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: International Conference on Computer Vision (ICCV) (2011)

[65] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc., ??? (2019). http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf