Single Stage Weakly Supervised Semantic Segmentation of Complex Scenes

Peri Akiva
Rutgers University
peri.akiva@rutgers.edu

Kristin Dana
Rutgers University
kristin.dana@rutgers.edu

Abstract

The costly process of obtaining semantic segmentation labels has driven research towards weakly supervised semantic segmentation (WSSS) methods, using only image-level, point, or box labels. Such annotations introduce limitations and challenges that results in overly-tuned methods specialized in specific domains or scene types. The over reliance of image-level based methods on generation of high quality class activation maps (CAMs) results in limited applicable dataset complexity range, mostly focusing on object centric scenes. Additionally, the lack of dense annotations requires methods to increase network complexity to obtain additional semantic information, often done through multiple stages of training and refinement. Here, we present a single-stage approach generalizable to a wide range of dataset complexities, that is trainable from scratch, without any dependency on pre-trained backbones, classification, or separate refinement tasks. We utilize point annotations to generate reliable, on-the-fly pseudo-masks through refined and spatially filtered features. We are to demonstrate SOTA performance on benchmark datasets (PascalVOC 2012), as well as significantly outperform other SOTA WSSS methods on recent real-world datasets (CRAID, CityPersons, IAD, ADE20K, CityScapes) with up to 28.1% and 22.6% performance boosts compared to our single-stage and multi-stage baselines respectively.

1. Introduction

The fundamental computer vision task of semantic segmentation seeks to assign class labels to specific pixels in a given input image. The rapid development of deep learning methods has resulted in significant progress in performance [1, 2], stability [3], and accessibility [4, 5] of semantic segmentation algorithms, often seen in real world applications such as autonomous vehicles [6], precision agriculture [7], medical diagnosis [8], image restoration and editing [9], sports [10], and remote sensing [11, 12]. While such algorithms provide insightful information about the scene, it requires large amounts of pixel-wise labeled data [13, 14], which is often expensive and time consuming to collect [15]. To alleviate this requirement, recent efforts have focused on weakly supervised semantic segmentation (WSSS) using image-level [16, 17, 18, 19], center point [15, 20], scribbles [21, 22], or bounding box [23] labels. The balance between cost and utilization is essential in determining what kind of annotations are needed. Image-level annotations may be cheap to produce, but are not generalizable to complex scenes and require more complex, multi-stage networks not practical for real world applications. On the other hand, pixel-wise annotations may be too expensive and time consuming as an up-front cost.

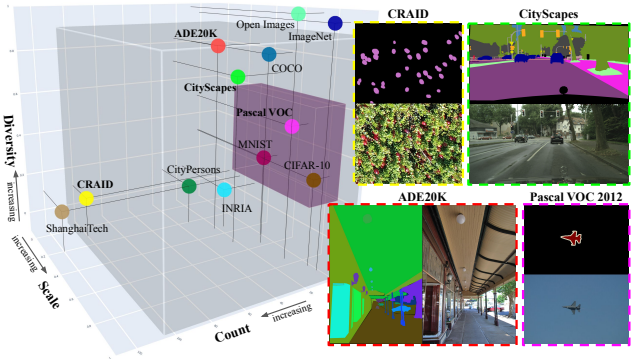

Figure 1: Scene complexity qualitative illustration of common datasets with respect to complexity parameters *count*, *scale*, and *diversity*. Current SOTA has largely only explored datasets within the purple highlighted region, mainly reporting on the object-centric dataset Pascal VOC 2012. This works aims to expand the applicable scene complexity ranges (highlighted in gray) for WSSS. Best viewed in color and zoomed.

In this work, we define the complexity of a scene based on three main parameters: *diversity*, *scale*, and *count*. *Diversity* measures the homogeneity of the dataset, *e.g.* a blood cell dataset is homogeneous while ImageNet [24] is more diverse. *Scale* measures the average area objects occupy in images, and *count* measures the average frequency objects appear in a given image. We consider a scene complex when it is at extreme ends of ranges of two parameters or more. For example, complex scenes may have many

small instances of the same object, or they may have many different objects. A visualization of those parameters with example datasets is shown in Fig. 1.

Current SOTA methods address different levels of complexity with different approaches. The more complex tasks of high count or diverse segmentation require points, bounding boxes, or pixel-wise annotations [25, 26, 27, 28], while object centric segmentation [29, 30] can be achieved with image-level labels alone. This results in overly-tuned methods that suffer significant performance degradation on scenes outside of their distributional scene complexity. More specifically, image-level based methods depend on class activation maps (CAMs) [31, 32, 33] to obtain pixel-wise coverage and localization of objects in the scene. CAMs are features generated from a pre-training classification procedure, and are often noisy, and only cover the most discriminating parts of objects under ideal complexity conditions. However, once object counts increase, object scale decreases, and/or class diversity changes, the quality of CAMs are significantly degraded, often beyond utility [16]. Even when changing only one parameter, such as with images that have few (or binary) object labels (low diversity), the classification pre-training procedure becomes too easy, and while the resulting coarse CAMs (from image-level labels) are enough for classification, they are insufficient for localization and/or segmentation and therefore produce poor pseudo-masks. For that reason, SOTA image-level based methods have only focused on datasets within the highlighted complexity range illustrated in Fig. 1, without little focus on more complex semantic segmentation datasets such as ADE20K [13] or CityScapes [34]. On the other hand, current methods that use point supervision employ strong implicit assumptions on object scale [35, 36] or dataset diversity [20, 37]. Here, we propose a method that significantly increases the applicable dataset complexity range without making strong assumptions.

Another element that increases our method's versatility and applicability is its single-stage, end-to-end train-ability. The importance of single stage has been recently recognized with developments in segmentation [19, 38, 39], pose estimation [40, 41], object detection [42, 43, 44, 45, 46], image retrieval [47], and image generation [48]. To understand why this is significant for our task, we consider the commonly featured image-level based WSSS methods, pipeline, and resulting weaknesses. Image-level based methods often require computationally expensive stages such as training of multiple networks, region proposal generation, and refinements. These methods are referred to as *multi-stage* WSSS since they include multiple stages of training and evaluation before performing final inference. Such *multi-stage* requirements make adaptation to new datasets more difficult. Any change in data distribution requires significant effort, and approaches such as online learning [45, 49] become impractical to adapt. Additionally, the entirety of the method requires all elements in the multi-stage pipeline to work, and individual module failures may affect the entire system. For example, the failure of pre-training stages (*i.e.* training a classifier for CAM generation) on low diversity scenes would cause the entire system to under-perform even if the refinement module provides utility under those complexity parameters. In contrast, our method eliminates the dependency on prior pre-training and refinement tasks, achieving competitive performance in a *single-stage* approach.

This work presents a versatile *single-stage* WSSS approach applicable for datasets in large range of complexities, independent of object counts, scale, or dataset diversity. Our method can be trained in a single stage, without separate pre-training, refinement, or evaluation stages, making it flexible and extendable in application driven systems. We choose to use point annotations since, while only costing an additional 2 seconds per image in annotation time (20.0 sec/image compared to 22.1 sec/image on average for PascalVOC [15]), it provides spatial information essential for correctly localizing and segmenting objects in complex scenes. The method comprises two main novel contributions. First, a point generator module that transforms few points to many points using a basic intuition: Given a user-defined object point, the task of sampling *another* object point is not so hard. In fact, classical work on random walks in image segmentation can be re-formulated for this problem. Our approach is a point augmentation by iteratively scattering the original points by small affine perturbations followed by random walks. The point-set obtained by this iterative scatter-then-walk procedure is termed *point blot*, analogous to ink blot. The resulting point blot has significantly more utility compared to the original point-clicks, and is entirely deterministic (no training required). The second contribution in our framework is the *expanding distance fields*, a new instantiation of the classic distance fields [50], which acts as a spatial attention filter to ensure captured features are spatially accurate. When considering early training iterations of an un-trained network, outputs are expected to be noisy and unstable, producing unreliable pseudo-masks. To mitigate such errors, our expanding distance fields module filters spatially inaccurate feature activations which stabilizes training by preventing accumulation of bias in generated pseudo-masks. Lastly, we present our adaptation of pixel adaptive convolution layers to deterministically refine features such that local consistency is preserved in output features and subsequent pseudo-masks.

2. Related Work

2.1. Semantic Segmentation

Semantic segmentation is a dense image prediction task that predicts class labels for every pixel in a given image. The quick progress in deep learning and convolutional

neural networks [51] has powered the development of the fully convolutional network (FCN) [2], which is the basis to many current SOTA semantic segmentation methods [1, 3, 52, 53]. Typical design of semantic segmentation networks utilize encoder-decoder architectures, in which deep features are learned, and up-sampled to match the input image size. More recent work improve this base design by incorporating skip connections [3], contextual information [53], self-attention mechanisms [54], enlarged receptive fields [52], pyramid pooling [1, 55], and refiner networks [56]. While those networks often provide SOTA performance, they still require expensive, fully supervised ground truth.

2.2. Class Activation Maps and Region Proposals

The activated neurons of a deep learning network with response to an input image are called class activation maps (referred as CAMs or attention maps) [31]. They represent regions the network finds most distinctive for a given class label. Initial work leveraging CAMs were used for object localization [31, 32, 57, 58, 59] and network interpretability [33], but were recently adopted for semantic and instance segmentation tasks [16, 17, 18, 60]. Most approaches utilize CAMs, region proposals, or auxiliary data to generate pseudo-masks for segmentation methods. Since CAMs tend to be noisy and irregular in shape, much focus in the WSSS domain has been devoted towards refining outputs to improve CAM coverage accuracy and consistency [61, 62].

2.3. Weakly Supervised Semantic Segmentation

The majority of work done in the WSSS domain is accomplished in a multi-step process: train a classification or segmentation network, apply the network on the training set to extract CAMs, which are then refined and thresholded before used to train a separate segmentation network. Early work such as BoxSup [23] utilize bounding boxes to update pre-defined region proposals to generate ground truth masks for the training set. AffinityNet [18] leverages image level labels to generate affinity labels obtained through selection of high confidence points on amplified CAMs. Similarly, PRM (Peak Response Map) [16] back-propagates through local extrema points in attention maps to generate instance-wise pseudo-masks. Additional methods [17, 29, 60, 61, 63] follow a similar *multi-stage* approach using image level labels for pseudo-mask generation. As mentioned before, and as presented by [16], image-level based CAMs significantly degrade with respect to scene complexity, often beyond utility to any downstream tasks. For that reason, image-level based WSSS methods have focused on low complexity scenes.

The weaknesses of image-level driven WSSS methods were also recognized by recent point [35, 64, 65], box [66], and scribble [67] based methods. [35] segments buildings from overhead images in a single stage, but requires both building center point annotations and estimated radius around the point that captures the building. [65] employs point annotations and four separate networks structured as a teacher-student architecture, with two teacher networks and two student networks. Apart from the high network complexity, each network requires separate training and refinement before generating pseudo-masks.

2.4. Single Stage WSSS Learning

Single stage WSSS methods [19, 29, 30] are less common due to the challenge of implicitly obtaining reliable spatial and contextual information from weak labels. Triple-S [20] uses point supervision and shape priors as spatial and contextual cues for the network. However, the use of shape priors is highly restrictive, and explicitly provides spatial and contextual information to the network, making the method too task specific. In contrast, Araslanov *et al*. [19] train a segmentation-aware classification network using normalized global weighted pooling (nGWP), iterative mask refinement, and focal mask penalty. Normalized global weighted pooling allows concurrent classification and segmentation training, while the output mask prediction is iteratively refined using Pixel Adaptive Convolution (PAC) layers introduced in [68]. While [19] shows significant improvement in single stage WSSS, the method requires a pre-trained backbone to achieve good performance, and as shown by our experiments, is not generalizable to more complex datasets. Pre-trained backbones (trained on the benchmark dataset or similar) are essentially trained classification networks, similar to what is used in multi-stage methods. The utility of pre-trained weights removes biases and randomness present during initial training steps, allowing for superior pseudo-mask generation, "skipping" the challenging stage of generating pseudo-masks in early iterations. Without the pre-trained weights, generated pseudo-masks would be significantly worse, degrading segmentation performance, and propagating bias in the learning process. Generally, as seen in [32, 33], a trained classification network provides "free" localization of the objects by locating peaks in class activation maps. Such localization would not be available unless the backbone is pre-trained, or trained first. In contrary, our method is generalizable to any dataset, and is trainable from scratch. Similar to [19], we also utilize Pixel Adaptive Convolution layers [68] for feature refinement, and subsequent pseudo-mask generation.

3. Pseudo-Masks from Points

The motivation behind our method is to obtain reliable, on-the-fly pseudo-masks from initial points to train a semantic segmentation network. Intuitively, the better the ground truth labels, the better the network's performance. Pseudo-masks are typically obtained through some thresh-

olding of high confidence (high activation) features. When training from scratch, such features tend to be noisy, and directly thresholding such features will generate poor pseudo-masks which will result in sub-optimal training and performance. We address that challenge by using the Expanding Distance Fields module (section 3.2), which filters wrongly activated regions, and captures and amplifies correctly activated regions. It also introduces a new aggregation approach and expansion mechanism that alleviates overfitting to features around ground truth points. We also employ a Point Blot Generator (section 3.3) and its point blot output to provide superior utility compared to points alone, capturing additional locally available contextual information, and accelerating training progress. As seen in Figure 3, we incorporate a feature refinement network (section 3.1) to work in tandem with our Expanding Distance Fields to produce intermediate pseudo-masks, which are superimposed with point blots to make the final pseudo-masks for supervision.

3.1. Pixel Adaptive Convolution Refinement Network

We construct our Pixel Adaptive Convolution Refinement Network (PAC Refinement Network or PAC Refiner) using a sequence of pixel adaptive convolution layers introduced in [68]. PAC layers allow for dynamic modification of kernel weights based on some underlying conditions, and are commonly used in feature refinement work [19, 69, 70, 71, 72, 73] with user-defined kernel functions. Here, we use PAC layers to dampen activated regions in the output features that are not locally consistent. Our PAC Refinement Network considers the local similarity between a given pixel and its neighbors measured by the euclidean distance. We seek to amplify local features when a pixel is similar to its neighboring pixels, and dampen local features when a pixel is dissimilar to its neighboring pixels. We use local standard deviation in color space and mean in feature space to normalize kernel weights and avoid over amplification. Consider image input $X \in \mathcal{R}^{3\times H\times W}$, network output $\tilde{x} \in \mathcal{R}^{C\times H\times W}$, and corresponding softmax $\tilde{x}_s \in \mathcal{R}^{C\times H\times W}$, where C,H,W represent the number of classes, height, and width. A single pixel adaptive convolution layer forward pass generates a scalar matrix $M \in \mathcal{R}^{C\times H\times W}$ with local elements in M determined by the adaptive convolutional kernel function:

$$k_{c,i,j} = -\frac{(X_{i,j} - X_{l,n})^2}{\sigma_{i,j}} \mu_{c,i,j}, \tag{1}$$

where (i, j) correspond to the current location of kernel k for class c, (l, n) represent all neighboring pixels of (i, j) within the kernel, $\sigma_{i,j}$ is the standard deviation of the current kernel region in X, and $\mu_{c,i,j}$ is the mean value of the current kernel region in $\tilde{x}_s$ for class c. This kernel function ensures high kernel values when the center pixel is similar to its neighbors (amplifier), and low kernel values when the center pixel is dissimilar to its neighbors (dampener). Then, we further dampen the kernel weights when standard deviation is large or activation is low as a normalization mechanism. Note that the mean of $\tilde{x}_s$ cannot amplify the features, but only dampen them. This effect can be seen in Fig. 5, where the activated features around the bird are locally inconsistent with the corresponding color space input, and therefore dampened. A visualization of this process is also shown in Fig. 2. For L layer network, we generate L scalar matrices, each constructed with different PAC parameters (kernel sizes, dilations, and strides). We then sequentially multiply (per class) the L matrices with our output features to obtain the refined features $\tilde{x}_{refined}$:

$$\tilde{x}_{refined} = M^{L-1} M^{L-2} \cdots M^0 \tilde{x}. \tag{2}$$

Note that since kernel weights are functions of local color and feature statistical representation (standard deviation and mean), the refined features are the output of a single forward pass of the network. There are no learned weights in this operation, making it computationally inexpensive.

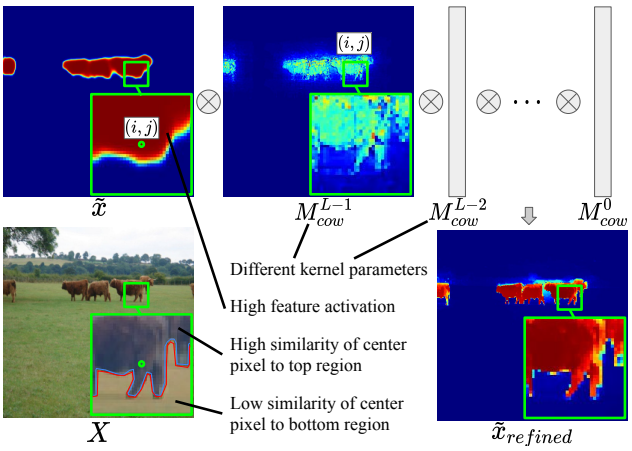

Figure 2: **PAC Refinement Network** example and intuition. Given features $\tilde{x}$ and input image X, our refinement network seeks to generate a scalar matrix M_c for class c (class "cow" in the example above) such that locally contained high similarity regions are amplified, and low similarity regions are dampened. In this example, we select a region over an edge which contains high activation over locally inconsistent region in color-space (brown cow and green grass). The normalized euclidean distance kernel with various kernel parameters (kernel size, dilation, stride) determines the dampening or amplification effect over that region for a given class. Best viewed in color and zoom.

3.2. Expanding Distance Fields

Expanding Distance Fields aim to impose global consistency and correct localization in the refined feature space (obtained from 3.1) by leveraging background, if available,

and object point annotations (also referred to as negative and positive points respectively) to generate distance fields (section 3.2.1). These distance fields are then updated by our expansion mechanism (section 3.2.2) which allows the distance field to incrementally incorporate more refined features into the final output.

3.2.1 Distance Field Aggregation

The use of distance fields [50], which converts a masked image (Y_p) to a distance-based grayscale with intensities depicting its nearest similarly labeled neighbor, is common in interactive segmentation methods [74, 75, 76], where it is used as auxiliary data produced from user inputs such as points and scribbles. Here, we use it as a point-guided filter to enforce object localization consistency in the refined features and subsequent generated pseudo-masks. Our usage of distance fields filters is essential in stabilizing training in early iterations, during which output features lack sound structure and localization to make reliable pseudo-masks. Distance fields are computed by taking the minimum Euclidean distance between a given point and the rest of same class points present in the scene. Given image $X \in \mathcal{R}^{H\times W\times 3}$ and ground truth points $Y_p \in \mathcal{R}^{H\times W\times 1}$, where $Y_p(i,j) \in \{0, 1, \cdots, \mathcal{C}\}$, we use Y_p to obtain class-wise distance fields $\mathcal{D} \in \mathcal{R}^{C\times H\times W}$, where C is the number of classes. For example, to generate a distance field $\mathcal{D}_c$ for some class c, we compute the value of $\mathcal{D}(c,i,j)$ at location (i,j) with respect to all other points, $\{(a,b) : (a,b) \in Y_p,\ (a,b) \neq (i,j)\}$, using

$$\mathcal{D}(c,i,j|Y_p) = \min_{\forall (a,b)\in Y_p, (a,b)\neq (i,j)} dist(p_{c,i,j}, p_{c,a,b}) \quad (3)$$

where p_c is a point in Y_p that belongs to class c, and $dist$ is the euclidean distance. We repeat this for all classes in the image to obtain $D \in \mathcal{R}^{C\times H\times W}$, including for background points. If background points are not provided and/or cannot be obtained, we consider all other points that don't belong to class c as background for $\mathcal{D}_{\neg c}$. Typically, such distance fields are concatenated to the input image of interactive segmentation methods. Instead, we leverage the distance fields to enforce object localization consistency on intermediate pseudo-masks. We invert the normalized background distance field $D_{\neg c}$, and perform element-wise multiplication with all other distance maps:

$$\mathcal{D}_c = (1 - \mathcal{D}_{\neg c}) \odot \mathcal{D}_c \ \forall\ c \in \{1, \cdots, \mathcal{C}\} \quad (4)$$

Inverting $D_{\neg c}$ imposes low values in regions known to belong to the background class. By taking the element-wise product between $D_{\neg c}$ and all other distance maps, we remove regions in D_c that may be ambiguous or inconsistent with the underlying object's location. This can be observed in Figure 5, where the wrongly activated region (marked with a red box) is dampened by the distance field.

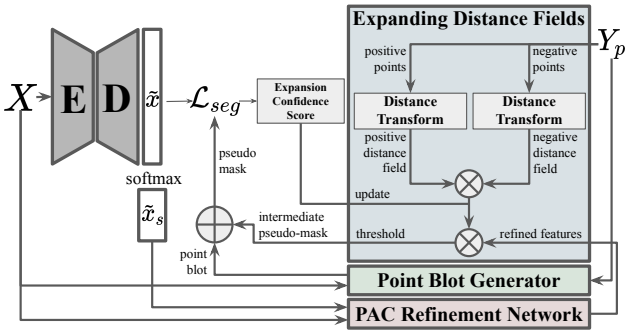

Figure 3: **Pseudo-Masks from Points (PMP)** overall architecture. Input image is fed to a fully convolutional network, and supervised by a pseudo-mask generated by the Expanding Distance Fields and Point Blot Generation modules. The network's output softmax features, $Tildex_s$, are fed to our PAC Refinement Network to be refined with accordance to local statistics of underlying features and color distributions. Then the refined features, $\tilde{x}_{refined}$ are multiplied element-wise with the expanded distance fields, which are calculated using ground truth points Y_p (composed of negative and positive points corresponding to background and objects) , and thresholded to make intermediate pseudo-masks, which are superimposed with point blot masks Y_m to make the final pseudo-mask. When training a network from scratch, early iterations tend to be unstable, producing noisy outputs and therefore unreliable pseudo-masks. Our novel Expanding Distance Fields allow from-scratch training by preventing accumulation of error in generated pseudo-masks. Best viewed in color and zoomed.

3.2.2 Expansion Mechanism

Using points as seeds to create distance fields inherently creates bias towards the regions around those seeds, especially when objects are large. For that reason, we employ our novel expansion mechanism, which aims to represent increasing dependability of the model by adaptively increasing the spatial region in which features may pass. Typically, early training iterations tend to produce noisy outputs, which are spatially filtered by the Expanding Distance Fields. As training progresses, better output feature representations are expected, and therefore less vigorous spatial filtering is required. If the distance fields are used alone without the expansion mechanism, the intermediate pseudo-masks tend to provide partial coverage for images with large objects, only focusing on the region around the seed points. Instead, we define an expansion confidence score, $\mathcal{E}_{score}$ as a function of the network's learning progress. In initial stages of training, we consider the seed point as the pixel with highest confidence, corresponding to the value of 1. As the network learns features corresponding to that class, we incrementally lower the highest confidence threshold. By doing so, we expand the distance field from the seed point

Figure 4: **Qualitative results** of our generated pseudo-masks and point blots on Pascal VOC [77] training set. Our method provides pseudo-masks converging towards fully-supervised ground truth, allowing for better performance, even when training from scratch. Pixels with low certainty (not color coded) are ignored. Dark gray pixels represent background class. Best viewed in color and zoomed.

outward, essentially enlarging the region of high confidence and allowing more refined features in to be included in the output. Formally,

$$\gamma = \frac{\mathcal{L}^{(e-1)}}{\mathcal{L}^{(e)}} - 1, \tag{5}$$

$$\mathcal{E}_{score} = \mathcal{E}_{score} + max(min(\gamma, \eta), \omega), \tag{6}$$

where $\mathcal{L}^{(e)}$ is the accumulated loss at epoch e, γ is the improvement ratio between the current and previous epochs, and η and ω are the upper and lower limits for confidence improvements to be added to $\mathcal{E}_{score}$ at that epoch. Note that performance degradation at a given epoch will result in a lower confidence score for the next epoch. We use the confidence score to modify our distance map aggregation by adding it to the distance fields, and clipping any values below 0 and above 1 as follows,

$$\mathcal{D}_{c,x,y} = \begin{cases} 1 & \text{if } \mathcal{D}_{c,x,y} + \mathcal{E}_{score} \geq 1 \\ 0 & \text{if } \mathcal{D}_{c,x,y} + \mathcal{E}_{score} \leq 0 \\ \mathcal{D}_{c,x,y} + \mathcal{E}_{score} & \text{otherwise} \end{cases} \tag{7}$$

where x and y represent all possible locations of distance field for class c. Classes not present in the image are ignored. Note that we use different expanding confidence scores for the background and the objects, and the object expanding scores increase 2 times faster than the background expanding score. The importance of this module is also visually demonstrated in the supplementary material through epoch-by-epoch distance field instances and their corresponding pseudo-masks.

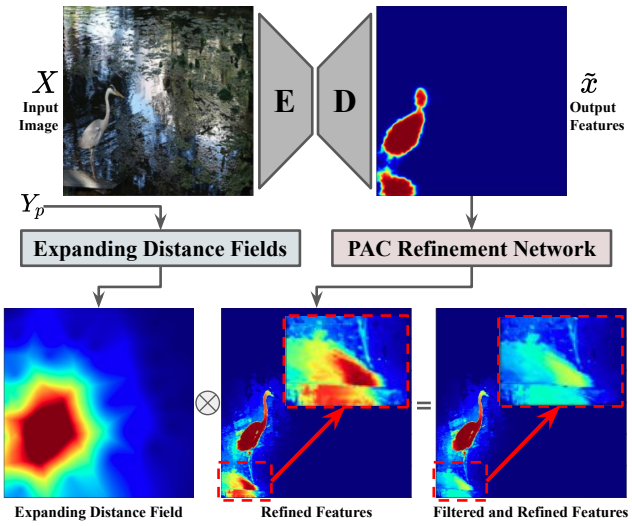

Figure 5: **Expanding Distance Fields and PAC Refinement Network** joint utility example on Pascal VOC 2012 [77]. The network output features are fed to the PAC Refiner, which output is multiplied element-wise with the expanding distance fields obtained from point annotations and expansion confidence score. Observe that the rock is wrongly activated for the bird class, which is dampened by the expanding distance field. The regions highlighted by the red dash boxes indicate wrongly activated regions and their dampened outputs. The final output is thresholded to make the intermediate pseudo-mask. Best viewed in color and zoomed.

The final step of the Expanding Distance Fields module performs an element-wise product between the refined features and the aggregated distance fields, followed by thresholding, to obtain the final pseudo-mask. Since the PAC refinement network only smooths and ensures local consistency, without global perspective, it often has activated regions that are not part of the objects. By multiplying its output with the aggregated distance fields, we spatially constrain the class activation maps to regions determined to be relevant by the distance fields. This is visually demonstrated in Figure 5 which shows the transition between each stage up to the final pseudo-mask.

3.3. Point Blot Generator

The purpose of this method is to generate a set of new local ground truth pixels from image X, and annotated points Y_p through iterative operations of perturbations and random walks over the input image X. The set of new ground truth pixels, named point blots, capture neighboring pixels that are "obviously" part of the object. Such additional pixels are essential in the early iterations because they provide reliable baseline pseudo-masks before the network is able to

			CRAID [20]	CityPersons [78]	IAD [79]	ADE20K [13]	CityScapes [34]
	Sup.	# of stages	mIoU (%)				
DeepLab-v3 [1]	$\mathcal{F}$	1	81.3	80.7	82.4	45.5	78.8
AffinityNet [18]	$\mathcal{I}$	4	49.8	49.7	53.8	0.7	8.2
PDML [80]	$\mathcal{P}$	4	-	-	-	**19.6**	-
SEAM [61]	$\mathcal{I}$	4	54.3	51.1	58.2	2.8	17.3
Triple-S [20]	$\mathcal{P}, \mathcal{D}$	1	68.7	-	-	-	-
Araslanov *et al*. [19]	$\mathcal{I}$	1	54.9	48.2	57.6	2.0	11.8
Ours	$\mathcal{P}$	1	**72.1**	**62.8**	**72.4**	16.4	**39.9**

Table 1: mIoU (%) accuracy on CRAID [20], CityPersons [78], IAD [79], ADE20K [13], and CityScapes [34] validation sets. $\mathcal{F}$, $\mathcal{I}$, $\mathcal{B}$, $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{D}$ represent full, image, box, saliency, point, and prior data annotations respectively. Our method is generalizable to arbitrary datasets, significantly outperforming our single-stage and multi-stage weakly supervised baselines on the selected real world datasets.

generate meaningful features. The role of these point blots decreases as the intermediate pseudo-mask generated by the PAC Refiner and Expanding Distance Fields improve.

Let image $X \in \mathcal{R}^{H\times W\times 3}$ and ground truth points $Y_p \in \mathcal{R}^{H\times W\times 1}$ be an input sample to the Point Blot Generation module. We obtain an initial mask, Y_m, using a random walk over X with Y_p as seeds. Then, we perturb Y_p using a random affine transformation to obtain new points $\tilde{Y_p}$, which are used as seeds for a random walk over X to generate a candidate mask $\tilde{Y_m}$. While we can guarantee that all points in Y_p lay on the correct objects, we cannot assume the same for $\tilde{Y_p}$, and consequently cannot assume that $\tilde{Y_m}$ is a good candidate mask as a whole. Instead, we partition Y_m and $\tilde{Y_m}$ into current and candidate blobs, $B, \tilde{B}$, using the connected component algorithm [81], with each current blob $b \in B$ corresponding to a candidate blob $\tilde{b} \in \tilde{B}$. We then calculate the Kullback–Leibler divergence (KLD) distance [82] between the distributions of the underlying image features enclosed by the pixels of b and $\tilde{b}$. A candidate blob is accepted as an expansion to its corresponding current blob if it fulfils two requirements: 1) the KLD distance is smaller than threshold ϕ, and 2) the intersection over union of b and $\tilde{b}$ is above threshold δ. This set of perturbations is repeated for t iterations with increasing perturbation intensity, in which random affine transformations sample from increasing ranges of rotations and translations. The KLD distance ensures that color intensity distribution of pixels in blobs are similar to each other, while the intersection over union threshold requires that we expand gradually, without creating disjoint blobs. The increased perturbations also ensure that we first explore neighboring regions to obtain successive expansion.

The point blot generation pipeline can be seen in Figure 1 in the supplementary material, and output samples in Figure 4. The method allows us to capture additional neighboring pixels around points without sacrificing excessive computation resources, increasing computational time per iteration by roughly 18.74% (with parameters described in the supplementary material).

4. Experiments

We train and evaluate the performance of our method on six datasets: Pascal VOC 2012 [77], Cranberry from Aerial Imagery Dataset (CRAID) [20], CityPersons [78], Inria Aerial Dataset (IAD) [79], ADE20K [13], and CityScapes [34]. The first is to illustrate our method's performance on a standard benchmark dataset, and the rest are examples of real world applications within various complexity ranges. While standard benchmark datasets are essential for baseline efficacy assessment, we want to demonstrate our method's generalizability and versatility in domains more common in real world applications. For our baselines, we report formal performance metrics if a performance is reported for the dataset. If performance is not reported on a given dataset and implementation is publicly available, we follow training and evaluation procedure in accordance to the method's reported approach and record the best performing result. Methods that don't have publicly available implementation and don't report on any of the complex datasets are not reported in Tab. 1. We provide implementation details, pseudo-code, and datasets details in the supplementary material.

5. Results

Table 1 demonstrates the wide range of applicable complexities of our method, which performs significantly better than our single-stage baselines ([19]) on the CRAID, CityPersons, IAD, ADE20K, and CityScapes datasets. Our method significantly outperforms our image-level single-stage baseline by up to 28.1% across datasets, and our best image-level multi-stage baseline by up to 22.6%. While PDML [80], our multi-stage point-supervision baseline, achieves better performance on ADE20K by 3.2%, it requires 4 separate stages of training and a significantly more

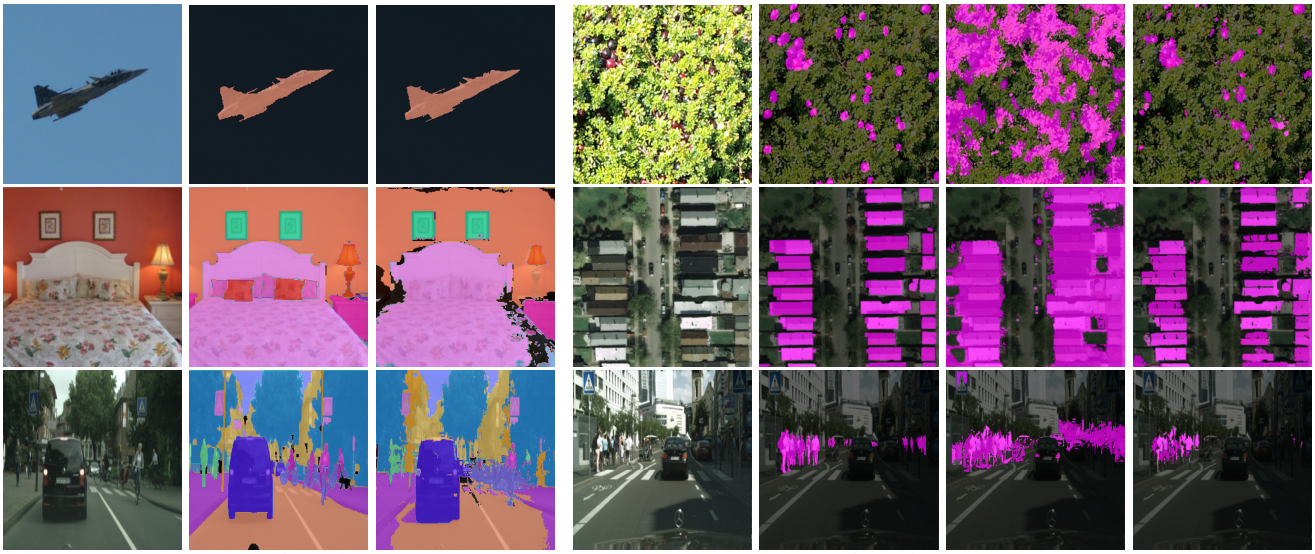

Figure 6: **Qualitative results** of our method on Pascal VOC 2012 [77] (left, first row), ADE20K [13] (left, second row), CityScapes [34] (left, third row), CRAID [20] (right, first row), IAD [79] (right, second row), and CityPersons [78] (right, third row) trained with points. Observe that our method provides significantly more refined predictions than our single-stage baseline (trained on image-level) on real world datasets (right side). Best viewed in color and zoomed. Dark gray pixels represent background class.

complex architecture. On the other hand, our method only requires a single stage and still obtains competitive performance. It can be seen that current single and multi stage WSSS methods struggle to maintain performance on data outside the benchmark complexity distribution. The poor performance of image-level label driven methods, single or multi stage, stems from the dependency on the preceding classification task to provide good class activation maps essential for localization. As the scene complexity changes, the quality of activations maps exponentially decreases, and so are resulting pseudo-masks. When the images have few (or binary) classes, the classification task becomes too easy which results in coarse feature outputs and sub-optimal localization and CAM coverage. The feature quality degradation greatly worsens with respect to the number of objects in the scene, making binary scenes such as in CRAID and CityPersons, which have large counts of small objects, increasingly difficult for any method using image-level labels for segmentation. This can also be observed in the qualitative results, with our baseline producing coarse outputs for CRAID and CityPersons datasets. In contrary, images in Pascal VOC 2012 have an average of 2.37 objects per image, making it easier to generate features without significant spatial guidance. Thorough empirical analysis of performance degradation of image-level based methods with respect to object sizes and counts is discussed in [16].

While not the focal point of this method (as an application-driven method), we also present a comparison between SOTA baselines (excluding CRF post processing) and our method on the Pascal VOC dataset in Table 4 of the supplementary material. In the single-stage approach, our method outperforms [19] by 1% on validation and 0.3% on test sets, even though we train our network from scratch, while [19] uses a pre-trained backbone. It is important to note that, as shown in [31, 32, 33], localization is "free" when a trained classification network (i.e. pre-trained backbone) is available. From our experiments, without the usage of a pre-trained backbone, [19] performs significantly worse. By using points our method can be used in broader complexity ranges and on non-standard datasets without incurring significant additional annotation costs. Ablation study and additional quantitative and qualitative results of our method for all datasets are available in the supplementary material.

5.1. Conclusion

This paper presents a practical single-stage WSSS method applicable to non-standard datasets for which pre-trained backbones are not available, or pre-training classification task is insufficient. By utilizing our expanding distance fields and point blots, our method is able to achieve SOTA performance on the benchmark dataset as well as significantly better performance than single-stage SOTA methods on real-world and application-driven domains.

Acknowledgement This project was sponsored by the USDA NIFA AFRI Award Number: 2019-67022-29922.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[2] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

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

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[5] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[6] Dingfu Zhou, Jin Fang, Xibin Song, Liu Liu, Junbo Yin, Yuchao Dai, Hongdong Li, and Ruigang Yang. Joint 3d instance segmentation and object detection for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Peri Akiva, Benjamin Planche, Aditi Roy, Kristin Dana, Peter Oudemans, and Michael Mars. Ai on the bog: Monitoring and evaluating cranberry crop risk. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2493–2502, January 2021.

[8] Qihang Yu, Dong Yang, Holger Roth, Yutong Bai, Yixiao Zhang, Alan L. Yuille, and Daguang Xu. C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] Josh Myers-Dean and Scott Wehrwein. Semantic pixel distances for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[10] Anthony Cioppa, Adrien Deliege, Maxime Istasse, Christophe De Vleeschouwer, and Marc Van Droogenbroeck. Arthus: Adaptive real-time human segmentation in sports through online distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[11] Peri Akiva, Matthew Purri, Kristin Dana, Beth Tellman, and Tyler Anderson. H2o-net: Self-supervised flood segmentation via adversarial domain adaptation and label refinement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 111–122, January 2021.

[12] Matthew J. Leotta, Chengjiang Long, Bastien Jacquet, Matthieu Zins, Dan Lipsa, Jie Shan, Bo Xu, Zhixin Li, Xu Zhang, Shih-Fu Chang, Matthew Purri, Jia Xue, and Kristin Dana. Urban semantic 3d reconstruction from multiview satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[13] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[16] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3800, 2018.

[17] Shuxin Wang, Shilei Cao, Dong Wei, Renzhen Wang, Kai Ma, Liansheng Wang, Deyu Meng, and Yefeng Zheng. Lt-net: Label transfer by learning reversible voxel-wise correspondence for one-shot medical image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018.

[19] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020.

[20] Peri Akiva, Kristin Dana, Peter Oudemans, and Michael Mars. Finding berries: Segmentation and counting of cranberries using point supervision and shape priors. In *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 50–51, 2020.

[21] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016.

[22] Jae-Hun Lee, ChanYoung Kim, and Sanghoon Sull. Weakly supervised segmentation of small buildings with point labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7406–7415, 2021.

[23] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[25] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. In *AAAI*, 2021.

[26] Yuting Liu, Miaojing Shi, Qijun Zhao, and Xiaofang Wang. Point in, box out: Beyond counting persons in crowds. In *CVPR*, 2019.

[27] Dingkang Liang, Wei Xu, Yingying Zhu, and Yu Zhou. Focal inverse distance transform maps for crowd localization and counting in dense crowd. *arXiv preprint arXiv:2102.07925*, 2021.

[28] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *ICCV*, 2019.

[29] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018.

[30] Seunghoon Hong, Donghun Yeo, Suha Kwak, Honglak Lee, and Bohyung Han. Weakly supervised semantic segmentation using web-crawled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7322–7330, 2017.

[31] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[32] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[34] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] Jae-Hun Lee, ChanYoung Kim, and Sanghoon Sull. Weakly supervised segmentation of small buildings with point labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7406–7415, October 2021.

[36] Youngmin Oh, Beomjun Kim, and Bumsub Ham. Background-aware pooling and noise-aware loss for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6913–6922, June 2021.

[37] Muming Zhao, Jian Zhang, Chongyang Zhang, and Wenjun Zhang. Leveraging heterogeneous auxiliary tasks to assist crowd counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12736–12745, 2019.

[38] Ruochen Fan, Ming-Ming Cheng, Qibin Hou, Tai-Jiang Mu, Jingdong Wang, and Shi-Min Hu. S4net: Single stage salient-instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6103–6112, 2019.

[39] Yuan Cheng, Rui Lin, Peining Zhen, Tianshu Hou, Chiu Wa Ng, Hai-Bao Chen, Hao Yu, and Ngai Wong. Fassst: Fast attention based single-stage segmentation net for real-time instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2210–2218, 2022.

[40] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6951–6960, 2019.

[41] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020.

[42] Tianyu Wang, Xiaowei Hu, Chi-Wing Fu, and Pheng-Ann Heng. Single-stage instance shadow detection with bidirectional relation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–11, 2021.

[43] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14494–14503, 2021.

[44] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020.

[45] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Retinatrack: Online single stage joint detection

and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14668–14678, 2020.

[46] Jun Yu, Xinlong Hao, and Peng He. Single-stage face detection under extremely low-light conditions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2021.

[47] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11772–11781, 2021.

[48] Tianyi Chen, Yi Liu, Yunfei Zhang, Si Wu, Yong Xu, Feng Liangbing, and Hau San Wong. Semi-supervised single-stage controllable gans for conditional fine-grained image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9264–9273, 2021.

[49] Léon Bottou. Online algorithms and stochastic approxima-p tions. *Online learning and neural networks*, 1998.

[50] Heinz Breu, Joseph Gil, David Kirkpatrick, and Michael Werman. Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, 1995.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[53] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[54] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[55] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[56] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[57] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2119–2128, 2016.

[58] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[59] Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly supervised object localization with progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3520, 2016.

[60] Arslan Chaudhry, Puneet K Dokania, and Philip HS Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1707.05821*, 2017.

[61] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020.

[62] Wataru Shimoda and Keiji Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5208–5217, 2019.

[63] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019.

[64] Dmitrii Marin and Yuri Boykov. Robust trust region for weakly supervised segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6608–6618, October 2021.

[65] Hongjun Chen, Jinbao Wang, Hong Cai Chen, Xiantong Zhen, Feng Zheng, Rongrong Ji, and Ling Shao. Seminar learning for click-level weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6920–6929, October 2021.

[66] Shiyi Lan, Zhiding Yu, Christopher Choy, Subhashree Radhakrishnan, Guilin Liu, Yuke Zhu, Larry S. Davis, and Anima Anandkumar. Discobox: Weakly supervised instance segmentation and semantic correspondence from box supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3406–3416, October 2021.

[67] Jingshan Xu, Chuanwei Zhou, Zhen Cui, Chunyan Xu, Yuge Huang, Pengcheng Shen, Shaoxin Li, and Jian Yang. Scribble-supervised semantic segmentation inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15354–15363, October 2021.

[68] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11166–11175, 2019.

[69] Anne S Wannenwetsch and Stefan Roth. Probabilistic pixel-adaptive refinement networks. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11642–11651, 2020.

[70] Varun Ravi Kumar, Marvin Klingner, Senthil Yogamani, Stefan Milz, Tim Fingscheidt, and Patrick Mader. Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–71, 2021.

[71] Aysegul Dundar, Karan Sapra, Guilin Liu, Andrew Tao, and Bryan Catanzaro. Panoptic-based image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8070–8079, 2020.

[72] Xiangyu Xu, Yongrui Ma, and Wenxiu Sun. Learning factorized weight matrix for joint filtering. In *International Conference on Machine Learning*, pages 10587–10596. PMLR, 2020.

[73] Stefanie Tanujaya, Tieh Chu, Jia-Hao Liu, and Wen-Hsiao Peng. Semantic segmentation on compressed video using block motion compensation and guided inpainting. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2020.

[74] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[75] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[76] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3129–3136. IEEE, 2010.

[77] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[78] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *CVPR*, 2017.

[79] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3226–3229. IEEE, 2017.

[80] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8843–8850, 2019.

[81] Michael B Dillencourt, Hanan Samet, and Markku Tamminen. A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM (JACM)*, 39(2):253–280, 1992.

[82] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.