Pixel Contrastive-Consistent Semi-Supervised Semantic Segmentation

Yuanyi Zhong[1*], Bodi Yuan[2], Hong Wu[2], Zhiqiang Yuan[2], Jian Peng[1], Yu-Xiong Wang[1]

[1] University of Illinois at Urbana-Champaign
{yuanyiz2, jianpeng, yxw}@illinois.edu

[2] X, The Moonshot Factory
{bodiyuan, wuh, zyuan}@google.com

Abstract

We present a novel semi-supervised semantic segmentation method which jointly achieves two desiderata of segmentation model regularities: the label-space consistency property between image augmentations and the feature-space contrastive property among different pixels. We leverage the pixel-level ℓ_2 loss and the pixel contrastive loss for the two purposes respectively. To address the computational efficiency issue and the false negative noise issue involved in the pixel contrastive loss, we further introduce and investigate several negative sampling techniques. Extensive experiments demonstrate the state-of-the-art performance of our method (PC^{2}Seg) with the DeepLab-v3+ architecture, in several challenging semi-supervised settings derived from the VOC, Cityscapes, and COCO datasets.

1. Introduction

Modern deep learning based solutions [3,4,45] to semantic segmentation typically require large-scale pixel-wise annotated datasets [11, 15, 32] (*i.e.*, the high-data regime). When the deep models are trained with limited labeled data (*i.e.*, the low-data regime), however, their performance drops drastically due to over-fitting. The performance drop in the low-data regime and the high annotation cost associated with dense pixel labeling have motivated the community to study semi-supervised segmentation methods [17,26,37,56,57]. In the semi-supervised setting, a model is trained with the help of an additional large-scale unlabeled dataset. A successful semi-supervised method is useful in practice – only a portion of the production data needs to be densely annotated to deliver a satisfactory model.

As shown in Fig. 1, *our key insight* is that there are *two desired regularity properties* which a semi-supervised segmentation model should possess. The first one is the *consistency property in the label space*. The output segmentation mask of the model should be invariant (or equivariant) to the transformations of the input. When an input image goes through two different color augmentations, the segmentation mask should stay the same, since the object semantics and locations are unchanged. The second property is the *contrastive property in the feature space*. By "contrastive," we mean that the intermediate features of the model should have the discriminative power to group visually similar pixels together while distinguishing them from visually dissimilar pixels. Such a contrastive property will enable the model to classify pixels into correct semantic categories.

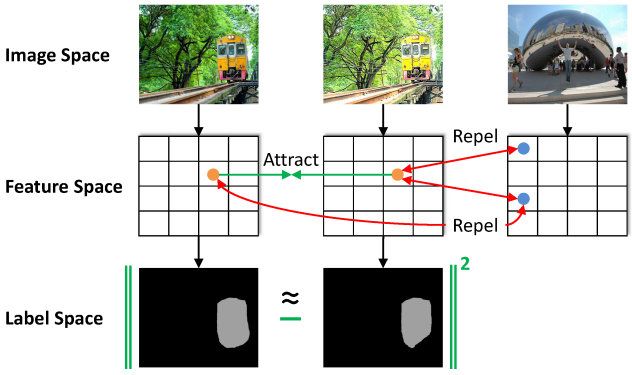

Fig. 1: Two desired properties of a semi-supervised segmentation model and our corresponding techniques to simultaneously achieve such properties: (1) Consistency in the label space – the predicted mask should be invariant to augmentations. We thus leverage the pixel-wise ℓ_2 loss between two views of the same unlabeled image. (2) Contrastiveness in the feature space – the features should be able to distinguish similar pixel pairs from dissimilar pairs. To this end, we introduce the pixel contrastive loss that pulls positive pixel pairs closer and pushes negative pairs away.

State-of-the-art semi-supervised segmentation methods, in fact, benefit from using the aforementioned label-space consistency property on the unlabeled images, in the forms of invariance under data augmentations [17, 57], invariance under feature perturbations [37], consistency of different network branches [26], and model self-consistency [56]. However, the explicit application of the feature-space

*Work partly done during an internship at X, The Moonshot Factory.

contrastive property and the possibility of the joint label-consistent and feature-contrastive regularization have been largely under-explored.

Based on our insight, this paper explores how to leverage and *simultaneously* enforce the consistency property in the label space and the contrastive property in the feature space, leading to a novel *pixel contrastive-consistent semi-supervised segmentation* (PC^{2}Seg) method. For the label consistency property, we introduce a simple pixel-wise ℓ_2 consistency loss between the outputs of weakly and strongly augmented views of the same image. For the feature contrastive property, we extend the popular InfoNCE-based contrastive loss [5, 19] and make significant modifications *for its use at the pixel level* on an intermediate feature map.

More concretely, the pixel contrastive loss in semantic segmentation faces the unique technical challenges of high computational cost and harmful false negative examples, compared with the standard image-level contrastive learning. The potentially high computation is due to the need of contrasting a large number of pixels. The harm of false negative examples (*i.e.*, when examples of the same class as the anchor are mistakenly chosen as the contrasting examples) has been noted in image-level contrastive learning [10, 23]. This problem becomes particularly severe in segmentation, where accurate per-pixel predictions are required compared with image classification. To overcome these issues, we develop simple-yet-effective negative sampling strategies. For each positive pair of pixels, we sample only a moderate number of negative pixels from the mini-batch for efficiency and avoid false negative pixels by a *simple cross-image and pseudo-label weighting* heuristic.

Our contributions are three-fold:

1. We propose the first framework that leverages both pixel-consistency (in the label space) and pixel-contrastive (in the feature space) properties for semi-supervised semantic segmentation. We show that these two properties are complementary and their synergy is important.
2. We generalize the existing image-level contrastive learning to pixel-level. To overcome the computational cost and false negative difficulties inherent in segmentation tasks, we propose a novel negative sampling technique and investigate its four variants.
3. We demonstrate state-of-the-art performance on multiple widely-used benchmarks. This is achieved relatively easily by leveraging the proposed framework together with standard loss functions and data augmentations, without sacrificing efficiency compared to other semi-supervised methods.

2. Related Work

Contrastive Learning. Self-supervised or unsupervised visual representation learning has been gaining momentum [5, 6, 8, 18, 19]. At the center of the recent progress is the contrastive learning based pretext tasks [5, 6, 19], which learn representations that discriminate similar image pairs (constructed from different augmentations of the same images) from dissimilar, negative image pairs. A variety of strategies have been investigated to choose appropriate negative pairs. In MoCo [7, 19], a memory buffer and a momentum encoder are maintained to provide negative samples. In SimCLR [5, 6], the negative samples are the large training mini-batches. Improper negative pairs might hurt learning performance: Particularly related to our negative pixel sampling strategies, [10, 23, 49] have proposed ways to alleviate the bias issue caused by incorrect (false) negative images by modifying the contrastive loss function. PC^{2}Seg does not change the loss function but rather samples the negative examples strategically. SimSiam [8] and BYOL [18] find that simple consistency training without negatives can also be effective with carefully designed architectures and training procedures. However, our ablation study shows that PC^{2}Seg still benefits from negative samples compared with feature-space consistency training alone.

Pixel-level contrastive learning has not been well-explored until very recently. [38, 44, 47, 51] explore dense pixel-level self-supervised pretraining. They show that pixel-level pretext tasks can transfer better to segmentation than image-level self-supervised learning. In this paper, we attempt to benefit from pixel-level contrastive learning in the semi-supervised rather than the unsupervised setting. We present a strong semi-supervised approach that jointly optimizes a contrastive loss on the intermediate features and a consistent loss on the output masks.

Contrastive learning can be used in the supervised setting [27] to improve generalization. Researchers have attempted to apply supervised contrastive learning for semantic segmentation [46, 54]. [46] directly leverages pixel contrasting in supervised segmentation training, while [54] adopts it during the first supervised stage in a multi-stage semi-supervised setting. Compared with these attempts, we focus on the single-stage semi-supervised setting, where *self-supervised* contrastive loss is jointly applied on the *unlabeled* images branch without using any ground-truth labels.

Semi-Supervised Learning. Semi-supervised learning aims at leveraging labeled data and a large amount of unlabeled data. The idea of consistency regularization is inherent in many successful semi-supervised approaches. Iterative pseudo-labeling and re-training [30, 42, 56] implicitly enforces consistent predictions between the current model (student) and its past versions (teacher), leading to smaller entropy and larger class separation of the predicted label distribution. FixMatch [39], VAT [35], and UDA [50] leverage consistency regularization between weak and strong (or local adversarial [35]) augmented views of the same unlabeled image in a single-stage training pipeline. S4L [52]

explores a self-supervised auxiliary task (*e.g.*, predicting rotations) on unlabeled images jointly with a supervised task.

Semantic Segmentation. Semantic segmentation is the task of predicting pixel-level category labels from images. High segmentation accuracy can be reached by deep fully convolutional neural networks (FCNs) trained on large datasets [3,4,45]. In these models, the convolutional nature of FCNs is exploited to generate dense prediction masks. Following [17, 57], our work builds upon the widely-used DeepLab-v3+ segmentation model [4].

Semi-Supervised Semantic Segmentation. It is compelling to study semi-supervised segmentation approaches to reduce the mask labeling cost. [40] synthesizes additional training data with generative adversarial networks (GANs). [22, 34] couple an adversarial loss on the predicted masks with the standard supervised loss. In the line of consistency training, the output predictions of the unlabeled images are encouraged to be consistent across different augmented views [17,28,36,57], feature embedding perturbations [37], and different networks as in co-training [26, 48]. When the consistency loss is set up explicitly as using one branch's prediction as the target to train the other branch, the former predictions are often called pseudo labels [17, 57]. Self-training, which retrains the model with pseudo labels on the unlabeled data given by the previous iteration of the model, has also shown promising results for semantic segmentation [2, 55, 56]. While consistency training is extensively studied, the use of (pixel) contrastive leaning has largely eluded discussion in the semi-supervised segmentation setting.

Finally, alternative ways to reduce the annotation cost includes leveraging various forms of weak labels such as image labels [1, 33, 41, 48, 57], bounding boxes [12], and scribbles [31], or even purely unlabeled data as in unsupervised segmentation by clustering [25] and self-supervised tasks [53]. We refer interested readers to these works as the problem settings are different.

3. Method

3.1. Overview

Fig. 2 summarizes the overall architecture and training procedure of our pixel contrastive-consistent semi-supervised segmentation (PC^{2}Seg) approach. The entire pipeline consists of two data streams, with one stream for labeled images and the other for unlabeled images. The complete loss function is the sum of the labeled and unlabeled stream losses:

$$\mathcal{L} = \mathcal{L}^{\text{label}}(x, y) + \mathcal{L}^{\text{unlabel}}(x), \tag{1}$$

where we denote an image as x, and its ground-truth segmentation mask as y.

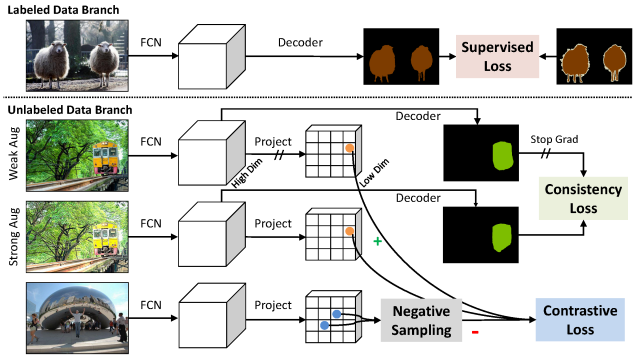

Fig. 2: PC^{2}Seg overview. The training pipeline has two input streams: One for labeled images and the other for unlabeled images. The images pass through a shared fully convolutional network (FCN) to generate mid-level features and a shared decoder to generate the mask predictions. The labeled branch is trained as usual with the supervised cross-entropy loss. Two augmented views (weak and strong) are generated from unlabeled images. Then we use the predicted masks of the weak view as pseudo labels and impose a consistency loss between the predicted masks of the two views. A pixel contrastive loss is applied on the projected low-dimensional features of an intermediate layer with negative pixels sampled by the strategies in Sec. 3.2.1.

For the labeled images, the typical supervised cross-entropy loss between the predictions and the segmentation masks is applied at per-pixel locations. For the unlabeled images, in line with existing semi-supervised [39, 57] and self-supervised [5, 8, 19, 23] methods, we build a siamese network architecture with shared weights that operates on two augmented views (weak and strong) of a single unlabeled image. The weak augmentations consist of the usual crop, flip, and resize transformations, while the strong augmentations add color (brightness, contrast, and hue) and Cutout [14] transformations on top of the weak augmentations following [57]. We then jointly apply the label consistency loss ℓ^{cy} on the output masks and the unlabeled pixel constrastive loss ℓ^{ce} on the intermediate features:

$$\mathcal{L}^{\text{unlabel}} = \sum_i \lambda_1 \ell^{\text{ce}}(\boldsymbol{z}_i, \boldsymbol{z}_i^+, \{\boldsymbol{z}_{in}^-\}_{n=1}^N) + \lambda_2 \ell^{\text{cy}}(\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_i^+). \tag{2}$$

Here we use i to index the pixels, use $\boldsymbol{z}_i$, $\boldsymbol{z}_i^+$, and $\boldsymbol{z}_{in}^-$ to represent the anchor pixel from one augmented view, the positive pixel from the other augmented view, and the negative pixels, respectively. $\hat{\boldsymbol{y}}_i$ and $\hat{\boldsymbol{y}}_i^+$ are the predicted pixel class softmax probabilities of the two views. λ_1 and λ_2 are trade-off hyper-parameters. Exactly which intermediate feature map to apply ℓ^{ce} is a design choice.

Consistency Loss. We adopt the normalized ℓ_2 pixel consistency loss (equivalently, $1 -$ cosine similarity) between the corresponding predicted softmax probabilities of the

weak and strong views:

$$\ell_i^{\text{cy}} = 1 - \cos(\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_i^+). \tag{3}$$

The cosine similarity is $\cos(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2}$. In practice, we sharpen the softmax for the weak branch output $\hat{\boldsymbol{y}}_i$ by dividing the logits by temperature 0.5. We also put a stop-gradient operation on the weak output as in [5, 8, 18, 57], so that the gradients only back-propagate into the strong branch but not the weak branch. This effectively makes the weak branch predictions $\hat{\boldsymbol{y}}_i$ act as the pseudo labels to train the strong branch. Empirically we found that our ℓ_2 consistency loss outperforms the cross-entropy loss in [57].

3.2. Pixel Contrastive Loss

Suppose the feature map, of either the weak branch or the strong branch, has shape $B \times H \times W \times D$, where B is the batch size, H and W are the height and width, and D is the feature dimension. We denote $i \in \{1, 2, \ldots, B \times H \times W\}$ as the index of one pixel location on this feature map. Given the feature vector $\boldsymbol{z}_i \in \mathbb{R}^D$ of the anchor pixel, the goal of contrastive learning is to increase its similarity to a positive pixel $\boldsymbol{z}_i^+$ and reduce its similarity to N negative pixels $\{\boldsymbol{z}_{in}^-\}_{n=1}^N$. A popular choice of the loss function to achieve such goal is the InfoNCE [21] loss, which is relevant to the noise contrastive mutual information estimator. When combining the InfoNCE loss with the cosine similarity, we arrive at the pixel contrastive loss in the form of

$$\ell_i^{\text{ce}} = -\log \frac{e^{\cos(\boldsymbol{z}_i, \boldsymbol{z}_i^+)/\tau}}{e^{\cos(\boldsymbol{z}_i, \boldsymbol{z}_i^+)/\tau} + \sum_{n=1}^{N} e^{\cos(\boldsymbol{z}_i, \boldsymbol{z}_{in}^-)/\tau}}. \tag{4}$$

Here τ is a temperature hyper-parameter to control the scale of terms inside exponential, which is set to 0.07 throughout our experiments following prior work [5, 6]. The use of cosine similarity is shown to be effective in existing contrastive learning work [5, 5, 18, 19, 43, 46].

Directly optimizing Eq. 4, however, is challenging. The first difficulty is that the raw feature vectors can be quite high-dimensional, *e.g.*, the last ResNet backbone feature map has shape $33 \times 33 \times 2,048$ in DeepLab [4], leading to high memory and computational cost. We therefore utilize a linear projection layer to reduce the feature dimension from $2,048$ to 128, consistent with [23, 46]. The projection head parameters of the weak and strong branches can be separated or tied together. We found that separated projection heads work slightly better in our experiments. In the weak branch, similar to the consistency loss, a stop gradient operation is inserted before the projection.

The next major difficulty lies in deciding the positive and negative pairs. In the ideal supervised learning setting, *i.e.*, when the ground-truth pixel labels are known, we can simply select the positive pixel from the pixels of the true category of the anchor pixel, and choose the negative pixels as the pixels from different categories [27]. By contrast, the issue becomes complicated when the labels are unknown as in our setting. Existing work on image-level contrastive learning circumvents the problem by forcing the positive example to come from another augmented view of the same image, while using both views of a large mini-batch [5] or a memory bank [19] as the source of negative examples.

However, these techniques are not easily generalizable to our task. For contrastive learning at pixel level, we still choose the positive pixel as the corresponding pixel under a random color augmentation. For example, if $\boldsymbol{z}_i$ belongs to the weak view, $\boldsymbol{z}_i^+$ can be the corresponding pixel in the strong view. As for the negative pixels, two issues occur if we directly borrow existing approaches in image-level contrastive learning: (1) Due to the huge number of pixels, we cannot afford to use all the pixels in the mini-batch as negative examples, because the memory and computational cost scale with $O(\text{num\_pixels}^2)$; (2) Since the task is at pixel-level, the segmentation model can be sensitive to noise from incorrect negative pixels, if we adopt a simple mini-batch or memory bank approach. A pixel of the same category as the anchor pixel might be wrongly chosen as a negative pixel, leading to ineffective or even misleading learning signal. To resolve these issues, we propose to sub-sample a fixed number of negative pixels for each anchor pixel, and develop several effective sub-sampling strategies as below.

3.2.1 Negative Sampling Strategies

Consider the i-th anchor pixel. Assume the N negative pixels $\boldsymbol{z}_{in}^-$ are sampled without replacement according to a discrete distribution p_{ij}, which is defined on a total of $M = B \times H \times W$ candidate pixels. More formally,

$$\boldsymbol{z}_{in}^- \sim \text{Discrete}\left(\{\boldsymbol{z}_j\}_{j=1}^M; \{p_{ij}\}_{j=1}^M\right). \tag{5}$$

Different sampling strategies essentially define different sampling distributions $\{p_{ij}\}_{j=1}^M$. We investigate and compare four of them in this paper, with illustration in Fig. 3:

Uniform. The most straightforward way is to sample negative pixels uniformly from the mini-batch. Suppose there are M valid pixels in the current mini-batch; each pixel $j = 1, 2, \ldots, M$ gets a uniformly distributed density, $p_{ij} = \frac{1}{M}$.

As one can imagine, since many pixels on the same image may belong to the same category, the uniform strategy can produce a large number of false negative pixels, which may hurt performance. This issue motivates us to study alternative sampling distribution.

Different Image. One idea to fix the false negative issue is to force the negative pixels to come from a different image, such that the chance of sampling an incorrect negative is reduced. Formally, we denote I_i and I_j as the image IDs of the anchor pixel i and a candidate negative pixel j, and $\mathbf{1}\{\cdot\}$

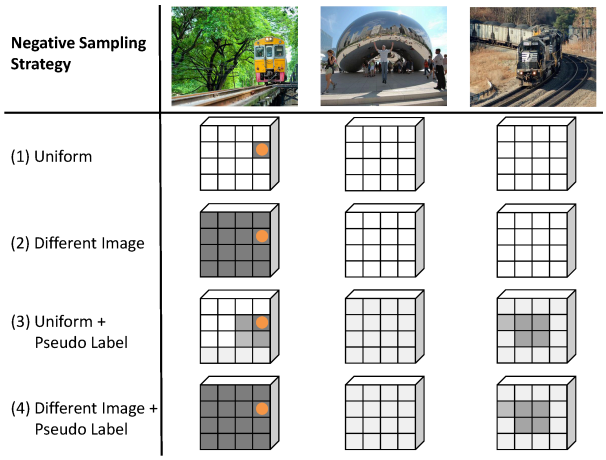

Fig. 3: Illustration of negative sampling strategies. The anchor pixel (at the location of the train) is marked with the orange dot. Darker shade means lower chance of sampling that pixel. (1) Uniform strategy only avoids sampling the anchor pixel. (2) Different Image avoids sampling from the same image. In (3)(4), the Pseudo Label correction reduces the chance of sampling the train pixels in the last column.

as the indicator function. Pixel j gets a non-zero density only if it belongs to a different image from pixel i.

$$p_{ij} = \frac{\mathbf{1}\{I_i \neq I_j\}}{\sum_{k=1}^{M} \mathbf{1}\{I_i \neq I_k\}}. \tag{6}$$

Pseudo-Label Debiased. Another idea to fix the issue is to leverage the model predictions to filter out possible pixels of the same class. This strategy is similar in spirit to [10, 23]. Suppose $\hat{y}_i$ and $\hat{y}_j$ ($\in \{1, 2, \ldots, \text{num\_classes}\}$) are the predicted pseudo labels of pixels i and j, respectively (through a bilinear image resizing if necessary). The probability of sampling pixel j as negative pixel can be weighted by $P(\hat{y}_i \neq \hat{y}_j)$, which biases the sampling process towards pixels that the model believes are from different categories. After a simple derivation, $P(\hat{y}_i \neq \hat{y}_j)$ can be rewritten as one minus the dot product of the predicted label probability vectors $\vec{\boldsymbol{y}}_i$ and $\vec{\boldsymbol{y}}_j$.

$$p_{ij} = \frac{P(\hat{y}_i \neq \hat{y}_j)}{\sum_{k=1}^{M} P(\hat{y}_i \neq \hat{y}_k)} = \frac{1 - \vec{\boldsymbol{y}}_i^\top \vec{\boldsymbol{y}}_j}{\sum_{k=1}^{M} 1 - \vec{\boldsymbol{y}}_i^\top \vec{\boldsymbol{y}}_k}. \tag{7}$$

The derivation is quite straight-forward. Ideally, we want to sample from the discrete measure: $\mathbf{1}\{j : y_i \neq y_j\} = P(y_i \neq y_j) \in \{0, 1\}$. Since the ground-truth is unknown, we may approximate $P(y_i \neq y_j)$ with the pseudo-labels $\hat{y}_i$ and $\hat{y}_j \in \{1, 2, \ldots, \text{num\_classes}\}$. Let the predicted probability of pixel i belonging to category c be $P(\hat{y}_i = c)$ and the vector $\vec{\boldsymbol{y}}_i = [P(\hat{y}_i = 1), P(\hat{y}_i = 2), \ldots]$, and then we have $P(y_i \neq y_j) \approx P(\hat{y}_i \neq \hat{y}_j) = 1 - P(\hat{y}_i = \hat{y}_j) = 1 - \sum_c P(\hat{y}_i = c, \hat{y}_j = c) = 1 - \sum_c P(\hat{y}_i = c) P(\hat{y}_j = c) = 1 - \vec{\boldsymbol{y}}_i^\top \vec{\boldsymbol{y}}_j$.

Different Image + Pseudo-Label Debiased. The above two strategies can be combined to further reduce the chance of sampling false negative pixels. In another word, we ensure that the negative pixels are coming from different images and at the same time reweight them by the pseudo labels.

$$p_{ij} = \frac{\mathbf{1}\{I_i \neq I_j\} \cdot (1 - \vec{\boldsymbol{y}}_i^\top \vec{\boldsymbol{y}}_j)}{\sum_{k=1}^{M} \mathbf{1}\{I_i \neq I_k\} \cdot (1 - \vec{\boldsymbol{y}}_i^\top \vec{\boldsymbol{y}}_k)}. \tag{8}$$

Given the density function p_{ij}, sampling the N pixels without replacement can be efficiently implemented with the Gumbel top-k trick [24, 29]. We omit the detail here.

4. Experiment

4.1. Datasets and Settings

VOC 2012. The PASCAL VOC 2012 segmentation dataset [15] comes with a finely labeled TRAIN split of 1,464 images, a VAL split of 1,449 images, and a coarsely labeled AUG split of 9,118 images. We use the public 1/2, 1/4, 1/8, and 1/16 subsets of TRAIN in [57] as the labeled data. The TRAIN and AUG sets are combined as the unlabeled dataset. We also compare with existing semi-supervised semantic segmentation methods under the 1.5k/9k split setting [22, 37, 40, 57], which treats TRAIN as the labeled set and TRAIN+AUG as the unlabeled set. VOC contains 20 object categories (excluding the background class). The performance is evaluated by the *mean intersection-over-union* (mIoU) metric on the VAL set.

Cityscapes. Cityscapes [11] contains images of urban driving scenes. The TRAIN_FINE split has 2,975 training images, and VAL_FINE has 5,000 images. We employ the public 1/4, 1/8, and 1/30 subsets of the TRAIN_FINE images in recent literature [16, 17, 22, 34, 36, 57] as the labeled data, while all images of the original TRAIN_FINE are used as the unlabeled data. We report the mIoU over the 18 foreground categories in the Cityscapes VAL_FINE set.

COCO. COCO [32] is a challenging detection and segmentation dataset of 80 categories. It is significantly larger (118,287 images in the official TRAIN set) and more diverse than VOC or Cityscapes. We reuse the same 1/32, 1/64, 1/128, 1/256, and 1/512 splits of the TRAIN set from [57] as the labeled data, and further enrich the benchmark by constructing the uniformly downsampled 1/8 and 1/16 labeled variants. We again evaluate the performance by the mIoU metric on the official VAL set (5,000 images).

Implementation Details. We adopt the DeepLab-v3+ architecture [4] as the test bed, which consists of a fully convolutional backbone, an atrous spatial pyramid pooling (ASPP) based encoder, and a shallow decoder. We study

Tab. 1: VOC 2012 1/2-1/16 labeled TRAIN set results. The settings follow [57]. The numbers of labeled images are shown inside the parentheses. R50 and R101 refer to ResNet 50 and 101. We measure the mIoU on the VOC 2012 VAL set. The mIoUs of compared methods are the reproduced results in [57] under these data splits. The results of [17, 57] and our method are obtained by DeepLab-v3+; [37] uses PSP-Net whose base performance is comparable to DeepLab-v3+, while other methods use DeepLab-v2. PC^{2}Seg obtains favorable results over existing methods. Due to the high variance, we also include the standard deviation of 3 runs on differently sampled 1/16 splits.

Method	Net	1/2 (732)	1/4 (366)	1/8 (183)	1/16 (92)
AdvSemSeg [22]	R101	65.27	59.97	47.58	39.69
CCT [37]	R50	62.10	58.80	47.60	33.10
MT [42]	R101	69.16	63.01	55.81	48.70
GCT [26]	R101	70.67	64.71	54.98	46.04
VAT [35]	R101	63.34	56.88	49.35	36.92
CutMix [17]	R101	69.84	68.36	63.20	55.58
PseudoSeg [57]	R50	70.42	64.85	61.88	54.89
PseudoSeg [57]	R101	72.41	69.14	65.50	**57.60**
Sup. baseline	R50	65.73	57.76	49.57	43.97
Sup. baseline	R101	66.28	60.02	49.83	40.02
PC^{2}Seg (Ours)	R50	70.90	67.62	64.63	56.90 $\pm$1.30
PC^{2}Seg (Ours)	R101	**73.05**	**69.78**	**66.28**	**57.00 $\pm$1.32**

Tab. 2: VOC 2012 1.5k/9k split results (TRAIN 1.5k images as labeled data and TRAIN 1.5k + AUG 9k as unlabeled data). The numbers of other methods are directly taken from the corresponding publications.

Method	Network	mIoU (%)
GANSeg [40]	VGG-16	64.10
AdvSemSeg [22]	ResNet-101	68.40
CCT [37]	ResNet-50	69.40
PseudoSeg [57]	ResNet-50	71.00
PseudoSeg [57]	ResNet-101	73.23
Sup. baseline	ResNet-50	68.81
Sup. baseline	ResNet-101	72.00
PC^{2}Seg (Ours)	ResNet-50	**72.26**
PC^{2}Seg (Ours)	ResNet-101	**74.15**

different backbones including ResNet-50 (the DeepLab modified beta variant [4]), ResNet-101 [20], and Xception-65 [9], to test the robustness of our approach to backbone variations. The backbones are initialized from their ImageNet [13] pretrained weights.

The hyper-parameters of our approach mainly include the loss coefficients, the negative sampling strategies, and other design choices of the pixel contrastive loss. They are tuned with the VOC 1/8 split and ResNet-50. Please refer to the ablation study in Sec. 4.3 for details. These hyper-parameters are used for other benchmarks without further tuning. The performance could be further improved by dataset-specific hyper-parameter selection. In the main results, we set the loss coefficients as $\lambda_1 = 0.3$ and $\lambda_2 = 1$. The contrastive loss is computed for each anchor pixel in the last backbone feature maps right before the encoder network (*i.e.*, Conv5 of ResNets and ExitFlow2 of Xception) of both the weak and strong views. The number of negative pixels is 200. They are sampled from both views with the "Different Image + Pseudo Label" strategy. Additional training details and DeepLab-related hyper-parameters are presented in the supplementary material.

Tab. 3: Cityscapes experiment results. The labeled data is varied from 1/30 to full of the original TRAIN_FINE set. We evaluate the mIoU on the VAL_FINE set. The result of [17] is reproduced by [57] based on DeepLab-v3+, while the results of [16, 22, 34, 36] are their reported numbers based on DeepLab-v2. R50 and R101 refer to ResNet 50 and 101. Our approach *achieves higher scores than all previous methods*. In the 1/4 and 1/8 settings, even the weaker ResNet-50 backbone performs quite favorably.

Method	Net	1 (2,975)	1/4 (744)	1/8 (377)	1/30 (100)
AdvSemSeg [22]	R101	-	62.3	58.8	-
s4GAN [34]	R101	65.8	61.9	59.3	-
DMT [16]	R101	68.16	-	63.03	54.80
ClassMix [36]	R101	-	63.63	61.35	-
CutMix [17]	R101	-	68.33	65.82	55.71
PseudoSeg [57]	R101	-	72.36	69.81	60.96
Sup. baseline	R50	73.64	72.86	68.06	55.25
Sup. baseline	R101	74.88	73.31	68.72	56.09
PC^{2}Seg (Ours)	R50	75.39	73.80	72.11	60.37
PC^{2}Seg (Ours)	R101	**75.99**	**75.15**	**72.29**	**62.89**

4.2. Main Results

VOC 2012. We report the results on the 1/2-1/16 labeled splits in Tab. 1. Our method (PC^{2}Seg) outperforms the prior state-of-the-art (PseudoSeg [57]) and other consistency-based approaches in almost all data splits with the ResNet-50 and ResNet-101 backbones. Note that the ratio of labeled data is small in this setting, since we use the entire TRAIN+AUG 10,575 images as the unlabeled data. We observe that the gain of our approach is *particularly noteworthy for moderately low-data regimes* such as the 1/2 and 1/4 splits. The more extreme case of 1/16 labeled split (92 labeled images) is quite challenging, as there is very little supervision signal for the semi-supervised model to learn in the first place. We thus augment the weak branch with the momentum averaging technique as in MoCo (momentum 0.99) and report the mean and standard deviation (std) of

Tab. 4: COCO results. The original TRAIN set is split into 1/512-1/8 subsets to be the labeled data. PseudoSeg [57] results are taken from their paper. PC^{2}Seg performs *consistently* better than the supervised baseline and [57]. The gain over the supervised baseline is the largest in the 1/256 split, while our 1/8 split result *almost matches the supervised full data result*.

Method	Network	Full (118k)	1/8 (14786)	1/16 (7393)	1/32 (3697)	1/64 (1849)	1/128 (925)	1/256 (463)	1/512 (232)
Sup. baseline	Xception-65	50.10	47.91	45.12	42.24	37.80	33.60	27.96	22.94
PseudoSeg [57]	Xception-65	-	-	-	43.64	41.75	39.11	37.11	29.78
PC^{2}Seg	Xception-65	**50.67**	**49.91**	**48.07**	**46.05**	**43.67**	**40.12**	**37.53**	**29.94**

3 runs of the 3 randomly generated 1/16 splits and observe that our method is *comparable* to [57].

We also compare the results of the 1.4k/9k split in Tab. 2. We find that our method outperforms the prior arts under this setting, achieving 74.15% mIoU with ResNet-101.

Cityscapes. The Cityscapes results are shown in Tab. 3. Across all semi-superivsed settings, PC^{2}Seg outperforms the supervised baseline by a large margin. Notably, with ResNet-101, the mIoU gap between the full set result (75.99%) and the 1/4 labeled setting result (75.15%) is only 0.84%. The performance of our method with ResNet-50 is surprisingly good in the 1/8 and 1/30 settings, suggesting that the deeper ResNet-101 backbone might be over-fitting in such settings due to the limited amount of labeled data.

COCO. For the COCO experiments, we mainly compare with the recent state-of-the-art PseudoSeg method [57] following their experiment protocols. Xception-65 [9], a stronger backbone than ResNet-101, is used. The results are shown in Tab. 4. We observe that our method *performs better in all data splits*. We even see gains in the full data setting, possibly because of the stronger data augmentation and regularization in our method. The VAL mIoU of the 1/8 labeled setting is quite promising, *almost catching the supervised learning result of the fully labeled data*.

4.3. Analyses

We report the ablation studies and diagnosis experiments in this section. Without special mentioning, all experiments are conducted with the ResNet-50 backbone and VOC 1/8 labeled split. The mIoUs are evaluated on the VOC VAL set.

Negative Sampling Strategy. One key technical contribution of our method is the negative sampling strategy. We measure the average False Negative Rates (FNR) during training with different negative sampling strategies in Tab. 5. We notice that FNR reduces with more complex sampling distributions. As expected, the Uniform strategy delivers the worst performance, suggesting that the noise introduced by false negative pixels may indeed hurt performance. The ideal case (marked as Oracle) pretends that the algorithm knows the actual ground-truth masks during contrastive learning and can be seen as an upper bound. Our best sampling strategy, Different Image + Pseudo Label, achieves an mIoU *almost as good as the Oracle version*.

Tab. 5: Quantitative comparison of negative sampling strategies in VOC 1/8 setting. We report the VAL mIoU and the False Negative Rates (FNR) of the sampled negative pixels. Oracle refers to the ideal case of using the ground-truth masks to guide the (uniform) negative pixel sampling.

Strategy	Unif	Diff	Unif+Pseu	Diff+Pseu	Oracle
mIoU (%)	62.70	63.33	64.38	64.63	64.78
FNR (%)	14.77	3.42	2.83	2.80	0

Tab. 6: Study on the number of sampled negative pixels. None refers to not using the feature-space pixel contrastive loss. As the number of negatives increases, the VAL mIoU rises, but the computation and the memory costs also rise. In our experiment, $N = 1,600$ actually causes the out-of-memory error. Therefore, we choose $N = 200$ in the main results as a trade-off between accuracy and efficiency.

N	None	50	100	200	400	800	1600
mIoU (%)	63.75	64.08	64.24	64.63	64.84	65.03	OOM

Number of Negative Pixels. Another importance factor in our pixel contrastive loss is the number of sampled negative pixels N. Previous work suggests that a large number of negatives may be beneficial to image-level contrastive learning [5, 19]. While we generally agree with this argument, using a large number of negative pixels can be costly in semantic segmentation. We believe that there should be a trade-off between efficiency and accuracy. In Tab. 6, we study the effect of the number of negative pixels. With the help of the negative sampling strategy to reduce false negative rates, our approach can already obtain competitive performance with only $N = 200$ negative pixels per anchor.

Loss Coefficients. We show the impact of the coefficients on the label consistency loss (Eq. 3) and the feature contrastive loss (Eq. 4) in Tab. 7. We found that $\lambda_1 = 0.3$ and $\lambda_2 = 1$ achieve the best result, therefore adopting these values in *all* other data splits and architectures. *The coefficients transfer well to other settings*. Another important observation with this hyper-parameter study is the *complementary nature of the label consistency and the feature contrastive properties*. We notice that the joint contrastive-consistent learning (64.63% when $\lambda_1 = 0.3, \lambda_2 = 1$) outperforms either the purely consistent version (63.75% when

Tab. 7: Effect of different combinations of unlabeled loss weights. As in Eq. 2, λ_1 (column) is the coefficient on the pixel contrastive loss in the feature space, λ_2 (row) is the coefficient on the consistency loss in the label space. Performance is measured by the VOC VAL mIoU when trained in the 1/8-labeled setting. $\lambda_1 = 0$ is the special case of applying only label consistency training, while $\lambda_2 = 0$ is the special case of applying only feature contrastive learning.

	$\lambda_1 = 0$	0.1	0.3	0.5	0.7	0.9
$\lambda_2 = 0$	49.57	51.92	52.29	52.28	52.74	53.53
$\lambda_2 = 0.5$	62.31	61.62	62.47	63.83	63.07	63.64
$\lambda_2 = 0.7$	63.46	63.26	63.71	63.79	63.35	63.43
$\lambda_2 = 1.0$	63.75	63.60	**64.63**	**64.48**	62.55	63.15
$\lambda_2 = 1.2$	61.33	64.48	64.30	64.16	63.08	63.03

Tab. 8: Training time comparison in the VOC 1/8 setting.

	Supervised	PseudoSeg	PC^{2}Seg consist-only	PC^{2}Seg
Time (min)	38	80	75∼80	80
VAL mIoU (%)	49.57	61.88	63.21	64.63

Tab. 9: Investigation of the variants that use different loss functions on the label space (row) and the feature space (column) with other hyper-parameters fixed. Overall, we observe that the ℓ_2 loss is more robust as a label-space consistency loss than the cross-entropy (CE) loss. In the 2nd row, the pixel contrastive loss outperforms the pixel consistency loss and the image-level contrastive loss.

mIoU (%)	None	Img Contrast	Pix Consist	Pix Contrast
Output CE	63.54	60.41	59.33	58.47
Output ℓ_2	63.75	62.49	63.21	**64.63**

$\lambda_1 = 0, \lambda_2 = 1$) or the best purely contrastive version (53.53% when $\lambda_1 = 0.9, \lambda_2 = 0$), supporting our insight.

Computational Cost. We report the training time comparison in Tab. 8. While achieving higher performance, the training time of PC^{2}Seg is comparable to the previous state-of-the-art semi-supervised method PseudoSeg [57]. If we remove the contrastive component of PC^{2}Seg, the time is reduced by less than 5min, suggesting that the extra cost of our pixel contrastive learning is low. We further show the cost reduced by our negative sampling strategy through a simple calculation in the supplementary material.

Comparison to Other Label-Space and Feature-Space Losses. In the label space, we compare two variants: the normalized ℓ_2 loss (Sec. 3) and the cross-entropy (CE) loss. In the feature space, we compare four variants: (1) no feature-space loss (None), (2) image contrastive loss [5], (3) pixel consistency loss, which is the normalized ℓ_2 distance between pixel features, and (4) pixel contrastive loss. Tab. 9 shows that the combination of the label ℓ_2 loss and the feature pixel contrastive loss achieves the best result.

Tab. 10: Varying the feature layer to apply the pixel contrastive loss. We report the VOC VAL mIoU in the 1/8 setting. Conv3-5 are the corresponding conv blocks of ResNet-50. Encoder refers to the feature map produced by the DeepLab ASPP net. Decoder refers to the final feature map right before classification. Conv5+Enc imposes a contrastive loss on two best-performing layers (Conv5 and Encoder), which does not bring extra gains over a single layer.

Layer	Conv3	Conv4	Conv5	Encoder	Decoder	Conv5+Enc
mIoU (%)	60.70	63.90	**64.63**	64.25	63.74	64.59

Tab. 11: Delaying the unlabeled loss until certain steps. The total number of steps is 30,000. Applying all losses jointly without delay achieves the best result.

Delayed Steps	0	2,000	4,000	8,000	12,000
mIoU (%)	**64.63**	63.30	62.67	62.56	62.50

Choice of Contrastive Learning Layer. In the main results, we choose to apply pixel contrastive learning on the last feature map of the backbone networks (Conv5 block of ResNet and ExitFlow2 block of Xception). We show results of applying it on other ResNet layers in Tab. 10. Earlier stage feature maps are larger in size and contain more low-level details, while later stage feature maps lose resolutions but contain more semantics. A mid-level intermediate layer yields the best performance.

Projection Layer. The projection layer performs dimensionality reduction on the feature vectors before contrasive learning. We find that using separate projections for the weak and strong branches yields higher mIoU than using a shared projection head (64.63% *vs.* 63.97%).

Delayed-Start of Semi-Supervised Learning. We mimic the two-stage variant that delays the start of semi-supervised learning until certain steps in Tab. 11. We notice that breaking the training into two stages in fact decreases the performance. Applying all losses jointly throughout the training (Delay = 0) achieves a better result in our experiments.

5. Conclusion

We propose a novel semi-supervised semantic segmentation approach based on feature-space contrastive learning and label-space consistency training. We also present the negative sampling techniques to improve the efficiency and effectiveness of pixel contrastive learning. Our approach outperforms existing methods on several semi-supervised segmentation benchmarks, suggesting that pixel contrastive-consistent learning is a promising research direction to improve semi-supervised semantic segmentation.

Acknowledgment: This work was supported in part by NSF Grant 2106825.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 3

[2] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *ECCV*, 2020. 3

[3] 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. *TPAMI*, 40(4):834–848, 2017. 1, 3

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1, 3, 4, 5, 6

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3, 4, 7, 8

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 2, 4

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2, 3, 4

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 6, 7

[10] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *NeurIPS*, 2020. 2, 5

[11] 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 *CVPR*, 2016. 1, 5

[12] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 3

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[14] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 1, 5

[16] Zhengyang Feng, Qianyu Zhou, Qiqi Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. DMT: Dynamic mutual training for semi-supervised learning. *arXiv preprint arXiv:2004.08514*, 2020. 5, 6

[17] Geoffrey French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *BMVC*, 2020. 1, 3, 5, 6

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 2, 4

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3, 4, 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[21] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019. 4

[22] Wei Chih Hung, Yi Hsuan Tsai, Yan Ting Liou, Yen-Yu Lin, and Ming Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *BMVC*, 2018. 3, 5, 6

[23] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. *arXiv preprint arXiv:2011.11765*, 2020. 2, 3, 4, 5

[24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017. 5

[25] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019. 3

[26] Zhanghan Ke, Kaican Li Di Qiu, Qiong Yan, and Rynson WH Lau. Guided collaborative training for pixel-wise semi-supervised learning. In *ECCV*, 2020. 1, 3, 6

[27] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 2, 4

[28] Jongmok Kim, Jooyoung Jang, and Hyunwoo Park. Structured consistency loss for semi-supervised semantic segmentation. *arXiv preprint arXiv:2001.04647*, 2020. 3

[29] Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The Gumbel-top-k trick for sampling sequences without replacement. In *ICML*, 2019. 5

[30] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013. 2

[31] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 3

[32] 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 *ECCV*, 2014. 1, 5

[33] Wenfeng Luo and Meng Yang. Semi-supervised semantic segmentation via strong-weak dual-branch network. In *ECCV*, 2020. 3

[34] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *TPAMI*, 43(4):1369–1379, 2021. 3, 5, 6

[35] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *TPAMI*, 41(8):1979–1993, 2018. 2, 6

[36] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *WACV*, 2021. 3, 5, 6

[37] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *CVPR*, 2020. 1, 3, 5, 6

[38] Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmaleck, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. In *NeurIPS*, 2020. 2

[39] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 2, 3

[40] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *ICCV*, 2017. 3, 5, 6

[41] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 3

[42] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 2, 6

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 4

[44] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *ICCV*, 2021. 2

[45] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020. 1, 3

[46] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, 2021. 2, 4

[47] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 2

[48] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 3

[49] Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. Conditional negative sampling for contrastive learning of visual representations. In *ICLR*, 2021. 2

[50] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020. 2

[51] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021. 2

[52] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4L: Self-supervised semi-supervised learning. In *ICCV*, 2019. 2

[53] Xiaohang Zhan, Ziwei Liu, Ping Luo, Xiaoou Tang, and Chen Loy. Mix-and-match tuning for self-supervised semantic segmentation. In *AAAI*, 2018. 3

[54] Xiangyun Zhao, Raviteja Vemulapalli, Philip Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. Contrastive learning for label-efficient semantic segmentation. *arXiv preprint arXiv:2012.06985*, 2020. 2

[55] Yi Zhu, Zhongyue Zhang, Chongruo Wu, Zhi Zhang, Tong He, Hang Zhang, R Manmatha, Mu Li, and Alexander Smola. Improving semantic segmentation via self-training. *arXiv preprint arXiv:2004.14960*, 2020. 3

[56] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. In *NeurIPS*, 2020. 1, 2, 3

[57] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. PseudoSeg: Designing pseudo labels for semantic segmentation. In *ICLR*, 2021. 1, 3, 4, 5, 6, 7, 8