Semi-Supervised Semantic Segmentation with Cross-Consistency Training

Yassine Ouali Céline Hudelot Myriam Tami
Université Paris-Saclay, CentraleSupélec, MICS, 91190, Gif-sur-Yvette, France.
{yassine.ouali,celine.hudelot,myriam.tami}@centralesupelec.fr

Abstract

In this paper, we present a novel cross-consistency based semi-supervised approach for semantic segmentation. Consistency training has proven to be a powerful semi-supervised learning framework for leveraging unlabeled data under the cluster assumption, in which the decision boundary should lie in low density regions. In this work, we first observe that for semantic segmentation, the low density regions are more apparent within the hidden representations than within the inputs. We thus propose cross-consistency training, where an invariance of the predictions is enforced over different perturbations applied to the outputs of the encoder. Concretely, a shared encoder and a main decoder are trained in a supervised manner using the available labeled examples. To leverage the unlabeled examples, we enforce a consistency between the main decoder predictions and those of the auxiliary decoders, taking as inputs different perturbed versions of the encoder's output, and consequently, improving the encoder's representations. The proposed method is simple and can easily be extended to use additional training signal, such as image-level labels or pixel-level labels across different domains. We perform an ablation study to tease apart the effectiveness of each component, and conduct extensive experiments to demonstrate that our method achieves state-of-the-art results in several datasets. Code is available at https://github.com/yassouali/CCT

1. Introduction

In recent years, with the wide adoption of deep supervised learning within the computer vision community, significant strides were made across various visual tasks yielding impressive results. However, training deep learning models requires a large amount of labeled data which acquisition is often costly and time consuming. In semantic segmentation, given how expensive and laborious the acquisition of pixel-level labels is, with a cost that is 15 times and 60 times larger than that of region-level and image-level labels respectively [33], the need for data efficient semantic segmentation methods is even more evident.

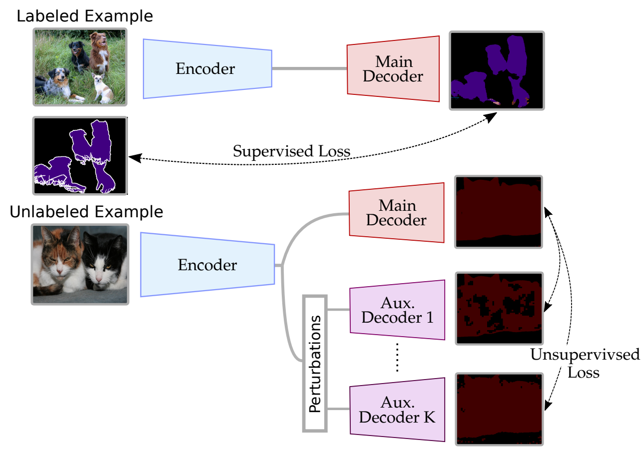

Figure 1. **The proposed Cross-Consistency training (CCT).** For the labeled examples, the encoder and the main decoder are trained in a supervised manner. For the unlabeled examples, a consistency between the main decoder's predictions and those of the auxiliary decoders is enforced, over different types of perturbations applied to the inputs of the auxiliary decoders. Best viewed in color.

As a result, a growing attention is drown on deep Semi-Supervised learning (SSL) to take advantage of a large amount of unlabeled data and limit the need for labeled examples. The current dominant SSL methods in deep learning are consistency training [43, 29, 50, 36], pseudo labeling [30], entropy minimization [17] and bootstrapping [42]. Some newly introduced techniques are based on generative modeling [28, 49].

However, the recent progress in SSL was confined to classification tasks, and its application in semantic segmentation is still limited. Dominant approaches [22, 53, 52, 31] focus on weakly-supervised learning which principle is to generate pseudo pixel-level labels by leveraging the weak labels, that can then be used, together with the limited strongly labeled examples, to train a segmentation network in a supervised manner. Generative Adversarial Networks (GANs) were also adapted for SSL setting [49, 23] by extending the generic GAN framework to pixel-level predictions. The discriminator is then jointly trained with an adversarial loss and a supervised loss over the labeled examples.

Nevertheless, these approaches suffer from some limitations. Weakly-supervised approaches require weakly labeled examples along with pixel-level labels, hence, they do not exploit the unlabeled data to extract additional training signal. Methods based on adversarial training exploit the unlabeled data, but can be harder to train.

To address these limitations, we propose a simple consistency based semi-supervised method for semantic segmentation. The objective in consistency training is to enforce an invariance of the model's predictions over small perturbations applied to the inputs. As a result, the learned model will be robust to such small changes. The effectiveness of consistency training depends heavily on the behavior of the data distribution, *i.e.*, the cluster assumption, where the classes must be separated by low density regions. In semantic segmentation, we do not observe the presence of low density regions separating the classes within the inputs, but rather within the encoder's outputs. Based on this observation, we propose to enforce the consistency over different forms of perturbations applied to the encoder's output. Specifically, we consider a shared encoder and a main decoder that are trained using the labeled examples. To leverage unlabeled data, we then consider multiple auxiliary decoders whose inputs are perturbed versions of the output of the shared encoder. The consistency is imposed between the main decoder's predictions and that of the auxiliary decoders (see Fig. 1). This way, the shared encoder's representation is enhanced by using the additional training signal extracted from the unlabeled data. The added auxiliary decoders have a negligible amount of parameters compared to the encoder. Additionally, during inference, only the main decoder is used, reducing the computation overhead both in training and inference.

The proposed method is simple and efficient, it is also flexible since it can easily be extended to use additional weak labels and pixel-level labels across different domains in a semi-supervised domain adaption setting. With extensive experiments, we demonstrate the effectiveness of our approach on PASCAL VOC [12] in a semi-supervised setting, and CityScapes, CamVid [3] and SUN [48] in a semi-supervised domain adaption setting. We obtain competitive results across different datasets and training settings.

Concretely, our contributions are four-fold:

- We propose a cross-consistency training (CCT) method for semi-supervised semantic segmentation, where the invariance of the predictions is enforced over different perturbations injected into the encoder's output.
- We propose and conduct an exhaustive study of various types of perturbations.
- We extend our approach to use weakly-labeled data, and exploit pixel-level labels across different domains to jointly train the segmentation network.
- We demonstrate the effectiveness of our approach with with an extensive and detailed experimental results, including a comparison with the state-of-the-art, as well as an in-depth analysis of our approach with a detailed ablation study.

2. Related Work

Semi-Supervised Learning. Recently, many efforts have been made to adapt classic SSL methods to deep learning, such as pseudo labeling [30], entropy minimization [17] and graph based methods [34, 26] in order to overcome this weakness. In this work, we focus mainly on consistency training. We refer the reader to [5] for a detailed overview of the field. Consistency training methods are based on the assumption that, if a realistic form of perturbation was applied to the unlabeled examples, the predictions should not change significantly. Favoring models with decision boundaries that reside in low density regions, giving consistent predictions for similar inputs. For example, Π-Model [29] enforces a consistency over two perturbed versions of the inputs under different data augmentations and dropout. A weighted moving average of either the previous predictions (*i.e.*, Temporal Ensembling [29]), or the model's parameters (*i.e.*, Mean Teacher [50]), can be used to obtain more stable predictions over the unlabeled examples. Instead of relying on random perturbations, Virtual Adversarial Training (VAT) [36] approximates the perturbations that will alter the model's predictions the most.

Similarly, the proposed method enforces a consistency of predictions between the main decoder and the auxiliary decoders over different perturbations, that are applied to the encoder's outputs rather than the inputs. Our work is also loosely related to Multi-View learning [60] and Cross-View training [7], where each input to the auxiliary decoders can be view as an alternate, but corrupt representation of the unlabeled examples.

Semi-Supervised Semantic Segmentation. A significant number of approaches use a limited number pixel-level labels together with a larger number of inexact annotations, *e.g.*, region-level [47, 9] or image-level labels [31, 61, 53, 32]. For image-level based weak-supervision, primary localization maps are generated using class activation mapping (CAM) [61]. After refining the generated maps, they can then be used to train a segmentation network together with the available pixel-level labels in a SSL setting.

Generative modeling can also be used for semi-supervised semantic segmentation [49, 23] to take advantage of the unlabeled examples. Under a GAN framework, the discriminator's predictions are extended over pixel classes, and can then be jointly trained with a Cross-Entropy loss over the labeled examples and an adversarial loss over the whole dataset.

In comparison, the proposed method exploits the unlabeled examples by enforcing a consistency over multiple perturbations on the hidden representations level. Enhancing the encoder's representation and the overall performance, with a small additional cost in terms of computation and memory requirements.

Recently, CowMix [13], a concurrent method was introduced. CowMix, using MixUp [56], enforces a consistency between the mixed outputs and the prediction over the mixed inputs. In this context, CCT differs as follows: (1) CowMix, as traditional consistency regularization methods, applies the perturbations to the inputs, but uses MixUp as a high-dimensional perturbation to overcome the absence of the cluster assumption. (2) Requires multiple forward passes though the network for one training iteration. (3) Adapting CowMix to other settings (*e.g.*, over multiple domains, using weak labels) may require significant changes. CCT is efficient and can easily be extended to other settings.

Domain Adaptation. In many real world cases, the existing discrepancy between the distribution of training data and and that of testing data will often hinder the performances. Domain adaptation aims to rectify this mismatch and tune the models for a better generalization at test time [40]. Various generative and discriminative domain adaptation methods have been proposed for classification [16, 14, 15, 4] and semantic segmentation [21, 58, 44, 24] tasks.

In this work, we show that enforcing a consistency across different domains can push the model toward better generalization, even in the extreme case of non-overlapping label spaces.

3. Method

3.1. The cluster assumption in semantic segmentation

We start with our observation and analysis of the cluster assumption in semantic segmentation, motivating the proposal of our cross-consistency training approach. A simple way to examine it is to estimate the local smoothness by measuring the local variations between the value of each pixel and its local neighbors. To this end, we compute the average euclidean distance at each spatial location and its 8 intermediate neighbors, for both the inputs and the hidden representations (*i.e.*, the ResNet's [20] outputs of a DeepLab v3 [6] trained on COCO [33]). For the inputs, following [13], we compute the average distance of a patch centered at a given spatial location and its neighbors to simulate a realistic receptive field. For the hidden representations, we first upsample the feature map to the input size, and then compute the average distance between the neighboring activations (2048-dimensional feature vectors). The results are illustrated in Fig. 2. We observe that the cluster assumption is violated at the input level, given that the low density regions do not align with the class boundaries. On the contrary, for the encoder's outputs, the cluster assumption is maintained where the class boundaries have high average distance, thus corresponding to low density regions. This observation motivates the following approach, in which the perturbations are applied to the encoder's outputs rather than the inputs.

3.2. Cross-Consistency Training for semantic segmentation

3.2.1 Problem Definition

In SSL, we are provided with a small set of labeled training examples and a larger set of unlabeled training examples. Let $\mathcal{D}_l = \{(\mathbf{x}_1^l, y_1), \ldots, (\mathbf{x}_n^l, y_n)\}$ represent the n labeled examples and $\mathcal{D}_u = \{\mathbf{x}_1^u, \ldots, \mathbf{x}_m^u\}$ represent the m unlabeled examples, with $\mathbf{x}_i^u$ as the i-th unlabeled input image, and $\mathbf{x}_i^l$ as the i-th labeled input image with spatial dimensions $H \times W$ and its corresponding pixel-level label $y_i \in \mathbb{R}^{C \times H \times W}$, where C is the number of classes.

As discussed in the introduction, the objective is to exploit the larger number of unlabeled examples ($m \gg n$) to train a segmentation network f, to perform well on the test data drawn from the same distribution as the training data. In this work, our architecture (see Fig. 3) is composed of a shared encoder h and a main decoder g, which constitute the segmentation network $f = g \circ h$. We also introduce a set of K auxiliary decoders g_a^k, with $k \in [1, K]$. While the segmentation network f is trained on the labeled set $\mathcal{D}_l$ in a traditional supervised manner, the auxiliary networks $g_a^k \circ h$ are trained on the unlabeled set $\mathcal{D}_u$ by enforcing a consistency of predictions between the main decoder and the auxiliary decoders. Each auxiliary decoder takes as input a perturbed version of the encoder's output, and the main encoder is fed the uncorrupted intermediate representation. This way, the representation learning of the encoder h is further enhanced using the unlabeled examples, and subsequently, that of the segmentation network f.

3.2.2 Cross-Consistency Training

As stated above, to extract additional training signal from the unlabeled set $\mathcal{D}_u$, we rely on enforcing a consistency between the outputs of the main decoder g_m and those of auxiliary decoders g_a^k. Formally, for a labeled training example $\mathbf{x}_i^l$, and its pixel-level label y_i, the segmentation network f is trained using a Cross-Entropy (CE) based supervised loss $\mathcal{L}_s$:

$$\mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{\mathbf{x}_i^l, y_i \in \mathcal{D}_l} \mathbf{H}(y_i, f(\mathbf{x}_i^l)) \tag{1}$$

with $\mathbf{H}(.,.)$ as the CE. For an unlabeled example $\mathbf{x}_i^u$, an intermediate representation of the input is computed using

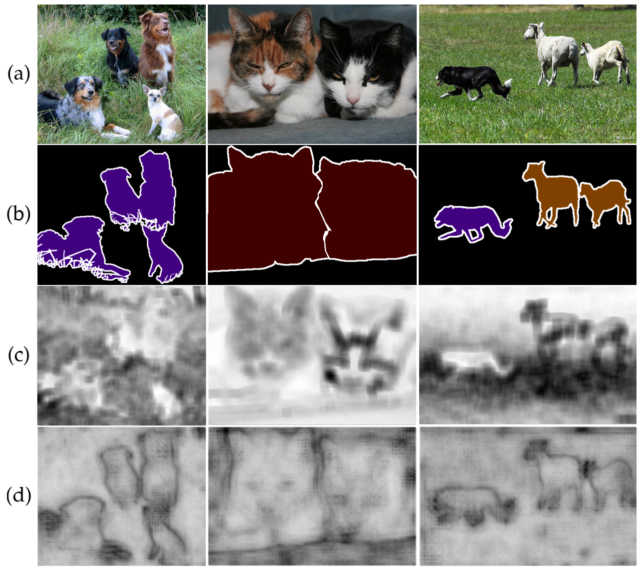

Figure 2. **The cluster assumption in semantic segmentation.** (a) Examples from PASCAL VOC 2012 *train set*. (b) Pixel-level labels. (c) *Input level.* The average euclidean distance between each patch of size 20×20 centered at a given spatial location extracted from the input images, and its 8 neighboring patches. (d) *Hidden representations level.* The average euclidean distance between a given 2048-dimensional activation at each spatial location and its 8 neighbors. Darkest regions indicate high average distance.

the shared encoder $\mathbf{z}_i = h(\mathbf{x}_i^u)$*. Let us consider R stochastic perturbations functions, denoted as p_r with $r \in [1, R]$, where one perturbation function can be assigned to multiple auxiliary decoders. With various perturbation settings, we generate K perturbed versions $\tilde{\mathbf{z}}_i^k$ of the intermediate representation $\mathbf{z}_i$, so that the k-th perturbed version is to be fed to the k-th auxiliary decoder. For consistency, we consider the perturbation function as part of the auxiliary decoder, (*i.e.*, g_a^k can be seen as $g_a^k \circ p_r$). The training objective is then to minimize the unsupervised loss $\mathcal{L}_u$, which measures the discrepancy between the main decoder's output and that of the auxiliary decoders:

$$\mathcal{L}_u = \frac{1}{|\mathcal{D}_u|} \frac{1}{K} \sum_{\mathbf{x}_i^u \in \mathcal{D}_u} \sum_{k=1}^{K} \mathbf{d}(g(\mathbf{z}_i), g_a^k(\mathbf{z}_i)) \quad (2)$$

with $\mathbf{d}(.,.)$ as a distance measure between two output probability distributions (*i.e.*, the outputs of a $softmax$ function applied over the channel dimension). In this work, we choose to use mean squared error (MSE) as a distance measure.

The combined loss $\mathcal{L}$ for consistency based SSL is then computed as:

$$\mathcal{L} = \mathcal{L}_\text{s} + \omega_u \mathcal{L}_\text{u} \quad (3)$$

* Throughout the paper, $\mathbf{z}$ always refers to the output of the encoder corresponding to an unlabeled input image $\mathbf{x}^u$.

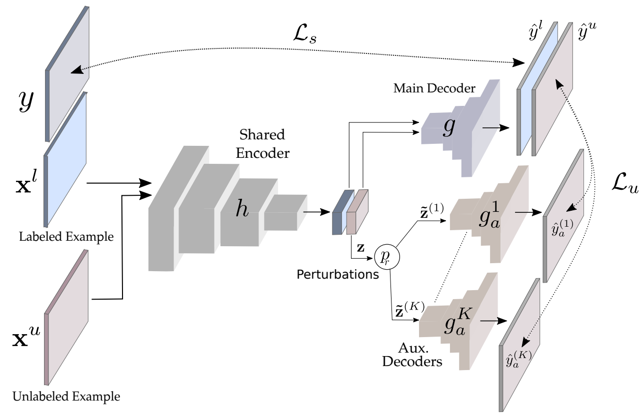

Figure 3. **Illustration of our approach.** For one training iteration, we sample a labeled input image $\mathbf{x}^l$ and its pixel-level label y together with an unlabeled image $\mathbf{x}^u$. We pass both images through the encoder and main decoder, obtaining two main predictions $\hat{y}^l$ and $\hat{y}^u$. We compute the supervised loss using the pixel-level label y and $\hat{y}^l$. We apply various perturbations to $\mathbf{z}$, the output of the encoder for $\mathbf{x}^u$, and generate auxiliary predictions $\hat{y}_a^{(i)}$ using the perturbed versions $\tilde{\mathbf{z}}^{(i)}$. The unsupervised loss is then computed between the outputs of the auxiliary decoders and that of the main decoder.

where ω_u is an unsupervised loss weighting function. Following [29], to avoid using the initial noisy predictions of the main encoder, ω_u ramps up starting from zero along a Gaussian curve up to a fixed weight λ_u. Concretely, at each training iteration, an equal number of examples are sampled from the labeled $\mathcal{D}_l$ and unlabeled $\mathcal{D}_u$ sets. The supervised loss is computed using the main encoder's output and pixel-level labels. For the unlabeled examples, we compute the MSE between the prediction of each auxiliary decoder and that of the main decoder. The total loss is then compute and back-propagated to train the segmentation network f and the auxiliary networks $g_a^k \circ h$. Note that the unsupervised loss $\mathcal{L}_u$ is not back-propagated through the main-decoder g, only the labeled examples are used to train g.

3.2.3 Perturbation functions

An important factor in consistency training is the perturbations to apply to the hidden representation, *i.e.*, the encoder's output $\mathbf{z}$. We propose three types of perturbation functions p_r: feature based, prediction based and random.

Feature based perturbations. They consist of either injecting noise into or dropping some of the activations of encoder's output feature map $\mathbf{z}$.

- F-Noise: we uniformly sample a noise tensor $\mathbf{N} \sim \mathcal{U}(-0.3, 0.3)$ of the same size as $\mathbf{z}$. After adjusting its amplitude by multiplying it with $\mathbf{z}$, the noise is then injected into the encoder's output $\mathbf{z}$ to get $\tilde{\mathbf{z}} = (\mathbf{z} \odot \mathbf{N}) + \mathbf{z}$. This way, the injected noise is proportional to each activation.

- F-Drop: we first uniformly sample a threshold $\gamma \sim \mathcal{U}(0.6, 0.9)$. After summing over the channel dimension and normalizing the feature map $\mathbf{z}$ to get $\mathbf{z}'$, we generate a mask $\mathbf{M}_{\text{drop}} = \{\mathbf{z}' < \gamma\}_{\mathbf{1}}$[†], which is then used to obtain the perturbed version $\tilde{\mathbf{z}} = \mathbf{z} \odot \mathbf{M}_{\text{drop}}$. This way, we mask 10% to 40% of the most active regions in the feature map.

Prediction based perturbations. They consist of adding perturbations based on the main decoder's prediction $\hat{y} = g(\mathbf{z})$ or that of the auxiliary decoders. We consider masking based perturbations (Con-Msk, Obj-Msk and G-Cutout) in addition to adversarial perturbations (I-VAT).

- Guided Masking: Given the importance of context relationships for complex scene understanding [37], the network might be too reliant on these relationships. To limit them, we create two perturbed versions of $\mathbf{z}$ by masking the detected objects (Obj-Msk) and the context (Con-Msk). Using $\hat{y}$, we generate an object mask $\mathbf{M}_{\text{obj}}$ to mask the detected foreground objects and a context mask $\mathbf{M}_{\text{con}} = 1 - \mathbf{M}_{\text{obj}}$, which are then applied to $\mathbf{z}$.
- Guided Cutout (G-Cutout): in order to reduce the reliance on specific parts of the objects, and inspired by Cutout [11] that randomly masks some parts of the input image, we first find the possible spatial extent (*i.e.*, bounding box) of each detected object using $\hat{y}$. We then zero-out a random crop within each object's bounding box from the corresponding feature map $\mathbf{z}$.
- Intermediate VAT (I-VAT): to further push the output distribution to be isotropically smooth around each data point, we investigate using VAT [36] as a perturbation function to be applied to $\mathbf{z}$ instead of the unlabeled inputs. For a given auxiliary decoder, we find the adversarial perturbation r_{adv} that will alter its prediction the most. The noise is then injected into $\mathbf{z}$ to obtain the perturbed version $\tilde{\mathbf{z}} = r_{adv} + \mathbf{z}$.

Random perturbations. (DropOut) Spatial dropout [51] is also applied to $\mathbf{z}$ as a random perturbation.

3.2.4 Practical considerations

A each training iteration, we sample an equal number of labeled and unlabeled samples. As a consequence, we iterate on the set $\mathcal{D}_l$ more times than on its unlabeled counterpart $\mathcal{D}_u$, thus risking an overfitting of the labeled set $\mathcal{D}_l$.

Avoiding Overfitting. Motivated by [41] who observed improved results by sampling only 6% of the hardest pixels, and [54] who showed an improvement when gradually releasing the supervised training signal in a SSL setting, we propose an annealed version of the bootstrapped-CE (ab-CE) in [41]. With an output $f(\mathbf{x}_i^l) \in \mathbb{R}^{C \times H \times W}$ in the form of a probability distribution over the pixels, we only compute the supervised loss over the pixels with a probability less than a threshold η:

$$\mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{\mathbf{x}_i^l, y_i \in \mathcal{D}_l} \{f(\mathbf{x}_i^l) < \eta\}_{\mathbf{1}} \mathbf{H}(y_i, f(\mathbf{x}_i^l)) \tag{4}$$

To release the supervised training signal, the threshold parameter η is gradually increased from $\frac{1}{C}$ to 0.9 during the beginning of training, with C as the number of output classes.

3.3. Exploiting weak-labels

In some cases, we might be provided with additional training data that is less expensive to acquire compared to pixel-level labels, *e.g.*, image-level labels. Formally, instead of an unlabeled set $\mathcal{D}_u$, we are provided with a weakly labeled set $\mathcal{D}_w = \{(\mathbf{x}_1^w, y_1^w), \dots, (\mathbf{x}_m^w, y_m^w)\}$ alongside a pixel-level labeled set $\mathcal{D}_l$, with y_i^w is the i-th image-level label corresponding to the i-th weakly labeled input image $\mathbf{x}_i^w$. The objective is to extract additional information from the weak labeled set $\mathcal{D}_w$ to further enhance the representations of the encoder h. To this end, we add a classification branch g_c consisting of a global average pooling layer followed by a classification layer, and pretrain the encoder for a classification task using binary CE loss.

Following previous works [1, 31, 22], the pretrained encoder and the added classification branch can then be exploited to generate pseudo pixel-level labels y_p. We start by generating the CAMs M as in [61]. Using $M \in \mathbb{R}^{C \times H \times W}$, we can then generate pseudo labels y_p, with a background θ_{bg} and a foreground θ_{fg} thresholds. The pixels with attention scores less than θ_{bg} (*e.g.*, 0.05) are considered as background. For the pixels with an attention score larger than θ_{fg} (*e.g.*, 0.30), they are assigned the class with the maximal attention score, and the rest of the pixels are ignored. After generating y_p, we conduct a final refinement step using dense CRF [27].

In addition to considering $\mathcal{D}_w$ as an unlabeled set and imposing a consistency over its examples, the pseudo-labels are used to train the auxiliary networks $g_a^k \circ h$ using a weakly supervised loss $\mathcal{L}_w$. In this case, the loss in Eq. (3) becomes:

$$\mathcal{L} = \mathcal{L}_{\text{s}} + \omega_u \mathcal{L}_u + \omega_w \mathcal{L}_w \tag{5}$$

With

$$\mathcal{L}_w = \frac{1}{|\mathcal{D}_w|} \frac{1}{K} \sum_{\mathbf{x}_i^w \in \mathcal{D}_w} \sum_{k=1}^{K} \mathbf{H}(y_p, g_a^k(\mathbf{z}_i)) \tag{6}$$

[†] $\{\text{condition}\}_{\mathbf{1}}$ is a boolean function outputting 1 if the condition is true, 0 otherwise.

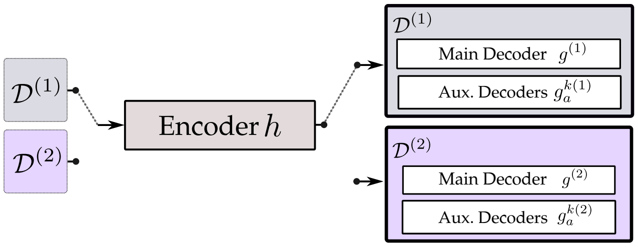

Figure 4. **CCT on multiple domains**. On top of a shared encoder, we add domain specific main decoder and K auxiliary decoders. During training, we alternate between the two domains, sampling labeled and unlabeled examples and training the corresponding decoders and the shared encoder at each iteration.

3.4. Cross-Consistency Training on Multiple Domains

In this section, we extend the propose framework to a semi-supervised domain adaption setting. We consider the case of two datasets $\{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}\}$ with partially or fully non-overlapping label spaces, each one contains a set of labeled and unlabeled examples $\mathcal{D}^{(i)} = \{\mathcal{D}_l^{(i)}, \mathcal{D}_u^{(i)}\}$. The objective is to simultaneously train a segmentation network to do well on the test data of both datasets, which is drown from the different distributions.

Our assumption is that enforcing a consistency over both unlabeled sets $\mathcal{D}_u^{(1)}$ and $\mathcal{D}_u^{(2)}$ might impose an invariance of the encoder's representations across the two domains. To this end, on top of the shared encoder h, we add domain specific main decoder $g^{(i)}$ and auxiliary decoders $g_a^{k(i)}$. Specifically, as illustrated in Fig. 4, we add two main decoders and $2K$ auxiliary decoders on top of the encoder h. During training, we alternate between the two datasets, at each iteration, sampling an equal number of labeled and unlabeled examples from each one, computing the loss in Eq. (3) and training the shared encoder and the corresponding main and auxiliary decoders.

4. Experiments

To evaluate the proposed method and investigate its effectiveness in different settings, we carry out detailed experiments. In Section 4.4, we present an extensive ablation study to highlight the contribution of each component within the proposed framework, and compare it to state-of-the-art methods in a semi-supervised setting. Additionally, in Section 4.5 we apply the proposed method in a semi-supervised domain adaptation setting and show performance above baseline methods.

4.1. Network Architecture

Encoder. For the following experiments, the encoder is based on a ResNet-50 [20] pretrained on ImageNet [10] provided by [55] and a PSP module [59]. Following previous works [59, 22, 1], the last two strided convolutions of ResNet are replaced with dilated convolutions.

Decoders. For the decoders, taking the efficiency and the number of parameters into consideration, we choose to only use 1×1 convolutions. After an initial 1×1 convolution to adapt the depth to the number of classes C, we apply a series of three sub-pixel convolutions [45] with ReLU non-linearities to upsample the outputs to original input size.

4.2. Datasets and Evaluation Metrics

Datasets. In a semi-supervised setting, we evaluate the proposed method on PASCAL VOC [12], consisting of 21 classes (with the background included) and three splits, training, validation and testing, with of 1464, 1449 and 1456 images respectively. Following the common practice [22, 59], we augment the training set with additional images from [19]. Note that the pixel-level labels are only extracted from the original training set.

For semi-supervised domain adaption, for partially overlapping label spaces, we train on both Cityscapes [8] and CamVid [3]. Cityscapes is a finely annotated autonomous driving dataset with 19 classes. We are provided with three splits, training, validation and testing with 2975, 500 and 1525 images respectively. CamVid contains 367 training, 101 validation and 233 testing images. Although originally the dataset is labeled with 38 classes, we use the 11 classes version [2]. For experiments over non-overlapping labels spaces, we train on Cityscapes and SUN RGB-D [48]. SUN RGB-D is an indoor segmentation dataset with 38 classes containing two splits, training and validation, with 5285 and 5050 images respectively. Similar to [24], we train on the 13 classes version [18].

Evaluation Metrics. We report the results using mIoU (*i.e.*, mean of class-wise intersection over union) for all the datasets.

4.3. Implementation Details

Training Settings. The implementation is based on the PyTorch 1.1 [39] framework. For optimization, we train for 50 epochs using SGD with a learning rate of 0.01 and a momentum of 0.9. During training, the learning rate is annealed following the *poly* learning rate policy, where at each iteration, the base learning rate is multiplied by $1 - (\frac{\text{iter}}{\text{max\_iter}})^{\text{power}}$ with power $= 0.9$.

For PASCAL VOC, we take crops of size 321×321 and apply random rescaling in the range of $[0.5, 2.0]$ and random horizontal flip. For Cityscapes, Cam-Vid and SUN RGB-D, following [24, 23], we resize the input images to 512×1024, 360×480 and 480×640 respectively, without any data-augmentation.

Reproducibility All the experiments are conducted on a V-100 GPUs. The implementation is available at https://github.com/yassouali/CCT

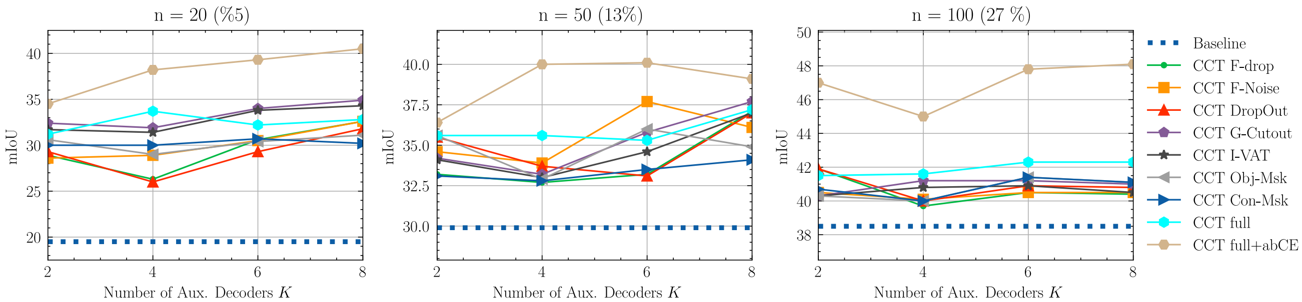

Figure 5. **Ablation Studies on CamVid with 20, 50 and 100 labeled images**. With different types of perturbations and a variable number of auxiliary decoders K, we compare the individual and the combined effectiveness of the perturbations to the baseline in which the model is trained only on the labeled examples. CCT full refers to using all of the 7 perturbations, *i.e.* the number of auxiliary decoder is $K \times 7$.

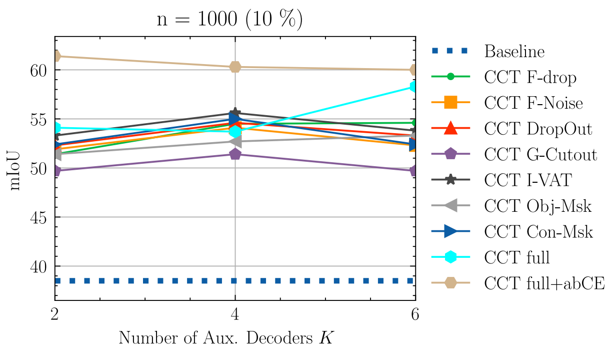

Figure 6. **Ablation study on PASCAL VOC**. Ablation study results with 1000 labeled examples using different perturbations and various numbers of auxiliary decoders K.

Method	Pixel-level Labeled Examples	Image-level Labeled Examples	Val
WSSL [38]	1.5k	9k	64.6
GAIN [32]	1.5k	9k	60.5
MDC [53]	1.5k	9k	65.7
DSRG [22]	1.5k	9k	64.3
Souly *et al.* [49]	1.5k	9k	65.8
FickleNet [31]	1.5k	9k	65.8
Souly *et al.* [49]	1.5k	-	64.1
Hung *et al.* [23]	1.5k	-	68.4
CCT	1k	-	64.0
CCT	1.5k	-	69.4
CCT	1.5k	9k	**73.2**

Table 1. **Comparison with the-state-of-the-art.** CCT performance on PASCAL VOC compared to other semi-supervised approaches.

Inference Settings. For PASCAL VOC, Cityscapes and SUN RGB-D, we report the results obtained on the validation set, and on the test set of CamVid dataset.

4.4. Semi-Supervised Setting

4.4.1 Ablation Studies

The proposed method consists of several types of perturbations and a variable number of auxiliary decoders. We thus start by studying the effect of the perturbation functions with different numbers of auxiliary decoders, in order to provide additional insight into their individual performance and their combined effectiveness. Specifically, we measure the effect of different numbers of auxiliary decoders K (*i.e.*, $K = 2$, 4, 6 and 8) of a given perturbation type. We refer to this setting of our method as "CCT {*perturbation type*}", with seven possible perturbations. We also measure the combined effect of all perturbations resulting in $K \times 7$ auxiliary decoders in total, and refer to it as "CCT full". Additionally, "CCT full+ab-CE" indicates the usage of the annealed-bootstrapped CE as a supervised loss function. We compare them to the baseline, in which the model is trained only using the labeled examples.

CamVid. We carried out the ablation on CamVid with 20, 50 and 100 labels; the results are shown in Fig. 5. We find that each perturbation outperforms the baseline, with the most dramatic differences in the 20-label setting with up to 21 points. We also surprisingly observe an insignificant overall performance gap among different perturbations, confirming the effectiveness of enforcing the consistency over the hidden representations for semantic segmentation, and highlighting the versatility of CCT and its success with numerous perturbations. Increasing K results in a modest improvement overall, with the smallest change for Con-Msk and Obj-Msk due to their lack of stochasticity. Interestingly, we also observe a slight improvement when combining all of the perturbations, indicating that the encoder is able to generate representations that are consistent over many perturbations, and subsequently, improving the overall performance. Additionally, gradually releasing the training signal using ab-CE helps increase the performance with up to 8%, which confirms that overfitting of the labeled examples can cause a significant drop in performance.

PASCAL VOC. In order to investigate the success of CCT on larger datasets, we conduct additional ablation experiments on PASCAL VOC using 1000 labeled examples, The results are summarized in Fig. 6. We see similar results, where the proposed method makes further improvement compared to the baseline with different perturbations, from 10 to 15 points. The combined perturbations yield a small increase in the performance, with the biggest difference with $K = 6$. Furthermore, similar to CamVid, when using the ab-CE loss, we see a significant gain with up to 7 points compared to CCT full.

Based on the conducted ablation studies, for the rest of the experiments, we use the setting of "CCT full" with $K = 2$ for $\mathrm{Con\text{-}Msk}$ and $\mathrm{Obj\text{-}Msk}$ due to their lack of stochasticity, $K = 2$ for I-VAT given its high computational cost, and $K = 6$ for the rest of the perturbations, and refer to it as "CCT".

4.4.2 Comparison to Previous Work

To further explore the effectiveness of our framework, we quantitatively compare it with previous semi-supervised semantic segmentation methods on PASCAL VOC. Table 1 compares CCT with other semi-supervised approaches. Our approach outperforms previous works relying on the same level of supervision and even methods which exploit image-level labels. We also observe an increase of 3.8 points when using additional image-level labels, affirming the flexibility of CCT, and the possibility of using it with different types of labels without any learning conflicts.

4.5. Semi-Supervised Domain Adaptation Setting

In real world applications, we are often provided with pixel-level labels collected from various sources, thus distinct data distributions. To examine the effectiveness of CCT when applied to multiple domains with a variable degree of labels overlap, we train our model simultaneously on two datasets, Cityscapes (CS) + CamVid (CVD) for partially overlapping labels, and Cityscapes + SUN RGB-D (SUN) for the disjoint case.

Method	n=50			n=100		
	CS	CVD	Avg.	CS	CVD	Avg.
Kalluri, *et al*. [24]	34.0	53.2	43.6	41.0	54.6	47.8
Baseline	31.2	40.0	35.6	37.3	34.4	35.9
CCT	35.0	53.7	**44.4**	40.1	55.7	**47.9**

Table 2. **CCT applied to CS+CVD.** CCT performance when simultaneously trained on two datasets with overlapping label spaces, which are Cityscapes (CS) and CamVid (CVD).

Cityscapes + CamVid. The results for CCT on Cityscapes and CamVid datasets with 50 and 100 labeled examples are given in Table 2. Similar to the SSL setting, CCT outperforms the baseline significantly, where the model is iteratively trained using only on the labeled examples, with up to 12 points for $n = 100$, we even see a modest increase compared to previous work. This confirms our hypothesis that enforcing a consistency over different datasets does indeed push the encoder to produce invariant representation across different domains, and consequently, increases the performance over the baseline while delivering similar results on each domain individually.

Method	Labeled Examples	CS	SUN	Avg.
SceneNet [35]	Full (5.3k)	-	49.8	-
Kalluri, *et al*. [24]	1.5k	58.0	31.5	44.8
Baseline	1.5k	54.3	38.1	46.2
CCT	1.5k	58.8	45.5	**52.1**

Table 3. **CCT applied to CS+CVD.** CCT performance when trained on both datasets Cityscapes (CS) and SUN RGB-D (SUN) datasets, for the case of non-overlapping label spaces.

Cityscapes + SUN RGB-D. For cross domain experiments, where the two domains have distinct labels spaces, we train on both Cityscapes and SUN RGB-D to demonstrate the capability of CCT to extract useful visual relationships and perform knowledge transfer between dissimilar domains, even in completely different settings. The results are shown in Table 2. Interestingly, despite the distribution mismatch between the datasets, and the high number of labeled examples ($n = 1500$), CCT still provides a meaningful boost over the baseline with 5.9 points difference and 7.3 points compared to previous work. Showing that, by enforcing a consistency of predictions on the unlabeled sets of the two datasets over different perturbations, we can extract additional training signal and enhance the representation learning of the encoder, even in the extreme case with non-overlapping label spaces, without any performance drop when an invariance of representations across both datasets is enforced at the level of encoder's outputs.

5. Conclusion

In this work, we present cross-consistency training (CCT), a simple, efficient and flexible method for a consistency based semi-supervised semantic segmentation, yielding state-of-the-art results. For future works, a possible direction is exploring the usage of other perturbations to be applied at different levels within the segmentation network. It would also be interesting to adapt and examine the effectiveness of CCT in other visual tasks and learning settings, such as unsupervised domain adaptation.

Acknowledgements. This work was supported by Ranstad corporate research chair. We would also like to thank Saclay-IA plateform of Université Paris-Saclay and Mésocentre computing center of CentraleSupélec and École Normale Supérieure Paris-Saclay for providing the computational resources.

References

[1] 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.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[3] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[4] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[5] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[7] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*, 2018.

[8] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[9] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[13] Geoff French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham Finlayson. Consistency regularization and cutmix for semi-supervised semantic segmentation. *arXiv preprint arXiv:1906.01916*, 2019.

[14] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[16] Bo Geng, Dacheng Tao, and Chao Xu. Daml: Domain adaptation metric learning. *IEEE Transactions on Image Processing*, 20(10):2980–2989, 2011.

[17] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[18] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077–4085, 2016.

[19] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011.

[20] 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.

[21] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[22] 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.

[23] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

[24] Tarun Kalluri, Girish Varma, Manmohan Chandraker, and CV Jawahar. Universal semi-supervised semantic segmentation. *arXiv preprint arXiv:1811.10323*, 2018.

[25] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Two-phase learning for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3534–3543, 2017.

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[27] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[28] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, pages 5534–5544, 2017.

[29] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning, 2016.

[30] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[31] 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 Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019.

[32] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[33] 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. *Lecture Notes in Computer Science*, page 740–755, 2014.

[34] Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. *arXiv preprint arXiv:1812.08781*, 2018.

[35] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2678–2687, 2017.

[36] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[37] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007.

[38] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015.

[39] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[40] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

[41] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4151–4160, 2017.

[42] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–152, 2018.

[43] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

[44] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision*, pages 86–103. Springer, 2018.

[45] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[46] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553. IEEE, 2017.

[47] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation, 2019.

[48] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[49] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5688–5696, 2017.

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

[51] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[52] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[53] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[54] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

[55] Ansheng You, Xiangtai Li, Zhen Zhu, and Yunhai Tong. Torchcv: A pytorch-based framework for deep learning in computer vision. https://github.com/donnyyou/torchcv, 2019.

[56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[57] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, 2018.

[58] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2030, 2017.

[59] 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.

[60] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017.

[61] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.