A Closer Look at Self-training for Zero-Label Semantic Segmentation

Giuseppe Pastore[1], Fabio Cermelli[1,2], Yongqin Xian[3],
Massimiliano Mancini[4], Zeynep Akata[3,4,5], Barbara Caputo[1,2]
[1]Politecnico di Torino, [2]Italian Institute of Technology, [3]MPI for Informatics,
[4]University of Tübingen, [5]MPI for Intelligent Systems

Abstract

Being able to segment unseen classes not observed during training is an important technical challenge in deep learning, because of its potential to reduce the expensive annotation required for semantic segmentation. Prior zero-label semantic segmentation works approach this task by learning visual-semantic embeddings or generative models. However, they are prone to overfitting on the seen classes because there is no training signal for them. In this paper, we study the challenging generalized zero-label semantic segmentation task where the model has to segment both seen and unseen classes at test time. We assume that pixels of unseen classes could be present in the training images but without being annotated. Our idea is to capture the latent information on unseen classes by supervising the model with self-produced pseudo-labels for unlabeled pixels. We propose a consistency regularizer to filter out noisy pseudo-labels by taking the intersections of the pseudo-labels generated from different augmentations of the same image. Our framework generates pseudo-labels and then retrain the model with human-annotated and pseudo-labelled data. This procedure is repeated for several iterations. As a result, our approach achieves the new state-of-the-art on PascalVOC12 and COCO-stuff datasets in the challenging generalized zero-label semantic segmentation setting, surpassing other existing methods addressing this task with more complex strategies. Code can be found at https://github.com/giuseppepastore10/STRICT.

1. Introduction

Tremendous progress has been made in semantic segmentation by deep learning [27, 11, 49] on large human-annotated datasets [8, 13]. As this requires expensive pixel-wise annotations [13], reducing the pixel-level supervision becomes important, e.g. weakly supervised [37, 19, 25, 32, 23, 42, 38] and few-shot [42, 38] learning. In the extreme case, the task is zero-label semantic segmentation [44, 7, 16] and the goal is to segment the novel classes not annotated during training. One major limitation of zero-label semantic segmentation is that the model is only evaluated on unseen classes. This is not realistic as any class could be present at test time. In generalized zero-label semantic segmentation (GZLSS) the model is required to segment both seen and unseen classes. GZLSS is challenging because it suffers from the severe class-imbalanced issue, leading to a significant performance drop on unseen classes. SPNet [44] fixes this issue by reducing the prediction scores of seen classes by a constant factor γ that is sensitive and hard to tune. CaGNet [16] and ZS3 [7] propose to learn generative models that synthesize features of unseen classes. Nevertheless, the generated features may have domain shift issues because the true distribution is never observed. It is also worth noting that the training set includes many unlabeled pixels from unseen classes due to the large amount of class co-occurrences. While those unlabeled pixels contain complementary information about unseen classes, SPNet [44] simply ignores them during training.

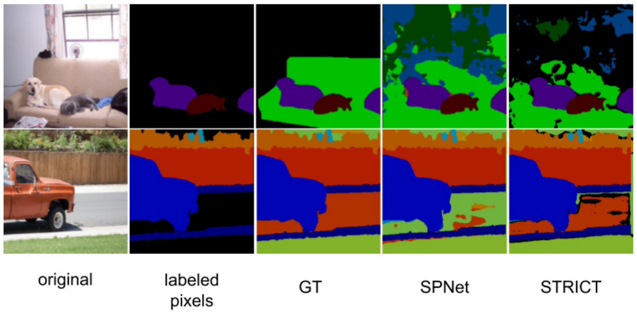


Figure 1: In generalized zero-label semantic segmentation, some unseen class pixels are ignored although they might be relevant at test-time. We approximate the ground truth by pseudo-labeling the unlabelled pixels, i.e. our Self-Training with Consistency Constraint (STRICT) is effective. In the Figure, *labeled pixels* and *GT* refers respectively to the masked and actual ground truth, *SPNet* and *STRICT* to the pseudo-labeled masks produced by SPNet and our method.

We propose to capture the latent information about unseen classes by supervising the model with self-produced pseudo-labels for the unlabeled pixels. Pseudo-labeling is not a novel concept, and it has been used as a self-training strategy for semi-supervised learning [22, 3, 35, 39, 18]. However, generating accurate pseudo-labels for unseen classes in semantic segmentation is difficult because the accuracy on unseen classes is often much lower than in the supervised case. Consequently, pseudo-labels generated from a pretrained GZLSS model (e.g., SPNet [44]) become noisy and may hurt the performance (as shown in Figure 1). To this end, we introduce an efficient consistency constraint to reduce the noise of pseudo-labels. The key assumption is that a pseudo-label is more likely to be correct if the model predicts the same label when presented different augmented versions of the same image. While being agnostic to complex hyperparameters tuning and robust in pseudo-labeling, our approach provides an effective fine-tuning of the model and a higher predictive capability on the unseen classes. By periodically updating the pseudo-label generator with the one fine-tuned through this process, we progressively improve the signal robustness for the unseen classes.

Our main contributions are: (a) we devise a self-training pipeline to obtain strong supervision for unseen classes from unlabelled pixels in GZLSS. The key component is a pseudo-label generator that enforces consistency constraints on data augmentations ; (b) we show that a model finetuned through such process progressively enhances its ability in predicting unseen classes, and consequently the quality of pseudo-labels. (c) we extensively analyze our approach on PascalVOC12, both excluding and including the background, and COCO-stuff datasets, demonstrating that our model stands as the state-of-the-art in GZLSS.

2. Related Works

Zero-Shot Learning. The ZSL models can be divided in four main categories [45] according to how they transfer knowledge from seen to unseen categories. The first provides a two-stage approach to obtain posterior class probabilities from intermediate attributes extracted from images through additional classifiers [21]. The second tackles the task as a visual-semantic embedding problem evaluating the compatibility between the visual space and the semantic one, so that proximity translates in a semantic relationship [43, 2, 34, 1, 34, 48]. The third category uses a class-level semantic conditioned generator to feed additional synthetic CNN features for unseen classes during the training of a discriminative classifier [46, 6]. The last category's models address the task in a completely generative way, by modeling the class-conditional distributions to capture semantic relationships among seen and unseen classes [4, 41, 24, 31]. In this work, we refer to the second and third ones as they inspired the existing methods for GZLSS.

Generalized Zero Label Semantic Segmentation. To the best of our knowledge, only SPNet [44], ZS3 [7], and CaGNet [16] directly address GZLSS. SPNet follows the second category's approach [34]: a segmentation model is entrusted to extract the visual features that are then projected in the semantic space by a matrix multiplication with a word embedding representation; ZS3 and CaGNet extend the features-generative approach used by [6] in classification: the former uses a Graph Convolutional Network to embed a contextual prior on categories disposition (*"mouse is commonly close to the keyboard"*, ...); the latter does the same but at pixel-wise level, feeding the generator with a contextual latent code instead of the random noise. ZS3 and CagNet don't directly address the GZLSS scenario, but they propose a variant of their models to do it through self-training, respectively indicated as *ZS5* and *CaGNet + ST*. In this work, we rely on SPNet as we want to demonstrate how an approach as simple as ours can enhance the prediction capability of a segmentation model.

Self-training in semantic segmentation. Pseudo-labeling has been widely used as a self-supervision strategy in poorly annotated computer vision scenarios [22, 3, 35, 39, 18]. In image classification, often a pre-trained reference model generates pseudo-labels for unlabelled pixels by embedding a fixed target distribution $q*$ during training [22, 3] or one continuously adapting to student p_θ's learning state [35]. [18] infers pseudo-labels in a transductive setting through label propagation on a nearest neighbor graph built with the features extracted by the model for labeled and unlabeled data. In semi-supervised semantic segmentation, many works rely on *consistency training*: PseudoSeg [50] generates pseudo-labels for unlabelled pixels by wisely fusing different sources of predictions, decoder and Grad-CAM, and then it imposes the consistency of the predictions of multiple augmented images with such pseudo-labels. [29] proposes to use adversarial training of a segmentation model that figures as a generator to strengthen the predictions for unlabeled data and to use the discriminator both to identify as good/fake predictions and as a quality measure to select most confident predictions. [33] forces an invariance of the predictions over different encoder's outputs perturbations. [12] shows that iteratively applying pseudo-labeling enhances Scene Segmentation in Urban Video Sequences. In ZSL, [47] subordinates the pseudo-labels selection to the model's confidence, in a transductive ZSL scenario. In GZLSS, [26, 3] enhances the quality of hard pseudo-labeling by first training its model according to an unbiased loss in a transductive way. Instead, ZS5 and CaGNet filter out the $p\%$ less confident labels self-produced in a GZS setting for the unlabelled pixels. We aim to self produce unseen labels for the unlabelled pixels as well, but without introducing any sensitive hypeparameters to improve robustness and generating ZS pseudo-labels.

3. Self-training with Consistency Constraints

GZLSS is particularly challenging because of the severe class-imbalanced issue, leading to a significant performance drop on unseen classes. SPNet [44] fixes this issue by reducing the prediction scores of seen classes by a constant factor γ. However, γ is hard to tune due to its sensibility i.e., a small perturbation of γ may lead to a significant change in performance. We argue that the unlabeled pixels ignored during training contain useful information about the unseen classes and incorporating those pixels into training would alleviate the class-imbalanced issue. Thus, we propose a self-training framework that leverages those unlabeled pixels by generating pseudo-labels for them.

3.1. Background: semantic projection network

First, we formally define the task and then describe the semantic projection network (SPNet) [44].

GZLSS Task formulation. Let $\mathcal{S} = \{1, \ldots, C^s\}$ and $\mathcal{U} = \{C^s + 1, \ldots, C^s + C^u\}$ denote two disjoint label sets of seen and unseen classes respectively. $\mathcal{T} = \{(x, y) | x \in \mathcal{X}, y_{mn} \in \{\{0\}, \mathcal{S}\}\}$ is the training set where x is an image of spatial size $N \times M$ in the RGB image space $\mathcal{X}$, y is its corresponding label mask with the same size, and y_{mn} is its corresponding class label at pixel (m, n) belonging to one of the seen classes S or the unlabeled unseen class denoted as 0. Moreover, each class label is represented by the word embedding (e.g., word2vec [28]) associated to its class name. We denote the word embedding matrices of seen and unseen classes with $W^s \in \mathbb{R}^{D \times |C^s|}$ and $W^u \in \mathbb{R}^{D \times |C^u|}$ with D being the dimension of the word embedding space. Given $\mathcal{T}$, W^s and W^u, the task of generalized zero-label semantic segmentation (GZLSS) is to learn a model that is capable to make pixel-wise predictions among both seen and unseen classes at test time.

Semantic projection network. SPNet [44] consists of a visual-semantic embedding module and a semantic projection layer. The former (denoted as ϕ) is based on a CNN backbone (e.g., DeepLab [10]), mapping an input image x to D feature maps of size $N \times M$ i.e., $\phi(x) \in \mathbb{R}^{D \times N \times M}$. This can be interpreted as mapping each pixel at (n, m) to a D-dimensional feature embedding $\phi(x)_{nm}$ in the semantic embedding space where knowledge transfer can be facilitated via word embeddings. The latter computes the inner product between the pixel embedding and word embeddings followed by the softmax that outputs the posterior probability over training classes,

$$P(\hat{y}_{nm} = c | x; W^s) = \frac{\exp(w_c^T \phi(x)_{nm})}{\sum_{c' \in \mathcal{S}} \exp(w_{c'}^T \phi(x)_{nm})} \qquad (1)$$

Algorithm 1: STRICT pseudo-code

```
$P_t \leftarrow$ ZLSS model at iteration $t$;
$P_{t-1} \leftarrow$ ZLSS model at previous iteration (t-1);
$\{A_1(\cdot), ..., A_k(\cdot)\} \leftarrow$ data augmentations;
$\mathcal{T} \leftarrow$ train set;
$T \leftarrow$ number of iterations;
for $t = 1, 2, ..., T$ do
    foreach $(x, y)$ in $\mathcal{T}$ do
        $\hat{y} \leftarrow$ model prediction$P_{t-1}(x)$;
        $A \leftarrow$ augmentations $\{A_1(x), ..., A_k(x)\}$;
        $\Gamma \leftarrow$ hard pseudo labeled masks$\{\bar{y}^k, ...\bar{y}^K\}$;
        $\bar{y} \leftarrow A_1^{-1}(\bar{y}^k) \cap \ldots \cap A_K^{-1}(\bar{y}^K)$;
        $\mathcal{L} \leftarrow \mathcal{L}_{CE}(x, y) + \lambda \mathcal{L}_{CE}(x, \bar{y})$;
        $P_t \leftarrow$ SGD model update;
    end foreach
    $P_{t-1} \leftarrow P_t$
end for
```

where $w_c \in \mathbb{R}^D$ is the c-th row of the matrix W^s, corresponding to the word embedding of class c. For a particular training example (x, y), the standard cross-entropy loss is,

$$\mathcal{L}_{CE} = \sum_{n,m=1}^{N,M} -\mathbb{1}[y_{nm} \neq 0] \log P(\hat{y}_{nm} = y_{nm} | x) \qquad (2)$$

where y_{nm} denotes the true class label at pixel (n, m) and $\mathbb{1}[y_{nm} \neq 0]$ is an indicator function that is 1 if $y_{nm} \neq 0$ otherwise 0. Note that the image x might include pixels from unseen classes, but those pixels are not labeled (i.e., $y_{nm} = 0$) and their losses are ignored for ZLSS. The network can be trained in an end-to-end manner by optimizing the above loss on the whole training set $\mathcal{T}$ of seen classes.

Inference. At test time, we predict all classes by searching for the class that yields the highest probability using the word embeddings of seen and unseen classes,

$$\arg\max_{c \in \mathcal{S} \cup \mathcal{U}} P(\hat{y}_{nm} = c | x; [W^s, W^u]) \qquad (3)$$

3.2. Iterative self-training pipeline

Figure 2 shows an overview of our self-training pipeline which consists of two major steps: (1) train the SPNet, and predict pseudo-labels for unlabeled pixels and (2) feed the pseudo-labels back to the training set and retrain the SPNet. The last two steps are iterative, which means that the refined model will be used to generate more accurate pseudo-labels for retraining the model. Algorithm 1 describes our iterative training pipeline in details.

Pseudo-label generation. Given the original training set $\mathcal{T}$ and word embeddings W^s from seen classes, we train the SPNet [44] by optimizing $\mathcal{L}_{CE}$ defined in Equation 2. Note that the training set $\mathcal{T}$ contains labeled pixels of seen

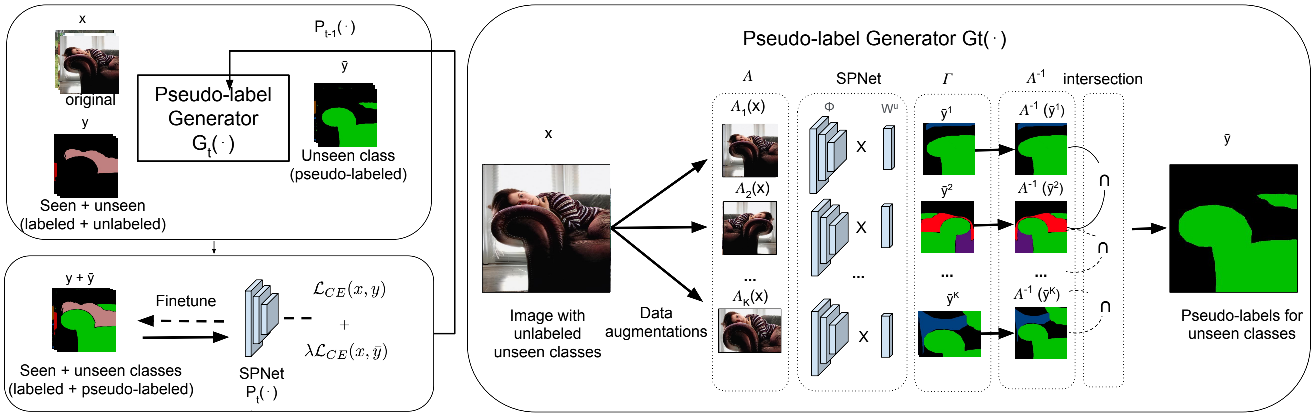


Figure 2: An overview of our STRICT model: at the iteration t, the pseudo-label generator G_t produces a pseudo labelled mask $\bar{y}^k$ for the unlabelled pixels of each of the K image' augmentations $\{A_1(x), \ldots, A_k(x)\}$. We then obtain the final pseudo-label mask $\bar{y}$ by applying the intersection operation on them. The model P_t is fine-tuned with the pixel-wise cross-entropy loss computed both on labeled (y) and pseudo-labeled ($\bar{y}$) pixels. At the iteration $(t+1)$, P_t will be used for the pseudo-label generator.

classes and unlabeled pixels of unseen classes. The unlabeled pixels are ignored when computing $\mathcal{L}_{SP}$ [44]. In other words, unseen classes are actually observed by the network but do not contribute to the training loss, which is a major difference from ZSL. We denote the model learned from this step with $P_0(\hat{y}|x;W^u)$ (see Equation 1) that outputs the probability distribution over unseen classes. Without loss of generality, we assume that the current iteration is t; in this step, the learned model P_{t-1} from the previous iteration $t-1$ is applied to unlabeled pixels to generate pseudo-labeled examples of unseen classes to finetune the model. Here, the unlabeled pixels refer to the pixels in the original training set $\mathcal{T}$ with $y_{nm} = 0$. However, pseudo-labels produced by simply making predictions with P_t contain a large amount of label noise which may hurt the model training. To this end, we propose a pseudo-label generator (denoted G) with consistency constraints to filter out potentially wrong pseudo-labels. It is worth noting that the pseudo-label generator makes predictions among only unseen classes (i.e., zero-label semantic segmentation setting) because it is known that the unlabeled pixels in the training set all belong to unseen classes. Specifically, for each of the training images containing unlabeled pixels, our pseudo-label generator computes $\bar{y} = G(x)$ where $\bar{y} \in \{\{0\}, U\}^{N\times M}$ denotes the pseudo-label mask of unseen classes for the image x and $\bar{y}_{nm} = 0$ if the pixel (n,m) belongs to a seen class. The technique details of our pseudo-label generator will be discussed in Section 3.3.

Iterative self-training. After the pseudo-label generation step, for each training image x, we will have a real label mask $y \in \{\{0\}, S\}^{N\times M}$ of seen classes and a pseudo-label mask $\bar{y} \in \{\{0\}, U\}^{N\times M}$ of unseen classes. y and $\bar{y}$ are then used to compute the labeled and pseudo-labeled loss terms separately. Formally, we optimize the following combined loss to finetune the model P_{t-1},

$$\mathcal{L} = \mathcal{L}_{CE}(x,y) + \lambda\mathcal{L}_{CE}(x,\bar{y}) \tag{4}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss defined in Equation. 2 and λ is a hyperparameter. The first loss term is the same with the original SPNet loss while the second term is computed on unlabeled pixels with the generated pseudo-labels being the supervision. The main insight is that the pseudo-labeled pixels circumvent the need of unseen class data, yielding a more balanced training set that facilitates the GZLSS. Intuitively, the better the model is, the more accurate pseudo-labels it can generate. Therefore, we can apply the finetuned model P_t from current iteration t back to generate pseudo-labels for the next iteration $t+1$. The finetuning and pseudo-label generation steps run iteratively according to the procedure described in Section 4.

3.3. Consistency constraints

Generating hard pseudo-labels (one-hot prediction) directly from the model P_t is not ideal because the noise level of one single prediction is high (as shown in Figure 1). CagNet [16] and ZS5 [7] address this issue by filtering out the pixels for which no unseen classes are present among the top p% softmax activations. Therefore, even if they ignore the predictions seen for the unlabelled pixels they obtain the softmax activations for them in the GZS scenario, hence including seen classes to the search space of unlabelled pixels that for sure are background or unseen. Moreover, their strength relies on the hyperparameter p, to be finetuned according to the confidence of the pseudo-label generator and to the dataset. We propose a simple approach to reduce the noise of pseudo-labels based on the consistency regularization [33, 40, 36, 30, 20]. The key assumption is that a pseudo-label is more likely to be correct if the model makes the same prediction from multiple augmented variants of the image. More formally, given an image x containing unlabeled pixels, we apply K different data augmentations (denoted as $A_k(\cdot)$) to obtain a set of K augmented images i.e., $\{A_1(x), \dots, A_K(x)\}$. Here we denote A_1 as an identity mapping and mainly consider the horizontal mirroring and scaling with different scaling factors as our data augmentation scheme since we find they work best in the experiments. We then apply the model P_t to generate hard pseudo-labels (one-hot labels) for every unlabeled pixels in each augmented image,

$$\bar{y}_{nm}^{k} = \arg\max_{c \in \mathcal{U}} P(\hat{y}_{nm} = c | A_k(x); W^u) \tag{5}$$
$$\forall k \in \{1, \dots, K\}, \forall (n,m) \in \mathcal{I}.$$

This yields a set of K hard pseudo-labeled masks $\{\bar{y}^k, \dots, \bar{y}^K\}$ for the image x. Intuitively, those data augmentations only transform the image spatially and the semantic of each pixel should remain the same. We then obtain the final pseudo-label mask by applying the intersection operation on those K masks,

$$\bar{y} = A_1^{-1}(\bar{y}^k) \cap \ldots \cap A_K^{-1}(\bar{y}^K), \tag{6}$$

where A_k^{-1} denotes the inverse data augmentation that transforms augmented masks back to the original coordinates. The intersection operation essentially filters out the pseudo-labels that are inconsistent across multiple augmented masks, yielding more accurate pseudo-labels. Although a similar consistency regularization has been explored in semi-supervised learning [5], we are the first to apply the consistency constraints for the GZLSS task.

4. Experiments

Datasets and metrics. We evaluate our approach on two datasets, PascalVOC12 [15] and COCO-stuff [8], following previous works [44, 16] for the data splits and the validation procedure. PascalVOC12 is an object segmentation benchmark, containing images of 20 foreground objects plus the background class. COCO-stuff is a large-scale dataset for scene segmentation, with 164K images containing 80 common objects and 91 stuff classes. Our train/val/test sets are mutually exclusive classes i.e., 11185/500/1449 images from 12/3/5 classes on VOC12 and 116287/2000/5000 images from 155/12/15 classes on COCO-stuff. As train and val sets belong to disjoint subsets of seen classes, we use the following two stage procedure for fine tuning: (i) we first select the best hyperparameters considering as seen the train classes and as unseen the validation ones; (ii) we perform training considering as seen both train and validation classes with *fixed* hyperparameters (i.e. without looking at the validation set again). For PascalVOC12, we perform additional experiments where the background is included among the set of seen classes. Following [44], we measure the generalized zero-label performance in terms of mean Intersection over Union (mIoU) on the seen (S) and unseen (U) classes, as well as the harmonic mean (HM) among the two.

Method	**PascalVOC12**			**COCO-stuff**		
	S	U	HM	S	U	HM
SPNet [44]	73.3	15.0	21.8	20.5	14.3	16.8
ZS3 [7]	77.3	17.7	28.7	34.7	9.5	15.0
CaGNet [16]	78.4	25.6	39.7	35.5	12.2	18.2
SPNet+ST	77.8	25.8	38.8	34.6	26.9	30.3
ZS5 [7]	78.0	21.2	33.3	34.9	10.6	16.2
CaGNet + ST [16]	78.6	30.3	43.7	**35.6**	13.4	19.5
STRICT	**82.7**	**35.6**	**49.8**	35.3	**30.3**	**32.6**

Table 1: Comparing with the state of the art on PascalVOC12 and COCO-stuff.

Baselines and implementation details. We compare our approach with three state-of-the-art GZLSS methods, namely the baseline SPNet [44], and two generative approaches, ZS3 [7] and CaGNet [16]. Additionally, we include the self-training variants of CaGNet (CaGNet+ST) and ZS3 (ZS5), both using the top percentage of the pixels assigned to unseen classes as pseudo-labels. Moreover, we report the results of another baseline, the calibrated SPNet trained by performing hard pseudo-labelling on unlabeled pixels, without any consistency strategy (SPNet+ST). For fair comparison with previous works, we use DeepLabV2 [10] as the segmentation model with an Imagenet [14] pretrained ResNet-101 [17] as backbone. We train our network with SGD, with a momentum of 0.9 and a weight decay of $5 \cdot 10^{-4}$. The learning rate is initially set to $2.5 \cdot 10^{-4}$ with a polynomial decay, as in [10]. After training the base model train the model for 2K iterations on PascalVOC12 and for 22K iterations for COCO-stuff, using a batch-size of 8 images. After training the network for 20K iterations for VOC and 100K on COCO with only supervision on seen class pixels, we keep the same hyperparameters and we fine-tune the network with our self-training strategy, considering one cycle of self-training finished after 2K iterations on PascalVOC12 and after 22K iterations for COCO-stuff. Results for SPNet and SPNet+ST are reported after running the approach under our framework.

4.1. Comparison with the state of the art

We compare our approach with the state of the art on both PascalVOC12 and COCO-stuff, reporting the results in Table 1. A first outcome of the experiments is that self-training strategies improve the performance of all methods and for all metrics. In particular, in PascalVOC12, ZS5 improves his not self-trained counterpart by almost 5% in HM and CaGNet+ST improves CaGNet by almost 4% on the same metric. Remarkably, SPNet+ST improves the base SPNet by 17% in HM, with a 11% improvement on the unseen classes and a 4.5% on the seen ones, on PascalVOC12. Similar observations hold for COCO-stuff, where the improvements on the HM are of 1.5% for ZS5 over ZS3, 1.3% of CaGNet over CaGNet and 13.5% of SPNet+ST over SPNet. Note that, with this simple strategy, SPNet surpasses all more complex generative approaches on COCO-stuff for unseen mIoU and HM, while achieving a lower harmonic mean than CaGNet on PascalVOC12. These results confirm that considering the co-occurrence of seen and unseen classes through self-training is always very beneficial. Moreover, the improvements are larger in non-generative methods (SPNet) than in generative ones. A second, clear outcome, is that our STRICT strategy outperforms every published results by a good margin. On PascalVOC12, it surpasses of 6.1% on harmonic mean and of 5.3% on unseen mIoU the previous state of the art (CaGNet+ST). The margin is even higher in the large scale COCO-stuff dataset, with our approach surpassing CaGNet of 16.9% on unseen class mIoU and of almost 13.1% on harmonic mean. If we compare STRICT with the SPNet+ST baseline, we see that the improvement is higher on PascalVOC12 (4.9% on seen classes, 9.8% on unseen classes and 11% on harmonic mean) while being less marked on COCO-stuff, with a 0.7% improvement on seen class, 3.4% improvement on unseen class mIoU and 2.3% on the harmonic mean. These improvements are outstanding, confirming the importance of equipping any zero-label semantic segmentation model with an effective self-training strategy. Note that the self-training approach greatly reduces the bias of the network on seen classes. Indeed, differently form SPNet, we do not need a calibration term to balance seen and unseen class predictions. Similarly, different from generative approaches, we do not rely on synthesised pixels, but rather exploit the more precise information coming from the unlabeled pixels of our images.

Method	PascalVOC12		
	S	U	HM
SPNet [44]	54.7	2.5	4.7
ZS3 [7]	59.0	4.0	7.5
SPNet+ST	72.7	4.0	7.6
ZS5	66.1	1.7	3.7
STRICT	**74.7**	**14.3**	**24.0**

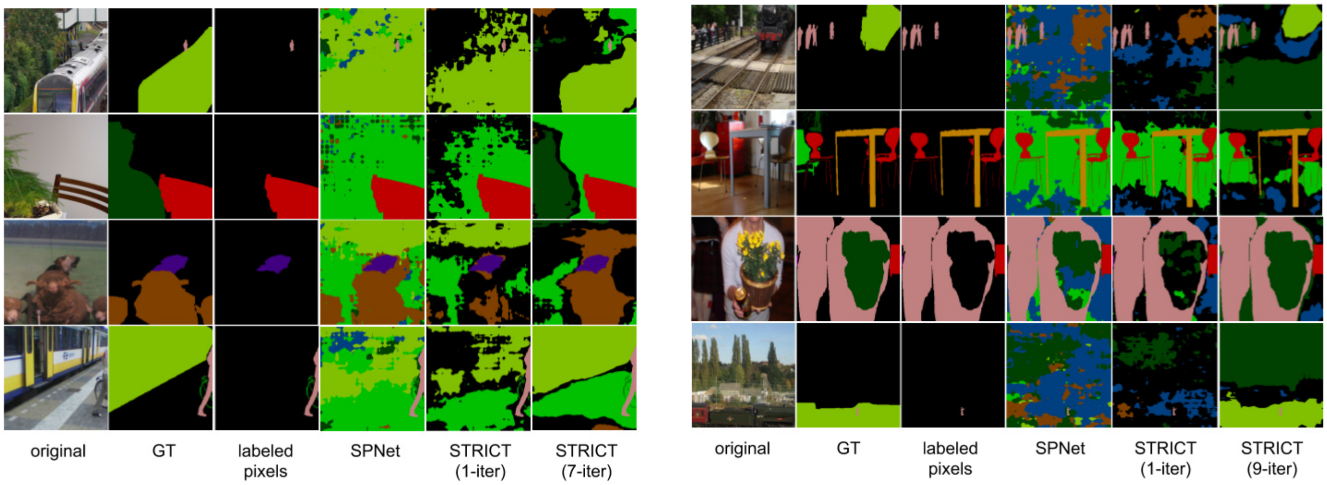
Table 2: PascalVOC12 results with background class included among the seen set.

Mirroring	Scaling	S	U	HM
		77.8	25.8	38.8
✓		80.4	27.2	40.7
	down	82.1	27.8	41.5
	up	82.0	31.1	45.1
	random	81.6	29.4	43.2
✓	down	**83.7**	29.2	43.3
✓	up	82.5	**32.9**	**47.0**
✓	random	83.2	31.4	45.6

Table 3: Ablation of different transformations for the consistency constraint of STRICT on PascalVOC12.

Impact of the background on PascalVOC12. Standard GZLSS approaches for object segmentation usually do not consider the distinction between foreground and background, performing evaluation only on pixels of foreground objects. Here we evaluate the change in performance when the background class is included in the search space. Note that this scenario is far more challenging since the pixels of unseen classes might be labeled as background, hampering the model capability to discriminate them.

Results are reported in Table 2, for our method, SPNet, ZS3 and their self-trained variants. All methods, both STRICT and the baselines, suffer a severe performance degradation when including the background in the classifier. Indeed, if we compare Table 1 with Table 2, we can see how SPNet achieves only 2.5% of mIoU on unseen classes (almost 12% lower than Table 1) with an overall 4.7% on harmonic mean (17% lower). With self-training, results improve only slightly, with SPNet+ST obtaining 4% mIoU on unseen classes and a 7.6% of harmonic mean. Surprisingly, ZS3 outperforms its self-trained counterpart ZS5 in this setting as learning a robust classifier for unseen classes in a generative fashion is difficult in segmentation, due to the high complexity of the images. Additionally, both the generation and the pseudo-labeling process are hampered by the bias of the network toward predicting background in place of unseen class pixels. Our STRICT approach is effective even in this setting, with an mIoU on unseen classes of 14.3% and an overall harmonic mean of 24%. We highlight how these results are on par with the performance of the calibrated SPNet on the standard scenario where the background is ignored, being slightly lower (3% on har-



(a) Pseudo-labels generated with STRICT for PascalVOC12 unseen classes when background is ignored.

(b) Pseudo-labels generated with STRICT for PascalVOC12 unseen classes when background is included.

Figure 3: Qualitative pseudo-labeling results of STRICT on PascalVOC12 without (left) and with (right) background as seen class. Train GT refers to labels for the unseen classes.

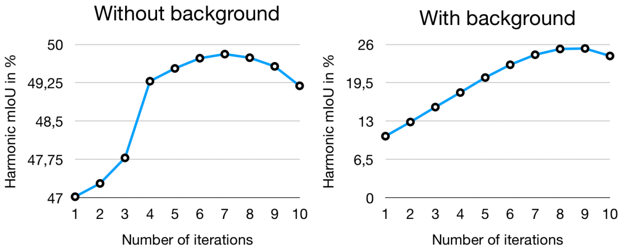


Figure 4: STRICT mIoU along with the number of iterative fine-tuning steps i.

monic mean and unseen class mIoU) than ZS3. Despite these promising results, the gap in performance among our model on the two scenarios is still large (25% on harmonic mean and 21% on unseen mIoU). This means that the technical challenges of GZLSS for object segmentation when background is included require additional technical components, explicitly addressing problems such as the semantic shift of the background class [9].

4.2. Ablation study

In this section, we study the effect of different components of our approach, namely the type of transformations applied for the consistency procedure and the number of self-training iterations on PascalVOC12.

Different image transformations. We first evaluate which image transformations are more effective for applying our self-training with consistency constraints. In particular, we consider simple and invertible image-level transformation such as three variants of multi-scaling (down, up and random scaling) and mirroring. We report the results of our analysis in Table 3. As the table shows, performing multi-scaling is, in general, more effective than applying only mirroring. Among the scaling alternatives, upscaling brings the best results, with the highest mIoU on unseen classes (31.1%) and harmonic mean (45.1%). Combining mirroring and upscaling, we obtain the best performance, with 32.9% of mIoU on unseen classes and 47% of harmonic mean.

Number of self-training iterations. An important aspect of our algorithm is the iterative self-training procedure, with the pseudo-labeling model updated after each iteration. Here, we analyze the impact of the number of self-training iterations for STRICT in Figure 4, where we report the results as mIoU on unseen classes and harmonic mean and in both cases with and without background included. As the Figure shows, for both metrics and settings performances tend to increase as the number of self-training iterations does. In particular, performances rapidly increase until six self-training iterations, after which they saturate and/or slightly decrease. This decrease can be caused by the fact that we do not have ground-truth for the unseen class pixels and noisy predictions can be reduced but not entirely eliminated by our consistency constraint.

4.3. Qualitative analysis

In this section, we report qualitative analysis on PascalVOC12 regarding i) the pseudo-labels generated by our model and ii) semantic segmentation results.

Pseudo-labels. Another crucial point of our algorithm is generating good pseudo-labels as supervision signal for our

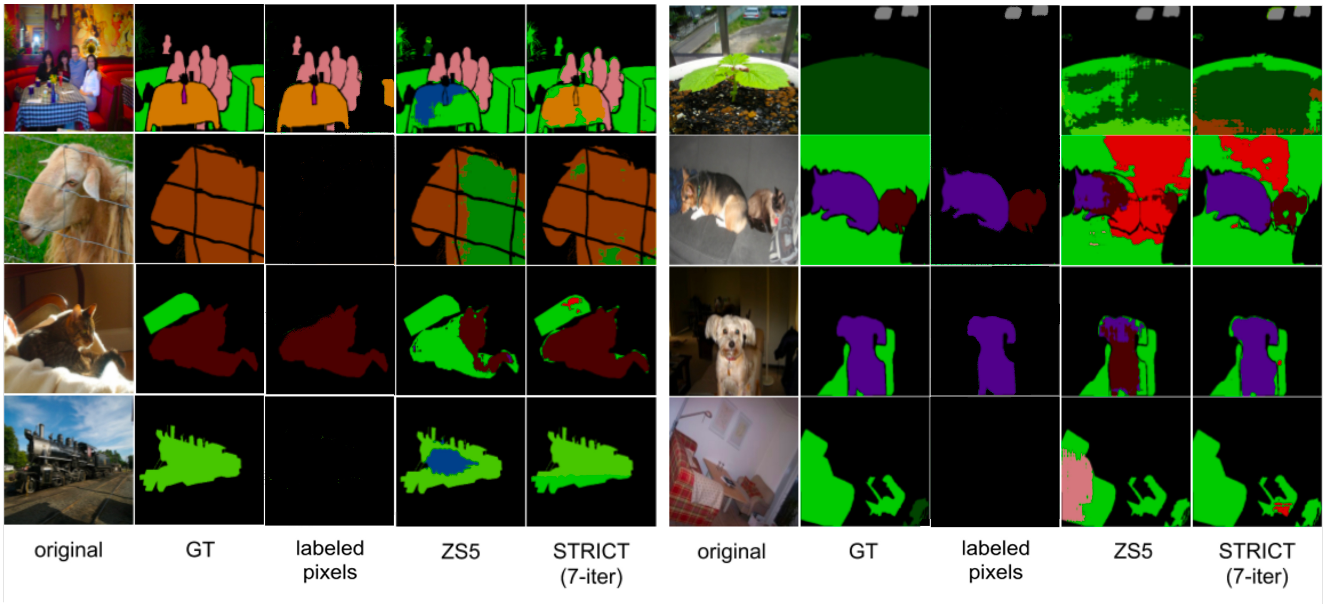


Figure 5: Qualitative results of STRICT with the original SPNet and ZS5 on PascalVOC12 when the background is ignored.

model on unseen class pixels. Figure 3a and 3b show some annotations on unseen classes obtained by our model, when the background is ignored and included during training respectively. For each original image, *GT* is the actual ground truth y, while *labeled pixels* represent the annotation for seen classes y^s that the model sees before the pseudo-labeling. From the Figures we see that, while our starting point (SPNet+ST) detects the presence of pixels of unseen classes, the predictions are noisy, with some pixels assigned to classes not present in the current image. Our consistency constraint (STRICT) allows to largely reduce the noise, eliminating most of the pseudo-labels assigned to pixels of classes not present in the current image (*e.g. train* in third row of Figure 3a, *tv/monitor* in first and fourth rows of Figure 3b). With more iterations, STRICT produces more refined pseudo-labels, where spatially coherent structure are present. This means the pseudo-label generator capture global information of unseen classes, something which is not possible to do with a single stage of pseudo-labeling.

Semantic segmentation. Finally, to compare our model with the other baselines, we show qualitative semantic segmentation results of our method and ZS5 on Figure 5. As the Figure shows, our model is able to correctly identify pixels of unseen (*e.g. sofa*) as well as seen (*e.g. person*) categories. Moreover, it achieves a good trade-off between seen and unseen classes. For instance, on the image of the second row, left, ZS5 misclassifies most of the pixels of the *sheep* (unseen class) as a *cow* (seen class), showing its bias toward seen classes. On the other hand, our model segments almost perfectly the *sheep*, with few pixels misclassified. A similar example is the *table* (unseen class) misclassified by ZS5 as *tv/monitor* while almost correctly segmented by our model. These images show also some drawbacks of our approach, meaning the dependency of the results on the number of co-occurring pixels. For instance, since *plant* occupies low portions of the images, it is hard for the network to produce consistent pseudo-labels for it, with consequently low recognition ability of the final model for that class. Future works might exploit strategies to regularize the supervision for unseen classes based on the number of pseudo-labels generated for each of them.

5. Conclusions

In this work, we proposed a self-training approach to learn the model to segment classes not annotated in the training set by leveraging on their semantic representation. Our self-training pipeline is simple, robust and highly scalable, as it relies on the ability of the model to predict consistent labeling among different augmented versions of the same image to filter the generated pseudo-labels and on the iterative strengthening of the pseudo-label generator. We demonstrated the effectiveness of this method on two commonly used benchmarks for semantic segmentation and we obtained that applying such simple considerations outperforms other more complex strategies in the GZLSS.

Acknowledgments

This work has been partially funded by the ERC 853489 - DEXIM, the ERC grant N. 637076 RoboExNovo, and by the DFG – EXC number 2064/1 – Project number 390727645.

References

[1] Zeynep Akata, Honglak Lee, and Bernt Schiele. Zero-shot learning with structured embeddings. 09 2014.

[2] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *CoRR*, abs/1503.08677, 2015.

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *CoRR*, abs/1908.02983, 2019.

[4] Sarkhan Badirli, Zeynep Akata, and Murat Dundar. Bayesian zero-shot learning. *CoRR*, abs/1907.09624, 2019.

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[6] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Generating visual representations for zero-shot classification. *CoRR*, abs/1708.06975, 2017.

[7] Maxime Bucher, Tuan-Hung Vu, Mathieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In *NeurIPS*, 2019.

[8] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *CoRR*, abs/1612.03716, 2016.

[9] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020.

[10] 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. *CoRR*, abs/1606.00915, 2016.

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[12] 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, 2020.

[13] 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.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[16] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. *Proceedings of the 28th ACM International Conference on Multimedia*, Oct 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[18] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. *CoRR*, abs/1904.04717, 2019.

[19] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Weakly supervised semantic labelling and instance segmentation. *CoRR*, abs/1603.07485, 2016.

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *CoRR*, abs/1610.02242, 2016.

[21] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[22] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[23] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation\\using stochastic inference. *CoRR*, abs/1902.10421, 2019.

[24] Yanan Li and Donghui Wang. Zero-shot learning with generative latent prototype model. *CoRR*, abs/1705.09474, 2017.

[25] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *CoRR*, abs/1604.05144, 2016.

[26] Haiyang Liu, Yichen Wang, Jiayi Zhao, Guowu Yang, and Fengmao Lv. Learning unbiased zero-shot semantic segmentation networks via transductive transfer, 2020.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 10 2013.

[29] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high- and low-level consistency. *CoRR*, abs/1908.05724, 2019.

[30] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning, 2018.

[31] Tanmoy Mukherjee and Timothy Hospedales. Gaussian visual-linguistic embedding for zero-shot recognition. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 912–918, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[32] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency

for object segmentation from image level labels. *CoRR*, abs/1701.08261, 2017.

[33] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training, 2020.

[34] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 1410–1418. Curran Associates, Inc., 2009.

[35] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le. Meta pseudo labels, 2020.

[36] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672, 2015.

[37] Olga Russakovsky, Amy L. Bearman, Vittorio Ferrari, and Fei-Fei Li. What's the point: Semantic segmentation with point supervision. *CoRR*, abs/1506.02106, 2015.

[38] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *CoRR*, abs/1709.03410, 2017.

[39] Qianru Sun, Xinzhe Li, Yaoyao Liu, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. *CoRR*, abs/1906.00562, 2019.

[40] Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780, 2017.

[41] Vinay Kumar Verma and Piyush Rai. A simple exponential family framework for zero-shot learning. *CoRR*, abs/1707.08040, 2017.

[42] Kaixin Wang, JunHao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. *CoRR*, abs/1908.06391, 2019.

[43] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh N. Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. *CoRR*, abs/1603.08895, 2016.

[44] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata. Semantic projection network for zero- and few-label semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8248–8257, 2019.

[45] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *CoRR*, abs/1707.00600, 2017.

[46] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. *CoRR*, abs/1712.00981, 2017.

[47] Lei Zhang, Peng Wang, Lingqiao Liu, Chunhua Shen, Wei Wei, Yanning Zhang, and Anton van den Hengel. Towards effective deep embedding for zero-shot learning. *CoRR*, abs/1808.10075, 2018.

[48] Ziming Zhang and Venkatesh Saligrama. Classifying unseen instances by learning class-independent similarity functions. 11 2015.

[49] 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.

[50] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation, 2020.