Semantic Projection Network for Zero- and Few-Label Semantic Segmentation

Yongqin Xian[1*] Subhabrata Choudhury[1*] Yang He[1] Bernt Schiele[1] Zeynep Akata[1,2]

[1]Max Planck Institute for Informatics
Saarland Informatics Campus

[2]Amsterdam Machine Learning Lab
University of Amsterdam

Abstract

Semantic segmentation is one of the most fundamental problems in computer vision. As pixel-level labelling in this context is particularly expensive, there have been several attempts to reduce the annotation effort, e.g. by learning from image level labels and bounding box annotations. In this paper we take this one step further and propose zero- and few-label learning for semantic segmentation as a new task and propose a benchmark on the challenging COCO-Stuff and PASCAL VOC12 datasets. In the task of zero-label semantic image segmentation no labeled sample of that class was present during training whereas in few-label semantic segmentation only a few labeled samples were present. Solving this task requires transferring the knowledge from previously seen classes to novel classes. Our proposed semantic projection network (SPNet) achieves this by incorporating class-level semantic information into any network designed for semantic segmentation, and is trained in an end-to-end manner. Our model is effective in segmenting novel classes, i.e. alleviating expensive dense annotations, but also in adapting to novel classes without forgetting its prior knowledge, i.e. generalized zero- and few-label semantic segmentation.

1. Introduction

In semantic image segmentation the aim is assign a label to every pixel in an image by partitioning it into several semantic regions and then learning the appearance of various classes as well as the background. Although deep CNN-based approaches have achieved good performance for this task, they require costly dense annotations to learn their numerous parameters. Hence, leveraging weak annotations via image-level labels [34, 32, 31] or point [5], bounding box [20], scribble-level annotations [25] recently gained interest. On the other hand, as humans, we easily learn to recognize a previously unseen, i.e. novel, class by associating it with classes that we know. However, segmenting such novel classes via modern machine learning techniques is still an open problem as this process requires knowledge transfer from known classes to previously unseen ones.

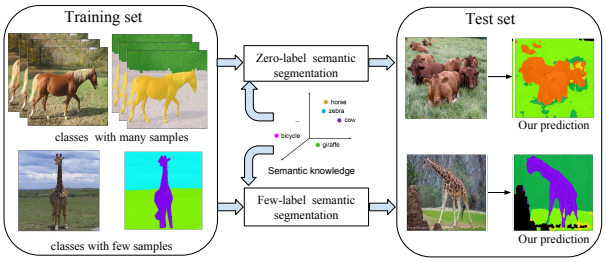


Figure 1: We propose (generalized) zero- and few-label semantic segmentation tasks, i.e. segmenting classes whose labels are not seen by the model during training or the model has a few labeled samples of those classes. To tackle these tasks, we propose a model that transfers knowledge from seen classes to unseen classes using side information, e.g. semantic word embedding trained on free text corpus.

Knowledge transfer to novel classes is not a new task. Learning to predict novel classes has been studied extensively in the context of image classification, i.e. zero-shot learning [23, 57, 7, 2]. In zero-label semantic segmentation (ZLSS), our aim is to segment previously unseen, i.e. novel, classes, in few-label semantic segmentation (FLSS) these novel classes have a small number of labeled training examples (see Figure 1). In this work, we also aim for learning without forgetting the previously seen classes, i.e. generalized ZLSS and FLSS. To achieve these aims, we propose Semantic Projection Network (SPNet) that incorporates semantic word embeddings to an arbitrary semantic segmentation network inspired by the success of zero-shot learning. Prior models that tackle few-shot semantic segmentation [42, 11] operate in the foreground-background segmentation setting. However, in our definition of FLSS the model has to predict all the classes in an image separately, which is more challenging and realistic. Our framework utilizes the similarity between different categories in a semantic segmentation network, enabling it to transfer

*Equal contributions

learned representations to other classes. Consequently, our model is able to segment scenes containing novel classes.

Our main contributions are as follows. (1) We introduce the (generalized) zero-label and few-label semantic image segmentation task in a realistic settings inspired by zero-shot learning for image classification. (2) We propose semantic projection network (SPNet), an end-to-end semantic segmentation model which maps each image pixel to a semantic word embedding space where it is projected with a fixed word embedding to class probabilities optimizing the cross-entropy loss. (3) We create a benchmark for (generalized) zero- and few-label semantic image segmentation with two challenging datasets, i.e. COCO-Stuff and PASCAL-VOC. Our analysis shows that the SPNet model achieves impressive results both quantitatively and qualitatively in (generalized) zero-label and few-label tasks. Furthermore, as a side-product, our model improves the state of the art in zero-shot image classification demonstrating that it successfully generalizes to other tasks.

2. Related work

In this section, we review prior work on zero-shot learning, semantic segmentation and their combination.

Zero- and few-shot image classification. Most advances in zero-shot image classification were achieved by visual-semantic embedding models [13, 1, 57, 50, 41, 14, 55] that learn a compatibility function between the image embedding space i.e. CNN image feature [16], and the class embedding space, i.e. class-level attributes [23]. As complementary tasks [24, 40] focuses on assigning multiple previously unseen labels to a single image, [22, 52, 53] on predicting novel actions in a video, and recently, [4] on unseen object detection. For a comprehensive overview of zero-shot learning models, we refer the reader to [51]. As for few-shot learning, [48, 15] stand out as generating data of weakly represented classes and meta learning approaches [46, 38, 44] regularize the model by sharing parameters and applying episode training strategy. In contrast to image classification where an image has only one class label, in semantic segmentation, each image has a dense label map that assigns a label for each pixel from a set of possible object classes. Given the large amount of class co-occurrence in images, it is unrealistic to build a training set that contains no pixels from target classes for semantic segmentation. Therefore, we allow models to see pixels from target classes without accessing their labels, which explains our terminology "zero-label".

Semantic segmentation with weak supervision. Modern semantic segmentation systems [27, 9, 3] are built on the encoder-decoder networks and trained with densely labeled annotations. Much efforts focus on improving semantic segmentation under fully supervised settings, e.g. adding global context information [59, 54, 26], applying graphical models as a post-processing step to refine the output [60, 9], etc. On the other hand, weakly supervised semantic segmentation, i.e. reducing the annotation effort, has recently gained momentum. As weak supervision, prior works use image-level annotation [34, 32, 31], point [5], scribble [25] and bounding box [20] annotations. Those methods propagate the supervision to larger regions by measuring objectness [5] and saliency [31], or applying graphical models [25]. Other methods refine the coarse annotated regions to more accurate ones [20, 32]. However, those models still require all the classes to be seen during training, thus cannot easily be adapted to new classes. In contrast, we focus on segmenting completely novel classes.

Semantic segmentation of novel classes. The term zero-shot semantic segmentation appears in prior works [18, 58]. The aim of [18] is to segment novel actor-action patterns during test time. While [58] proposes open-vocabulary scene parsing task that segments novel objects by performing hierarchical parsing, we leverage word embeddings to predict the exact unseen classes and address the few-label problem in a unified framework. For few-shot semantic segmentation, previous approaches [42, 37, 11, 56] follow the meta-learning setup [46, 44], which uses a support set to predict an query image. However, those approaches are restricted to output a binary mask and fail to segment an image with multiple classes. In contrast, our approach is operating in the more realistic (generalized) few-label semantic segmentation setting, i.e. pixel-level labeling of an image where labels come from both base and novel classes.

Semantic embeddings. In learning with limited labels, some form of side information is required to transfer the knowledge learned from seen classes to unseen classes. One popular form of side information is attributes [23] that, however, require costly expert annotation. Thus, there has been a large group of studies [2, 39, 36, 10] utilizing other sources such as Word2vec [29], fastText [19], or hierarchies [30] for building semantic embeddings. In this work, we utilize Word2Vec and fastText as they do not require dataset specific human annotation.

3. SPNet Model for Segmenting Novel Classes

Modern semantic segmentation models are built on fully convolutional encoder-decoder architectures [9, 27] that output intermediate feature maps and posteriors for individual classes. However, to segment novel classes these models need to be adapted to transfer knowledge from one class to the other. Such knowledge can be obtained from class-level semantic embeddings associating different classes. Hence, the main insight of our approach is to leverage semantic word embeddings, i.e. word2vec [29] or fast-text [19], to transfer knowledge learned from base classes to novel

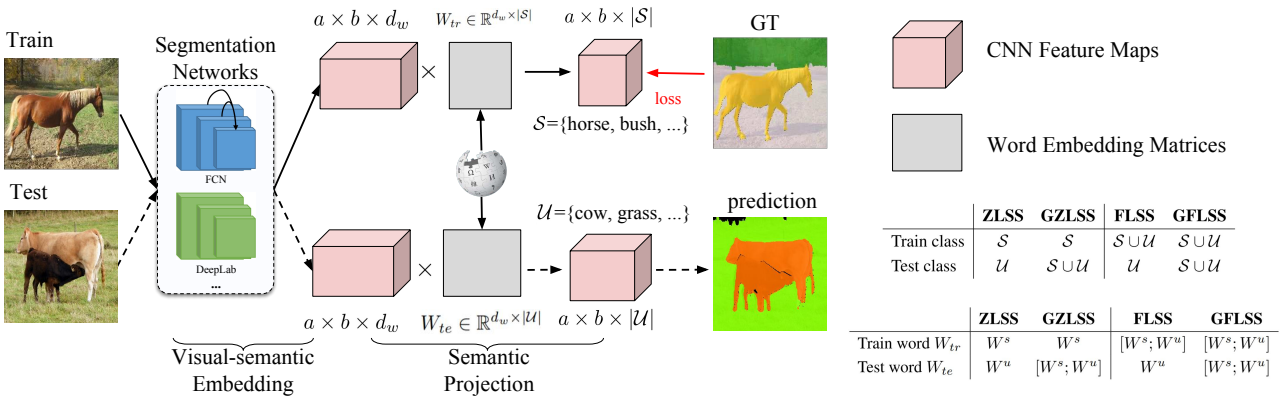


Figure 2: Our zero-label and few-label semantic segmentation model, i.e. SPNet, consists of two steps: visual semantic embedding and semantic projection. Zero-label semantic segmentation is drawn as an instance of our model. Replacing different components of SPNet, four tasks are addressed (Solid/dashed lines show the training/test procedures respectively).

classes in a two-step process. First, we propose to learn a visual-semantic embedding module that produces intermediate feature maps in the word embedding space. Second, we project those feature maps into class probabilities via a fixed word embedding projection matrix. At test time, by replacing the projection matrix with word embeddings of novel classes, our model is able to segment unseen categories. Our model is trained end-to-end and can be incorporated into any semantic segmentation network, i.e. FCN [27] and deeplab [9]. We illustrate our overall pipeline in Figure 2.

Task formulation. We denote the set of seen classes as $\mathcal{S}$ and a disjoint set of unseen classes as $\mathcal{U}$. Let $\mathcal{D}_s = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{Y}^s\}$ be our labeled training data of seen classes where x is an image in the image space $\mathcal{X}$, y is its corresponding label mask in the dense label mask space $\mathcal{Y}^s \subset \mathcal{S}^{a*b}$ of seen classes with a and b being the height and the width of the image respectively. Similarly, we define the label mask space of unseen classes as $\mathcal{Y}^u \subset \mathcal{U}^{a*b}$. In addition, $W^s \in \mathbb{R}^{d_w \times |\mathcal{S}|}$ and $W^u \in \mathbb{R}^{d_w \times |\mathcal{U}|}$ denote the word embedding matrices of seen and unseen classes where d_w is the word embedding dimension. Given $\mathcal{D}_s$, W^s, and W^u, the task of zero-label semantic segmentation (ZLSS) is to learn a model that takes an image as an input and predicts the label of each pixel among unseen classes. A more realistic setting is generalized zero-label semantic segmentation (GZLSS) where the learned model predicts both seen and unseen classes. As for the (generalized) few-label semantic segmentation task, a few labeled samples from unseen classes $\mathcal{D}_u = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{Y}^u\}$ are provided to the model during training. The test time target classes include only seen classes in few-label semantic segmentation (FLSS) whereas they include both seen and unseen classes in generalized few-label semantic segmentation (GFLSS). Here, we refer to the classes with a few labeled samples as unseen or novel, interchangeably. We summarize train class, test class and word embeddings used in different settings in Figure 2.

3.1. Semantic Projection Network (SPNet)

We address all four tasks with an unified model SPNet, which consists of two parts: visual-semantic embedding module and semantic projection layer.

i. Visual-semantic embedding module. This module is parameterized by a CNN and maps an input image $x \in \mathcal{X}$ into d_w feature maps via $\phi : \mathcal{X} \rightarrow \mathcal{R}^{a \times b \times d_w}$ of size $a \times b$. This is equivalent to embedding each pixel at (i, j) into a d_w dimensional class embedding vector $\phi(x)_{ij}$ that lies in the semantic embedding space shared by all the classes. The semantic embedding space constrains the output of the visual-semantic embedding extractor ϕ and transfers knowledge from seen to unseen classes. Note that this is different from a standard CNN where pixels are mapped into an unconstrained feature space.

ii. Semantic projection layer. The semantic projection layer maps the feature embedding $\phi(x)_{ij}$ into unnormalized logit scores followed by a softmax activation that outputs the probability distribution over each training category,

$$p(\hat{y}_{ij} = s|x; W^s) = \frac{\exp\left(w_s^\top \phi(x)_{ij}\right)}{\sum_{c \in \mathcal{S}} \exp\left(w_c^\top \phi(x)_{ij}\right)} \quad (1)$$

where $\hat{y}_{ij}$ represents the prediction for pixel (i, j), w_c is the c-th column of W^s normalized to have unit length.

In contrast to standard CNNs that predict the class posterior by adding 1×1 convolution layer or fully connected layer with learnable weights, our classifier weights W^s are predefined by a word embedding model, e.g. word2vec [29], and then fixed during training. The W^s and

the semantic projection layer estimate the compatibility between class prototypes and a feature embedding in terms of inner product similarity. Our proposed semantic projection layer is easy to implement by computing the tensor product between feature maps $\phi(x)$ and word embedding matrix W^s followed by the softmax activation function. After this layer, we directly optimize the standard cross-entropy loss over the spatial dimensions $(i, j) \in \mathcal{I}$,

$$\sum_{(i,j)\in\mathcal{I}} -\log p(\hat{y}_{ij} = y_{ij}|x) \tag{2}$$

which can be viewed as maximizing the negative log likelihood of predicting each pixel as its true label y_{ij}. Since there are no learnable parameters at the semantic projection layer, the optimization is over parameters of the visual-semantic embedding extractor ϕ. Compared to the standard semantic segmentation network, we have made subtle yet critical changes, i.e. mapping pixels to the semantic word embedding space followed by stacking a projection layer.

Inference. At the test time, in ZLSS and FLSS, we predict unseen classes by replacing the word embedding matrix in Eq. (1) with W^u. Each pixel label is predicted by:

$$\underset{u\in\mathcal{U}}{\operatorname{argmax}}\, p(\hat{y}_{ij} = u|x; W^u). \tag{3}$$

On the other hand, for GZLSS and GFLSS, we predict both seen and unseen class labels via their word embedding:

$$\underset{u\in\mathcal{S}\cup\mathcal{U}}{\operatorname{argmax}}\, p(\hat{y}_{ij} = u|x; [W^s; W^u]). \tag{4}$$

The extreme case of the imbalanced data problem occurs when there is no labeled training images of unseen classes, and this results in predictions being biased to seen classes. To fix this issue, we follow [8] and calibrate the prediction by reducing the scores of seen classes, which leads to:

$$\underset{u\in\mathcal{S}\cup\mathcal{U}}{\operatorname{argmax}}\, p(\hat{y}_{ij} = u|x; [W^s; W^u]) - \gamma\mathbb{I}[u \in \mathcal{S}] \tag{5}$$

where $\mathbb{I} = 1$ if u is a seen class and 0 otherwise, $\gamma \in [0, 1]$ is the calibration factor tuned on a held-out validation set.

Theoretically, the semantic projection layer allows our model to predict any class by simply copying its word embedding to the classifier weights. However, intuitively, the model can only perform well on the classes that share visual similarities with training classes. Hence, the word embedding ought to capture the similarity between classes.

Two-stage training in few-label setting. In our FLSS and GFLSS, we train a model with both D_s that includes a large number of samples per seen class and D_u that has only a few samples per unseen, i.e. novel, class. This is a typical imbalanced learning problem. The naive idea is to learn using both seen and unseen class samples within a mini-batch sampled uniformly from the whole training data. As expected, this leads to good performance on seen classes but inferior performance on unseen classes. Another strategy is to oversample unseen classes by first uniformly sampling a mini-batch of classes and selecting one sample from each of those classes. We found that this strategy remedies the imbalance issues to some extent but the results still remain unsatisfactory. On the other hand, fine-tuning the learned classifier on unseen class samples, i.e. after the initial optimization with only seen class samples, yields better results on unseen classes in FLSS as well as better overall results in GFLSS. Hence, we report our results in this setting.

3.2. Baseline: Hinge Visual-Semantic Loss (HVSL)

The choice of the loss function turns out to be important in zero-label semantic segmentation. Hence, in this section, we develop a baseline that shares the same embedding extractor ϕ as our SPNet but adopts the hinge visual-semantic loss instead of cross-entropy loss. Indeed hinge visual-semantic loss constitutes the most widely used loss function for zero-shot image classification [1, 4, 13, 57, 50]. In the context of semantic segmentation, we define the following hinge ranking loss for a single training example (x, y) as,

$$\sum_{(i,j)\in\mathcal{I}}\sum_{s\in\mathcal{S}}[\Delta(s, y_{ij}) + w_s^\top\phi(x)_{ij} - w_{y_{ij}}^\top\phi(x)_{ij}]_+ \tag{6}$$

where $\Delta(s, y_{ij}) = 1$ if $s \neq y_{ij}$ otherwise 0, $\phi(x)_{ij}$ is the visual-semantic embedding for pixel (i, j) in image x, y_{ij} is its corresponding ground-truth label. In practice, we follow [13] to truncate the sum by randomly sampling one class that is not ground-truth.

4. Experiments

In this section, we present both quantitative and qualitative results of zero-label semantic segmentation and few-label semantic segmentation.

Datasets. We evaluate our model on the challenging COCO-stuff [6] and PASCAL-VOC 2012 [12] datasets. COCO-stuff has 164K images with dense pixel-level annotations from 172 classes including 80 thing classes, 91 stuff classes. PASCAL-VOC is a smaller dataset which contains $13K$ images from 20 classes.

Word embeddings. Encoding the semantic similarity between labels plays an important role in bridging the gap between seen and unseen class predictions. In this work, we study two different word embedding models, i.e. word2vec [29] trained on Google News [47] and fastText [19] trained on Common Crawl [28]. The word embeddings of classes that contain multiple words are obtained by averaging the embeddings of each individual word.

	# classes		# images	
	train+val	test	train+val	test
COCO-Stuff	155+12	15	116287+2000	5000
PASCAL-VOC	12+3	5	11185 + 500	1449

Table 1: Statistics of data splits for COCO-Stuff and PASCAL-VOC datasets in terms of the number of classes and the number of images in the training and test splits.

	fastText (ft)	word2vec (w2v)	ft + w2v
HVSL	25.8	25.3	31.8
SPNet	**33.1**	**32.1**	**35.2**

Table 2: Effect of word embeddings: Mean IoU of unseen classes in ZLSS with different word2vec, fastText and their combination on COCO-Stuff. Both HVSL and SPNet are based on ResNet101.

Implementation details. We implement our SPNet model with PyTorch [33]. We apply ImageNet pretrained VGG-16 [43] and ResNet-101 [17] as our backbone to extract features, and our model is built on the DeepLab-v2 [9] that first extract features and apply atrous spatial pyramid pooling layer to produce the visual features, whose dimension is the same as the dimension of the semantic embedding space (i.e., 300 for fast-text and word2vec; 600 for their concatenation). In this work, for VGG backbone we apply Adam solver [21] with initial learning rate 1.0×10^{-4}, and for ResNet we use SGD with initial learning rate 2.5×10^{-4}. Following [9], we use the "poly" learning rate policy where current learning rate is the initial one multiplied by $(1 - \frac{iter}{max\_iter})^{power}$, and we set power to 0.9. Momentum and weight decay are set to 0.9 and .0005.

4.1. Zero-Label Semantic Segmentation Task

One of the contributions of our work is to propose a new task of zero-label semantic segmentation (ZLSS). In this section, we propose two benchmarks with zero-label data splits and detail the zero-label evaluation protocol.

Proposed zero-label dataset splits. The zero-label assumption, i.e. similar to the zero-shot assumption [51], states that none of the pixel values of the query images are allowed to belong to the classes that were used in any part of the training procedure, i.e. be it the model training or CNN training. This means that as CNNs are commonly trained on ImageNet 1K, none of the test classes should overlap with it. Following this rule, in COCO-Stuff dataset, we create a new zero-label class split by selecting 15 classes as unseen and the rest of the 167 classes as seen classes as they appear in ImageNet 1K which was used to pretrain ResNet.

In contrast to zero-shot image classification, we do not remove images that contain unseen classes from the training set, otherwise most of training images will be eliminated because seen and unseen classes co-occur frequently. Instead, we utilize the whole training set but ignore the labels of pixels belonging to unseen classes during training, i.e. these pixels do not effect the loss we optimize in any stage of the training. For PASCAL-VOC, since (a) only 4 classes are unseen in ImageNet 1K, (b) one of the candidate class 'person' has no semantically similar class present in the dataset, (c) all vehicles appear in ImageNet thus reducing candidate diversity - we simply take the first 15 classes as seen classes and the last 5 classes as unseen classes. We use the train/val split provided by the COCO-Stuff dataset: 118K training images as our training set and 5K validation images as our test set, and PASCAL-VOC: 11K training images and 1.4K test images. Following the cross-validation procedure of [51], we further hold out a subset of training classes as our validation set for tuning hyperparameters. More details about our data splits are shown in Table 1.

Evaluation protocol. The intersection-over-union (IoU), i.e. the standard evaluation criteria commonly used in semantic segmentation, quantizes the overlap between the predicted mask and the target mask. It is defined to be the size of the intersection between predicted and target regions divided by the union of them. For each class, its mean IoU is computed by averaging the IoU over all the query images.

In ZLSS, as the test-time search space is restricted to be unseen classes we report the mean IoU averaged over unseen classes. In GZLSS, the search space becomes the union of seen and unseen classes. In analogy to generalized zero-shot image classification [51], we report the mean IoU on seen classes, the mean IoU on unseen classes and the harmonic mean (H) of them, which is defined as,

$$H = \frac{2 * mIoU_{seen} * mIoU_{unseen}}{mIoU_{seen} + mIoU_{unseen}} \quad (7)$$

where $mIoU_{seen}$ and $mIoU_{unseen}$ represents the mean IoU of seen classes and unseen classes respectively. Similarly, in few-label semantic segmentation, we report mean IoU on unseen classes, but in generalized few-label semantic segmentation, the mean IoU over all classes is reported.

4.1.1 SPNet Model Analysis for ZLSS

In this section, we provide an extensive evaluation for different design choices of our model.

Effect of word embeddings. We compare our SPNet model with HVSL and study the effect of different word embeddings in Table 2. We investigate three types of word embeddings, i.e. fastText, word2vec and their concatenation. Our first observation is that SPNet performs significantly better than HVSL wrt. all the word embedding

	COCO-Stuff	PASCAL VOC
SPNet-VGG	26.3	47.4
SPNet-ResNet101	**35.2**	**49.5**

Table 3: Effect of CNN architectures: ZLSS with different CNN architectures, i.e. VGG and ResNet101 on COCO-Stuff and PASCAL-VOC. Word embedding is the ft + w2v.

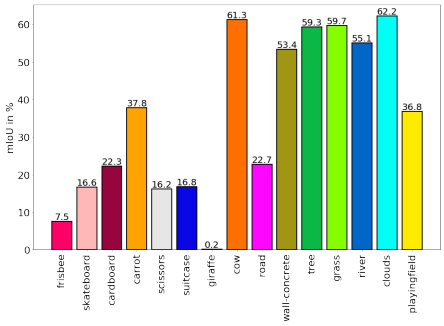


Figure 3: mIoU of unseen classes on COCO-Stuff ordered wrt average object size (left to right).

types, e.g. SPNet achieves 33.1 vs 25.8 with fastText, and 32.1 vs 25.3 with word2vec compared to HVSL. This implies that the cross-entropy loss is more suitable to the ZLSS task than hinge loss. Furthermore, we observe that fastText and word2vec achieve comparable results, and combining them significantly boosts the performance, e.g. mean IoU of SPNet are improved from 33.1 and 32.1 to 35.2. This indicates that fastText and word2vec contain complementary information. Hence, for the rest experiments, we use SPNet with fastText and word2vec combined.

Effect of CNN architectures. Our aim here is to compare different CNN architectures that are used as the backbone network to encode images in DeepLab-v2 [9]. Table 3 shows the ZLSS results with VGG16 [43] and ResNet101 [17]. We first observe that with VGG16, the results are lower than with ResNet101 on both COCO-Stuff and PASCAL-VOC which implies that ResNet101 generate stronger features than VGG16 for this task. Besides, these results show that our SPNet achieves reasonably good results in ZLSS with both CNN architectures. Specifically, on COCO-stuff, SPNet obtains 26.3% mIoU with VGG16 and 35.2% mIoU with ResNet101. This is promising because our model does not require expensive dense pixel-level annotations for each class, e.g. it is not trained with any of the 15 unseen class labels of COCO-Stuff. This also indicates that our model is easily adapted to various semantic segmentation architectures.

Effect of the object size. We study the difficulty of zero-label semantic segmentation as a function of object sizes.

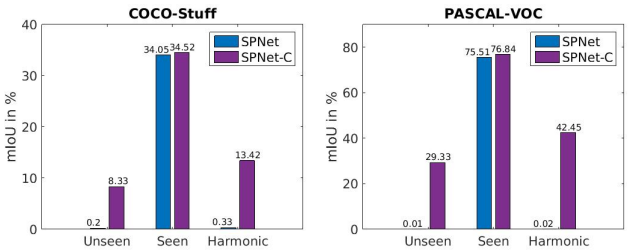


Figure 4: GZLSS results on COCO-Stuff and PASCAL-VOC. We report mean IoU of unseen classes, seen classes and their harmonic mean (perception model is based on ResNet101 and the semantic embedding is ft + w2v). SPNet−C represents SPNet with calibration.

Figure 3 presents a plot of per class mIoU score for the unseen classes in COCO-Stuff. The classes are ordered according to their average object sizes – with the largest on the right. It shows that there is a tendency that the performance is better for classes with larger objects. The plot also indicates that the knowledge transfer from seen to unseen classes is in general successful for the challenging stuff classes, such as, tree (59.3%), grass (59.7%, clouds (62.2%), considering the fact that they do not have semantically similar classes present in ImageNet 1K. We also observe that our model performs well for cow (61.3%) however the result is quite poor the other unseen animal class giraffe (0.2%).

4.1.2 Generalized Zero-Label Semantic Segmentation

GZLSS is a practical segmentation setting as the test time search space contains both seen and unseen classes, i.e. the pixel can be assigned to one of the seen or one of the unseen classes. Since the training images contain only labeled pixels of seen classes, at the test time, prediction will be biased to seen classes. Hence, this is a particularly challenging task. We alleviate this issue by using the calibrated classifier formulated in Eq. (5), which reduces the prediction scores of seen classes by a calibration factor γ. We select the optimal γ value based on the best harmonic mean IoU on a held-out validation set. Figure 4 shows the mean IoU on unseen classes, seen classes and their harmonic mean on COCO-Stuff and PASCAL VOC datasets.

On COCO-Stuff SPNet obtains 0.2% mean IoU on unseen classes while IoU on seen classes is high, i.e. 34.05%. This is expected, in fact the same trend is observed in generalized zero-shot image classification task [51, 8]. On the other hand, after calibration i.e. SPNet−C, on COCO-Stuff, mean IoU of unseen classes jumps to 8.33% while maintaining high mIoU on seen classes, i.e. 34.52% and overall SPNet−C achieves a harmonic mean of 13.42%. This is due to the fact that after calibration, i.e. reducing

	ZSL			GZSL		
	CUB	SUN	AWA	CUB	SUN	AWA
ALE [1]	54.9	58.1	59.9	34.4	26.3	**27.5**
SJE [2]	53.9	53.7	65.6	33.6	19.8	19.6
SYNC [7]	56.3	55.6	54.0	19.8	13.4	16.2
GFZSL [45]	49.3	60.6	**68.3**	0.0	0.0	3.5
SPNet	**56.5**	**60.7**	66.2	**36.6**	**39.6**	24.7

Table 4: SPNet loss on (generalized) zero-shot learning tasks. Top-1 accuracy on unseen classes is reported for ZSL and harmonic mean of seen and unseen classes is for GZSL.

prediction scores of seen classes, pixels get predicted as seen classes less frequently.

On PASCAL-VOC we observe a similar trend. While SPNet performs poorly on unseen classes, i.e. 0.01% mIoU, with calibration this increases to 29.33% mIoU. Accordingly, SPNet-C achieves an impressive 42.45% harmonic mIoU. These results demonstrate that our SPNet does not only tackle ZLSS but also can handle the more practical GZLSS via predictor calibration.

4.1.3 (Generalized) Zero-Shot Image Classification

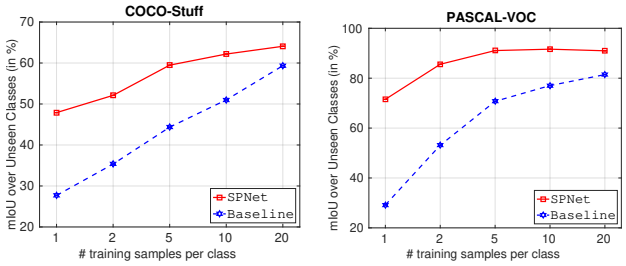
We evaluate our SPNet on the zero-shot image classification task on three benchmark datasets, i.e. CUB [49] (200 types of birds with 312 attributes), SUN [35] (717 scenes with 102 attributes) and AWA [23] (50 classes of animals with 85 attributes) with various sizes and complexities, following the data splits and evaluation protocol of [51]. We train SPNet with cross-entropy loss:

$$L(x, y) = -\log \frac{\exp\left(\phi(x)^\top V w_y\right)}{\sum_{c \in \mathcal{S}} \exp\left(\phi(x)^\top V w_c\right)} \qquad (8)$$

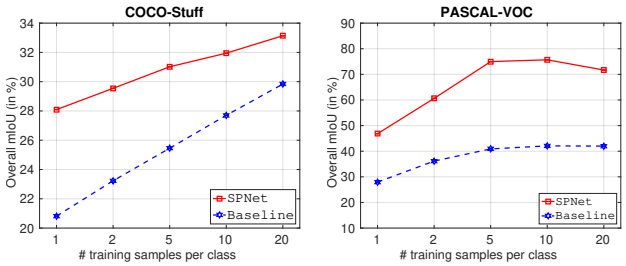
where $\phi(x)$ is 2048-dim image feature extracted from a pre-trained ResNet101 (no fine-tuning on the task), $w_c \in \mathbb{R}^{d_w}$ is the class attribute of class c, $V \in \mathbb{R}^{2048 \times d_w}$ is the linear embedding we aim to learn. Table 4 shows that both in ZSL and GZSL settings, our SPNet improves over the state of the art on both CUB and SUN while it obtains the second best results on AWA despite the simplicity of our model. Both ALE [1] and SJE [2] utilize the visual-semantic hinge loss, SYNC [7] align visual and semantic embedding space using manifold learning, and GFZSL [45] learns a generative model to capture the class conditional distribution. However, our SPNet simply projects image feature into the class embedding space and apply the standard softmax classifier with the class embedding being the weights.

4.2. Few-Label Semantic Segmentation Task

The (Generalized) few-label semantic segmentation (FLSS and GFLSS) tasks arise in many real-world applications



(a) Few-Label Semantic Segmentation



(b) Generalized Few-Label Semantic Segmentation

Figure 5: (Generalized) few-label semantic segmentation on COCO-Stuff and PASCAL VOC with increasing number of training samples per class, i.e. $n \in \{1, 2, 5, 10, 20\}$.

since class distribution in semantic segmentation is usually skewed, e.g. there are far more road pixels than bicycles. In contrast to ZLSS where the training set has no labeled example from unseen (novel) classes, in FLSS and GFLSS, the model is trained with all classes. At the evaluation time, the goal of FLSS is to segment only the novel classes, while GFLSS aims to segment both base and novel classes. For each novel class, we randomly draw $n \in \{1, 2, 5, 10, 20\}$ images that contain this class from the training set and disable ignore-label condition for those novel pixels. In addition, we develop a simple baseline based on the original DeepLab-v2 [9], which is finetuned on novel classes after an initial optimization on base classes. We carry out experiments in FLSS and GFLSS with the baseline and our SPNet on COCO-Stuff and PASCAL-VOC.

In FLSS task, Figure 5 (a) shows the comparison results with the baseline model [9]. Our SPNet yields significantly better results than the baseline in all cases on both COCO-Stuff and PASCAL VOC. In particular, when there is only 1 labeled example, our SPNet significantly outperforms the baseline, achieving a mean IoU of 47.90% over 27.69% in COCO-Stuff and 71.52% over 29.17% in PASCAL VOC on FZLSS. The accuracy improvement from 1 labeled sample to 5 labeled samples is significant, i.e. $\approx$ 20% mIoU for both COCO-Stuff and PASCAL VOC. These results demonstrate the effectiveness of our SPNet when the training samples are scarce.

As for GFLSS in Figure 5 (b), a similar trend is observed. Our SPNet improves over DeepLab in all cases. The accu-

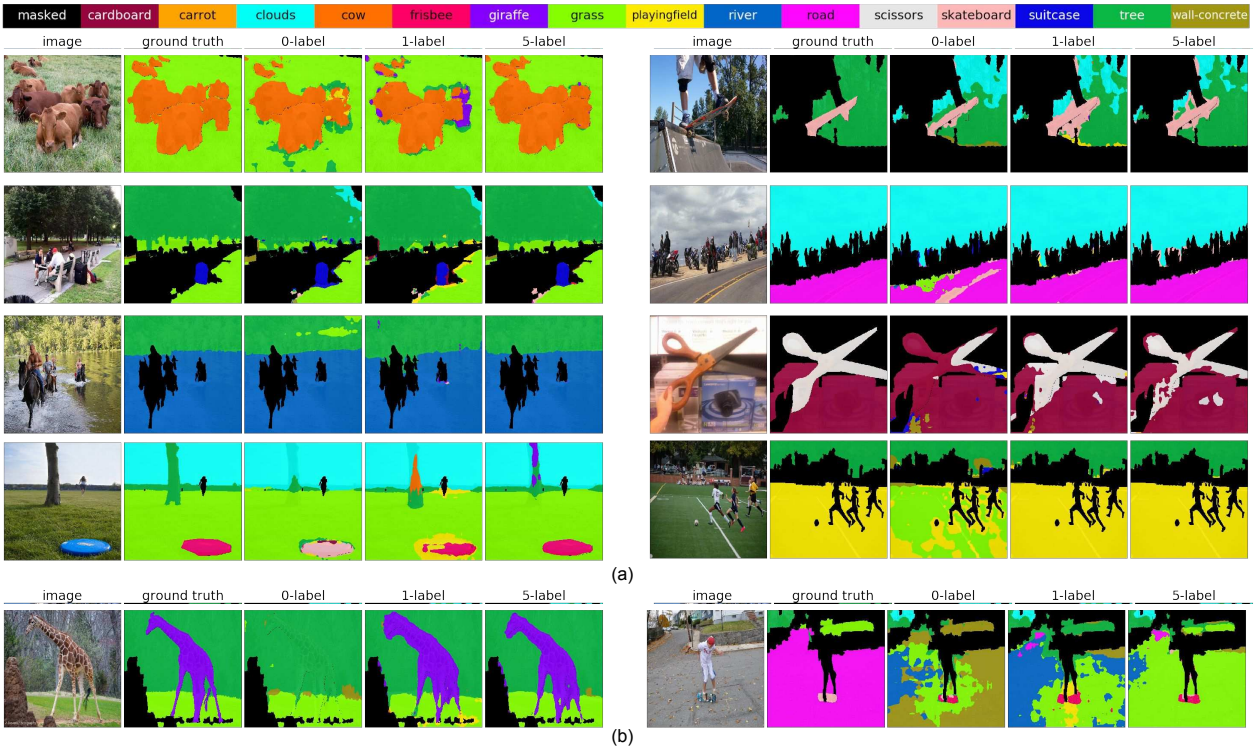


Figure 6: Qualitative results of our SPNet in 0-, 1- and 5-label semantic segmentation settings on COCO-Stuff on 15 novel classes (color coded at the top). Base classes are masked out with black color. (a) promising results (b) failure cases.

racy improvement is steady from 1 to $2, 5, 10, 20$ especially on COCO-Stuff. The difference between DeepLab and ours is 21.24% mIoU over both seen and unseen classes on PASCAL VOC when our model has access to only one labeled sample from novel classes.

4.3. Qualitative Results

Figure 6 shows the qualitative results obtained by our SPNet in ZLSS and FLSS on COCO-Stuff. Our target 15 novel classes are encoded with the colors shown at the top. Base classes are masked out with black color. Some interesting results are as follows. In the first row and left column, our SPNet is already able to segment two previously unseen classes cows and grass at ZLSS, i.e. 0-label, and results get refined after the model sees more examples. It is also worth noting that our SPNet is able to predict stuff classes, such as road, river, clouds etc., in ZLSS setting. For instance, SPNet successfully segments clouds and roads in the image at the second row and right column, and perfectly segments the river in the image at the third row and left column. Another interesting result is in the left column of 4th row where the model correctly segments the frisbee in 0-label setting but incorrectly labels most pixels as 'skateboard' which in fact is another sports category object. On the other hand, some failure cases are shown in the bottom row. Our SPNet fails to predict giraffe at 0-label because shape and appearance of a giraffe vary significantly from seen classes. However, seeing only 1 example is enough to recognize and segment it, which demonstrates the ability of our SPNet in learning from few examples. Again, the result gets refined with 5 labeled examples.

These results support our observations in the previous sections and indicate that our SPNet, although simple, adapts its knowledge attained in previously seen examples to unseen ones.

5. Conclusions

In this work, we propose SPNet to semantically segment novel classes with no labeled examples or with only a few samples, within the new tasks of zero-label semantic segmentation and few-label semantic segmentation respectively. This model consists of a visual-semantic embedding module that encodes images in the word embedding space and a semantic projection layer that produces class probabilities. Our SPNet is both conceptually and computationally simple but surprisingly effective and end-to-end trainable. We have shown its applicability across zero-shot image classification to zero-label and few-label semantic segmentation tasks on various benchmark datasets.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *TPAMI*, 2016. 2, 4, 7

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015. 1, 2, 7

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 2017. 2

[4] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran. Zero-shot object detection. In *ECCV*, 2018. 2, 4

[5] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 1, 2

[6] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 4

[7] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016. 1, 7

[8] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 4, 6

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2018. 2, 3, 5, 6, 7

[10] Z. Ding, M. Shao, and Y. Fu. Low-rank embedded ensemble semantic dictionary for zero-shot learning. In *CVPR*, 2017. 2

[11] N. Dong and E. P. Xing. Few-shot semantic segmentation with prototype learning. BMVC, 2018. 1, 2

[12] 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. 4

[13] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 2, 4

[14] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *TPAMI*, 2015. 2

[15] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[18] J. Ji, S. Buch, A. Soto, and J. C. Niebles. End-to-end joint semantic segmentation of actors and actions in video. In *ECCV*, 2018. 2

[19] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016. 2, 4

[20] A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 1, 2

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, 2015. 2

[23] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 2013. 1, 2, 7

[24] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. F. Wang. Multi-label zero-shot learning with structured knowledge graphs. In *CVPR*, 2018. 2

[25] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 1, 2

[26] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *ICLR workshop*, 2016. 2

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 3

[28] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin. Advances in pre-training distributed word representations. In *LREC*, 2018. 4

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 2, 3, 4

[30] G. A. Miller. Wordnet: a lexical database for english. *CACM*, 1995. 2

[31] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, B. Schiele, et al. Exploiting saliency for object segmentation from image level labels. In *CVPR*, 2017. 1, 2

[32] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 1, 2

[33] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 5

[34] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR workshop*, 2015. 1, 2

[35] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 7

[36] R. Qiao, L. Liu, C. Shen, and A. v. d. Hengel. Less is more: Zero-shot learning from online textual documents with noise suppression. In *CVPR*, 2016. 2

[37] K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine. Conditional networks for few-shot semantic segmentation. ICLR workshop, 2018. 2

[38] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2016. 2

[39] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 2

[40] Z. Ren, H. Jin, Z. Lin, C. Fang, and A. Yuille. Multiple instance visual-semantic embedding. In *BMVC*, 2017. 2

[41] B. Romera-Paredes and P. H. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 2

[42] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017. 1, 2

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 6

[44] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 2

[45] V. K. Verma and P. Rai. A simple exponential family framework for zero-shot learning. In *ECML*, 2017. 7

[46] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016. 2

[47] X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018. 4

[48] Y. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018. 2

[49] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, Caltech, 2010. 7

[50] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 2, 4

[51] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2018. 2, 5, 6, 7

[52] X. Xu, T. Hospedales, and S. Gong. Transductive zero-shot action recognition by word-vector embedding. *IJCV*, 2017. 2

[53] X. Xu, T. M. Hospedales, and S. Gong. Multi-task zero-shot action recognition with prioritised data augmentation. In *ECCV*, 2016. 2

[54] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 2

[55] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017. 2

[56] X. Zhang, Y. Wei, Y. Yang, and T. Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *arXiv preprint arXiv:1810.09091*, 2018. 2

[57] Z. Zhang and V. Saligrama. Zero-shot learning via joint semantic similarity embedding. In *CVPR*, 2016. 1, 2, 4

[58] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba. Open vocabulary scene parsing. In *ICCV*, 2017. 2

[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2

[60] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2