SEMI-SUPERVISED GENERATIVE GUIDANCE FOR ZERO-SHOT SEMANTIC SEGMENTATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ABDULLAH CEM ÖNEM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2022

Approval of the thesis:

SEMI-SUPERVISED GENERATIVE GUIDANCE FOR ZERO-SHOT SEMANTIC SEGMENTATION

submitted by **ABDULLAH CEM ÖNEM** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ______________

Assist. Prof. Dr. Ramazan Gökberk Cinbiş
Supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Assist. Prof. Dr. Emre Akbaş
Computer Engineering, METU ______________

Assist. Prof. Dr. Ramazan Gökberk Cinbiş
Computer Engineering, METU ______________

Assoc. Prof. Dr. Erkut Erdem
Computer Engineering, Hacettepe University ______________

Date: 13.01.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Abdullah Cem Önem

Signature :

ABSTRACT

SEMI-SUPERVISED GENERATIVE GUIDANCE FOR ZERO-SHOT SEMANTIC SEGMENTATION

Önem, Abdullah Cem
M.S., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Ramazan Gökberk Cinbiş

January 2022, 41 pages

Collecting fully-annotated data to train deep networks for semantic image segmentation can be prohibitively costly due to difficulty of making pixel-by-pixel annotations. In this context, zero-shot learning based formulations relax the labelled data requirements by enabling the recognition of classes without training examples. Recent studies on zero-shot learning of semantic segmentation models, however, highlight the difficulty of the problem. This thesis proposes techniques towards improving zero-shot generalization to unseen classes by exploiting unlabelled images. The main goal is to train a generative image model conditioned on zero-shot segmentation predictions in a semi-supervised manner, and use the feedback from the generative model to the segmentation based conditioning inputs as a guidance. In this manner, the zero-shot segmentation model is encouraged to make more accurate predictions so that it provides more informative conditional inputs to the generative model. To further improve the training dynamics of the generative model, the generative model is trained in the feature space provided by the early convolutional layer(s) of the segmentation architecture, overall forming a high-level to low-level generative feedback loop. Following the state-of-the-art, the approach is experimentally evaluated using the COCO-Stuff dataset.

Keywords: zero-shot semantic segmentation, zero-shot learning, computer vision, semantic segmentation, semi-supervised semantic segmentation, generative networks

ÖZ

SIFIR-ÖRNEK ANLAMSAL GÖRÜNTÜ BÖLÜTLEMEYE YÖNELİK YARI-DENETİMLİ ÜRETİCİ YÖNLENDİRME

Önem, Abdullah Cem
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Ramazan Gökberk Cinbiş

Ocak 2022 , 41 sayfa

Anlamsal görüntü bölütlemeye yönelik derin ağlar için veri etiketleme süreci piksel başına işaretlemenin zorluğundan ötürü engelleyici ölçüde maliyetli olabilmektedir. Bu bağlamda, sıfır-örnekli öğrenme tabanlı yöntemler, yeni sınıfların eğitim örneği gerektirmeden tanınmasını sağlayarak etiketli veri ihtiyacını azaltmaktadır. Fakat anlamsal bölütlemede sıfır-örnekli öğrenme üzerine olan son dönem çalışmaları problemin zorluğunun altını çizmektedir. Bu tezde, görülmeyen sınıflara yönelik sıfır-örnek genelleştirmenin etiketlenmemiş görüntülerden yararlanılarak iyileştirilmesine yönelik bir yaklaşım önerilmektedir. Ana amaç, sıfır-örnekli bölütleme tahminleri ile koşullu bir üretici görüntü modelini yarı-denetimli bir biçimde öğrenmek ve üretici modelden bölütleme tabanlı koşul girdilerine doğru olan geri beslemeyi yönlendirici olarak kullanmaktır. Bu yöntemle, sıfır-örnek bölütleme modelinin, üretici modele daha bilgilendirici koşul girdileri sunabilecek şekilde daha doğru tahminler yapmaya yönlendirilmesi hedeflenmektedir. Ek olarak üretici model, modelin eğitim dinamiklerini geliştirmek için bölütleme mimarisinin öncül evrişimsel katmanlarından elde edilen öznitelik uzayında tanımlanmakta, sonuç olarak yüksek seviyeden düşük seviyeye üretici geri besleme döngüsü oluşturulmaktadır. Yaklaşım yakın dönem çalışmalarla uyumlu olarak COCO-Stuff veri kümesinde deneysel olarak incelenmektedir.

Anahtar Kelimeler: sıfır-örnekli anlamsal bölütleme, sıfır-örnekli öğrenme, bilgisayarlı görü, anlamsal bölütleme, yarı-denetimli anlamsal bölütleme, üretici ağlar

To My Family

ACKNOWLEDGMENTS

I would first like to thank my thesis supervisor Assist. Prof. Dr. Gökberk Cinbiş for guiding my research, helping my work and this thesis solidify with his invaluable feedback and know-how. I would be lost without his dedicated care arriving so late at this field.

I would like to thank Prof. Dr. Vittorio Ferrari for his reviews and suggestions regarding the experiments I have conducted during the last year of my M.S. studies as they matured into my thesis.

I would like to thank Prof. Dr. Uluç Saranlı for establishing my fundamentals as a researcher by taking me in his research group in my undergrad and early graduate years.

Lastly, I would like to thank my family for supporting me in my career and any aspect of life until the very end.

This work was supported in part by the TÜBİTAK Grant 119E597. The numerical calculations reported in this thesis were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

TRUBA	Turkish National e-Science e-Infrastructure
ILSRVC	ImageNet Large Scale Visual Recognition Challenge
ASPP	Atrous Spatial Pyramid Pooling
GZSSS	Generalized Zero-Shot Semantic Segmentation
GZLSS	Generalized Zero-Label Semantic Segmentation
HoG	Histogram of Gradients
CNN	Convolutional Neural Network
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
GAN	Generative Adversarial Network
VAE	Variational Auto-Encoder
GMMN	Generative Moment Matching Network
PGM	Parallel Generative Models
NLL	Negative Log-Likelihood
AGM	Autoregressive Gaussian Model
IoU	Intersection-over-Union

CHAPTER 1

INTRODUCTION

Recognition problems in computer vision comprises querying certain areas, objects, actions and other aspects within image, a task that humans can accomplish easily with little or even no prior information [1]. In the 21st century, video and image content generated worldwide has soared to a point it makes automation of this task not a convenience but a requirement for the industry [2].

Owing to parallel processor units becoming fast and affordable in the last decade, deep neural networks have been the dominant approach for tackling computer vision problems. Following the success of AlexNet in surpassing well-established computer vision techniques in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [3] by ~11% less Top-5 error rate [4], the field has flourished: year by year networks are further optimized, new machine learning formulations are invented and more data is available to train the models. For example, in the image classification task, the recently proposed semi-supervised classification approach Meta Pseudo Labels [5] achieves 98.8% ImageNet Top-5 accuracy, greatly outperforming the average human performance of 94.9%.

Semantic segmentation is another task benefited from deep learning approaches. In semantic segmentation, the goal is to classify each pixel in the image into a class out of a predefined set or a *background* [6]. Predicting exactly which part of the image belongs to a particular class beyond localization or identification is especially desired in robotics [7], medical applications [8] or generating content [9]. Example input - output pairs from the COCO-Stuff segmentation dataset [10] is shown in Figure 1.1.

Mainstream studies on semantic segmentation embrace a supervised training sce-

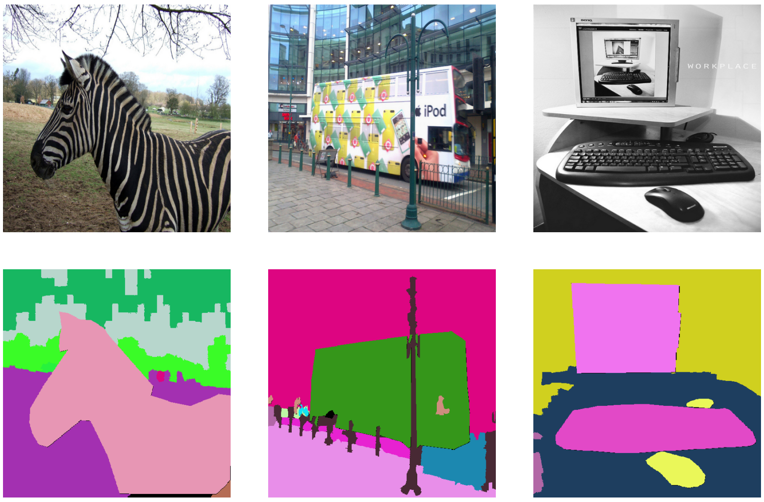

Figure 1.1: Example input - output pairs for the semantic segmentation problem.

nario. However, curating datasets for training supervised segmentation architectures is arduous and costly. Assuming an image contains 4-5 instances of various objects on average, annotating a single image by hand can take more than 5 minutes [11].

In this thesis, our goal is to improve performance with semi-supervised training schemes in the generalized zero-shot segmentation (GZSSS) objective, a scenario that ultimately aims to reduce dependence on large datasets. Our model is able to achieve better performance compared to the discriminative state-of-the-art in the COCO-stuff segmentation dataset. This chapter introduces the GZSSS problem, provides an overview of the issues that arise in recent discriminative implementations, summarizes the contributions of this work, and presents the outline of the rest of the text.

1.1 Problem definition

Relaxing the requirement for annotation has been the focus of some of the segmentation settings. Unlabeled data is vast and easily accessible compared to images with ground truth segmentations; within this frame of reference, semi-supervised semantic segmentation where the model has to learn over a mixture of unlabeled and labeled examples [12] is proposed.

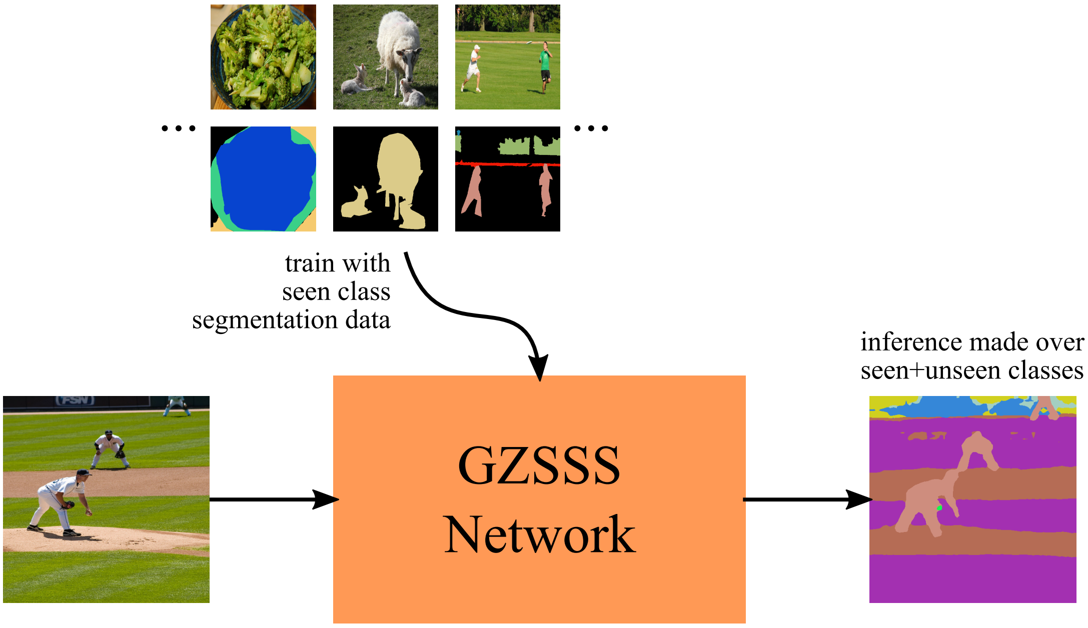

Figure 1.2: Generalized zero-shot semantic segmentation problem.

A more extreme task that aims to reduce the annotation effort is generalized zero-shot semantic segmentation (GZSSS). In this scenario, the model has access to the pixels and the ground truth labels of some of the classes in the training split of the dataset (seen classes), only the attributes of the other classes are provided (unseen classes) [13]. After training, the model is expected to make inference on both seen and unseen classes. A generic example of a GZSSS network is shown in Figure 1.2. Here, the model has no ground truth data regarding the unseen class *grass* (shown purple), but it is able to infer it through the information given with seen classes *broccoli*, *sheep*, *bush* and their semantic relationship to *grass*.

Adjusting segmentation datasets to the GZSSS objective results in thin training splits as most of the images contain unseen classes if proper class splits are sought [14]. To that end, generalized zero-label semantic segmentation (GZLSS) protocol is proposed, where the model has access to the pixels belonging to unseen classes in the training splits in addition to the information in GZSSS. Most recent work on GZSSS employ the GZLSS protocol and classify it under GZSSS [13, 15, 16]. We adopt this nomenclature and use the term GZSSS for both zero-label and zero-shot segmentation.

1.2 Proposed methods and models

GZSSS methods with discriminative backbones such as SPNet [14] can be heavily biased towards seen classes as only the seen class pixels with ground truth labels are available during training. Discriminative GZSSS approaches usually handle this issue by ad-hoc modification of classification scores after inference, for instance SPNet deducts a calibration factor from softmax classification scores. These parameters are brittle [17] and necessitate careful tuning with a validation set for each particular domain.

To aid this problem, we propose parallel generative models (PGM) attached to the early and middle layers of the discriminative backbone, conditioned on model classification scores of pixels. Assuming unseen and seen class features in the layers are distribution-wise distinct, the gradient signals in the conditional generative models are expected to direct the conditional inputs towards correct classes, since it is likely to be easier for the generative model to fit features to a relatively simple distribution for each class (seen+unseen) rather than fitting them into a complex distribution of features incorrectly marginalized to seen classes.

The PGM also makes it possible to train the GZSSS network in a semi-supervised setting where unlabeled data within the dataset can be utilized to regularize the generative models, which in turn directs the classification scores of the pixels coming from the GZSSS model. We use this setting to further exploit the additional information coming from unlabeled pixels to increase the segmentation quality of the network.

1.3 Contributions and novelties

The contributions of this thesis are listed as follows:

- We propose a way to reduce bias towards seen classes and improve unseen class classification performance in GZSSS networks as parallel generative models attached to the early and middle layers.
- Our method makes it possible for GZSSS networks to train in a semi-supervised

setting.

- Our method obtains competitive scores in COCO-Stuff dataset with GZSSS network SPNet.

1.4 Outline of the thesis

The rest of the thesis is organized as follows: Chapter 2 surveys the relevant literature and provides a summary of previous zero-label and semi-supervised segmentation methods. Chapter 3 lays out the formulation of semi-supervised zero-label segmentation setting and elaborates our approach. Chapter 4 presents our experimental settings, baselines, hyper-parameter tuning methods and results with respect to state-of-the-art discriminative GZSSS network, SPNet [14]. Chapter 5 concludes the thesis with future work regarding enhanced discriminative GZSSS models with PGM.

CHAPTER 2

RELATED WORK

In this chapter, we provide a literature review of the previous studies relevant to our work. We first discuss developments in supervised semantic segmentation before and after deep learning is incorporated into the area. We then present an overview of the recent semi-supervised semantic segmentation and zero-shot classification methods. Finally, we lay out an in-depth summary of the few generalized zero-shot segmentation approaches and point out the differences and contributions made in this thesis in comparison to these studies.

2.1 Semantic segmentation

Early semantic segmentation works employ a combination of hand-crafted features with traditional classifiers. Shotton et al. [18] utilize random decision forests [19] with textons at each node to classify pixels within an image, in [20] this idea is extended to other hand-crafted features such as histogram of gradients (HoG) [21]. Plath et al. [22] calculate scale invariant feature transforms (SIFT) [23] of the patches within the image extracted with a graph-cut and classify the patches with support vector machines (SVM) [24]. Post-processing methods in segmentation such as the application of conditional random fields (CRF) [22, 25] from this era are also adopted in recent works [26].

Later studies infer segmentation masks and region level labels over CNN features extracted from regions within the image. R-CNN [27] employs region proposal methods to collect these regions. This technique is improved with Faster R-CNN [28] where the region proposal method is replaced with a neural network, and Mask R-CNN [29] where an object mask prediction branch is attached alongside the bounding-box prediction branch. Other approaches gather regions through image pyramids [30] and zoom out from superpixels [31].

State-of-the-art architectures for semantic segmentation are mostly end-to-end networks that strive to collect contextual information for each pixel in different scales. PSPNet [32] has pooling kernels of varying spatial size, Deeplabv3+ [33] incorporates atrous spatial pyramid pooling (ASPP) module with dilated convolutions, lowering parameter count. Recent Gated Shape CNN [34] adds a shape stream to the encoder before the ASPP module with gated convolutions, emphasizing activations regarding shape during training to improve quality around segmentation boundaries. Another way of enforcing contextual information is the approach taken in U-Net [35], where residual connections are added to the spatially matching layers of encoder-decoder architectures.

Our work and other GZSSS approaches [14, 13, 16, 17] utilize pre-trained supervised semantic architectures as backbones, since contextual feature similarities of a particular seen class pixel and an unseen class pixel would be high provided that the class labels are semantically adjacent (both "cow" and "giraffe" would appear in pastures).

2.2 Semi-supervised semantic segmentation

Semi-supervised semantic segmentation, where part of the training samples have no ground-truth segmentation data is a sought-out objective in medical imaging area due to high cost of annotation of training sets requiring area expertise [36]. Some of the works use self-training methods: Bai et al. [37] iteratively switch between training the network with ground-truth labels and pseudo-labels coming from unlabeled samples, [38] employs mean teacher approach [39] with transformation consistency regularization in pseudo-labels. Architectures utilizing unlabeled data with generative losses are available as well, [40] learns a variational auto-encoder (VAE) [41] over features of labeled and unlabeled images.

Few studies on non-medical domains for semi-supervised semantic segmentation also incorporate generative networks. Following [42], Souly et al. [43] use a similar setting with generative adversarial networks (GAN) [44] with the discriminator on double duty as the classifier. Hung et al. [45] attempt to fit a distribution onto the segmentation labels instead of images and eliminates pseudo-labels with low discriminator scores. s4GAN+MLMT [46] builds upon this work and adds a second branch with mean teacher framework inferring the set of labels that exist in the image. In [47] self-training nature of the networks are further improved with consistency losses on features of dual student architectures.

Two works where generative networks are fit on the joint distribution of images x and segmentation labels y have recently appeared. The main goal of these approaches is to model $p(x, y)$ instead of the usual classification scheme $p(y|x)$. DatasetGAN [48] trains an ensemble of shallow classifiers from StyleGAN [49] features, however the annotations have to be made on GAN outputs. [50] does not exhibit this problem. In this model, the image is transformed and fed as noise vectors in the style layers in addition to the original latent noise input of the generator. The generator outputs segmentation results and images in two branches that share some parameters. The branches are trained with reconstruction loss of the image, cross-entropy loss of the segmentation labels, and the loss coming from the discriminator for image-label pair.

2.3 Generalized zero-shot image classification

In generalized zero-shot image classification task, the classifier has no access to any image that belong to some of the classification labels (unseen classes), however semantic relationships are present between seen and unseen classes for the model to exploit [51]. Early approaches map image features and label attributes in a common space, then calculate the similarity of mapped image/label features for inference: DeViSE [52] projects image features from a CNN and word embeddings of class labels into the same space and trains the model on hinge loss over the similarity of the mappings, in ALE and SJE [53, 54] this loss is weighted with ranking terms. Kodirov et al. [55] design the projection of image features onto class embeddings as an autoencoder with a regularizing reconstruction loss.

Many of the recent studies make use of generative networks to create synthetic features of images of unseen classes to alleviate bias towards seen classes in the previous works. Xian et al. [56] employ GANs conditioned on class attributes, [57, 58] utilize the same approach with VAEs and generative moment matching networks (GMMN) [59] respectively. Sarıyıldız et. al [60] propose a regularizer loss that pushes the model to generate synthetic image features such that the gradient signals coming from the classifier match with the real features.

Some of the GZSSS approaches also follow this trend of exploiting generative models to produce artificial unseen feature samples in zero-shot classification and semi-supervised segmentation [15, 13, 16]. In contrast, we aim to leverage generative losses to guide and regularize the representation, rather than explicitly sampling synthetic features for training purposes.

2.4 Generalized zero-shot semantic segmentation

To the best of our knowledge, three works jump-started this area approximately in the same timeline [14, 13, 15]. Two significant precursor works that attack problems very adjacent to GZSSS are worth mentioning. In the first one, Nata et al. [61] attempt figure-ground ZSSS by devising a confidence score of the unseen figure class as a weighted sum of the seen class confidences of a supervised semantic segmentation network in terms of word embedding proximities to the unseen class. Then a binary mask of pixels is generated based on confidence scores over a threshold. A logistic regression classifier is then trained to output this binary mask over the final layers of the supervised semantic segmentation network. The second is the study of Zhao et. al [32], in which semantic hierarchy of labels are leveraged to make predictions on unseen classes by fitting word embeddings and pixels into a joint space, preserving the hypernym/hyponym relationship of words through enforcing a partial order of vector magnitudes of mappings. This way, the model falls back to a hypernym of an unseen class during inference if there is no exact match in terms of mapping similarity.

The architecture of [15] features an encoder-decoder setup where synthetic features conditioned on supplied class embeddings (to be segmented into foreground during inference) are concatenated pixelwise with the image encodings obtained from the encoder. The result is then converted to a binary classification map by the decoder. Unlike our proposed method, which outputs multi-way classification scores for each class in a single pass, the approach essentially results in a method of one-way zero-shot segmentation, requiring a separate inference step for each class. The model also offers no way of exploiting unlabeled images without pseudo-label training, since the synthetic feature generator depends on a conditional hard class label input compared to soft GZSSS classification scores in our model that enables semi-supervised settings.

In ZS3Net [13], pixel features obtained from a discriminative segmentation network are synthesized with a GMMN conditioned on class labels. After the generative network learns this conditional feature generation task, a classifier is trained over genuine features that belong to seen pixel classes and synthesized features conditionally generated with unseen class word-embeddings. The study also includes extensions of the generative network with graph convolutional layers[62] for better encoding of context in the synthesized features. The approach in ZS3Net differs from our approach in the way that only hard class labels are accepted as conditional input.

SPNet [14] is the only fully discriminative GZSSS pipeline to the best of our knowledge. The architecture is comprised of a backbone semantic segmentation network and a set of fixed word embeddings. The backbone network is trained to map the pixels to a feature space in which features of pixels should have high similarity scores with the corresponding word embedding of the correct class. To adhere the GZSSS scheme, only seen class labels are present in the word embedding set during training. This set is then replaced by the unseen+seen class embeddings at inference to allow the model to classify pixels as unseen classes. SPNet offers much more stable training owing to it containing no generative counterpart. It is also the only study other than [14, 15] to report GZSSS results with COCO-Stuff [10] dataset, which is more suitable as a testbed for real life scenarios due to wide range of annotated classes. However, the approach suffers a heavy bias towards seen classes at inference due to the discriminative nature and the training scheme of the model. Due to it being a state-of-the-art method in terms of results in COCO-Stuff with relatively stable training, we choose SPNet in the instantiation of our approach for the experiments. A basic theoretical overview of SPNet is provided in Chapter 3 with further hyper-parameter selections in Chapter 4.

Lastly, two works recently appeared in the scene of GZSSS. Pastore et al. [17] study pseudo-label training of SPNet and proposes ways of eliminating pseudo-labels with low confidence through checking consistencies of pseudo-label outputs of the data augmented versions of the same unlabeled image. CaGNet [16] builds on the idea of [13] but spatial context is encoded into synthetic features with dilated convolutions instead of object context. We leave comparison of our model with these architectures for future work due to time constraints and no immediate relation to the methods we propose.

CHAPTER 3

PROPOSED METHOD

In this chapter, we present the formulations regarding our approach. We first provide a definition of the GZSSS task and a theoretical summary of the discriminatory GZSSS network SPNet [14] that we employ as a baseline. Then, we formally define the generic PGM architecture and the training algorithm. Finally, we supply the details regarding an example instantiation of our generic definition, which is experimentally evaluated in Chapter 4.

3.1 GZSSS training and evaluation protocol

The training and evaluation protocol for GZSSS is defined as follows: let S be the set of seen class labels and U be the set of unseen class labels with $S \cap U = \emptyset$. Let the training set be D_{train} containing image label pairs $(x_{train},\ y_{train})$ with $x_{train} \in \mathbb{R}^{3\times H\times W}$, $y_{train} \in (S \cup U \cup \{u\})^{H\times W}$ for width W and height H, u standing for unlabeled pixels. In GZSSS [14], each pixel level label l_{ij} at location $i,\ j$ of y_{train} is replaced with

$$l^{'}_{ij} = \begin{cases} l_{ij} & \text{if } l_{ij} \in S \\ u & \text{otherwise} \end{cases} \tag{3.1}$$

to obtain $y^{'}_{train}$, comprising $D^{'}_{train}$ of pairs $(x_{train},\ y^{'}_{train})$. The information access is restricted to $D^{'}_{train}$ during training of a particular GZSSS model.

For metrics on the test set D_{test} composed of $(x_{test},\ y_{test})$, ground truth test labels may belong to either of the seen or unseen classes. However, unlabeled pixels ($l = u$) are ignored.

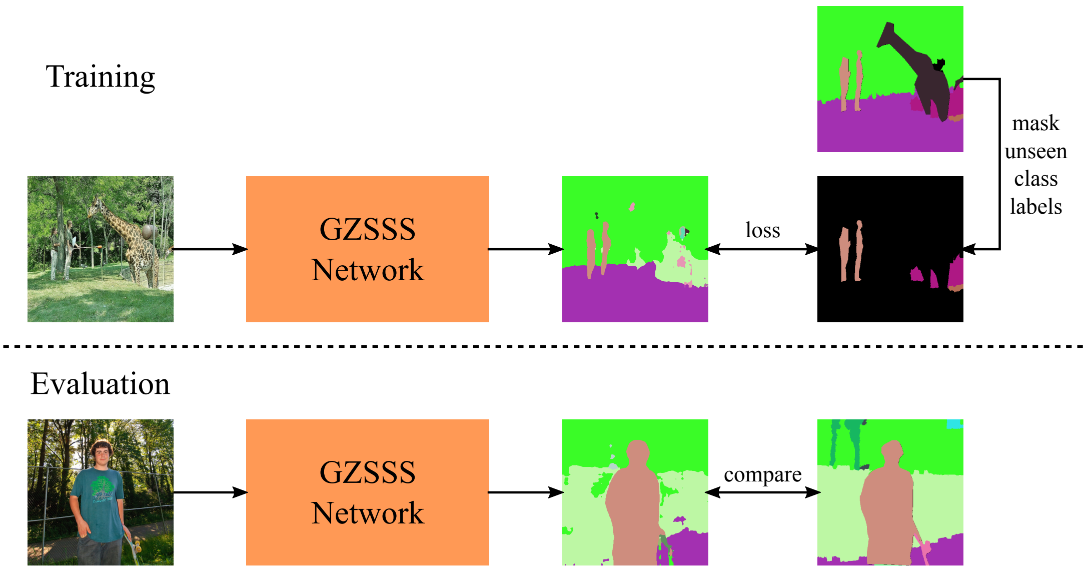

Figure 3.1: GZSSS training and evaluation protocol.

Figure 3.1 visualizes the GZSSS task. During training, ground truth regarding unseen classes is masked as unlabeled before it is incorporated into supervised training, however the visual information regarding unseen classes is still available through image pixels. During evaluation, both unseen and seen classes are evaluated with respect to the ground truth in the test split.

In transductive GZSSS, the model also has access to unlabeled test images x_{test}. We employ the transductive scheme in our experiments as a substitute for semi-supervised GZSSS where we have images with no segmentation data. Most of the current GZSSS works already incorporate unlabeled visual data of the unseen classes within the boundaries of the GZSSS task definition [14, 13, 15, 16, 17]. We believe this data leakage improves the model quality as the backbone segmentation networks most of these approaches possess collect visual spatial information through convolutional filters. In our implementation, we further exploit this notion by utilizing all unlabeled data to form a healthier distribution of features.

3.2 Theoretical overview of SPNet

We employ SPNet as a baseline in our experiments and use it as a component in the instantiation of our approach. SPNet consists of a visual-semantic embedding backbone ϕ that maps pixels into the same space with word embeddings of class labels. The inner product of the pixel features and the fixed word embeddings of the classes determine classification scores of the pixels.

Formally, let $w_c \in \mathbb{R}^d$ be the word embedding of class $c \in C$, normalized to a unit vector. Let $\phi(x)_{ij} \in \mathbb{R}^d$ be the pixel feature for spatial location i, j for an image x. The classification score $y^p_{ij}[c_0]$ for class c_0 is the estimated probability of y_{ij} as a softmax score:

$$y^p_{ij}[c_0] = \frac{\exp(w_{c_0}^\top \phi(x)_{ij})}{\sum_{c \in C} \exp(w_c^\top \phi(x)_{ij})} \tag{3.2}$$

with $y^p_{ij} \in [0,\ 1]^{|C|}$.

The model is trained with cross-entropy loss with the ground truth labels from the masked dataset in accordance with GZSSS task specification. Only the seen class word embeddings are available in training, C is substituted with seen classes S for the calculation in equation 3.2. The background class u is ignored during the calculation of the loss. For an image x and masked label y' the masked cross-entropy loss is defined as:

$$\sum_{i,j} -\mathbb{I}[y'_{ij} \neq u] \log(y^p_{ij}[y'_{ij}]) \tag{3.3}$$

where $\mathbb{I}$ is 1 if the expression in the brackets is true and otherwise 0.

At the inference step, C is switched out with $S \cup U$ for the model to be able to classify pixels as unseen classes. However due to the exclusion of unseen class labels and word embeddings from training SPNet suffers a heavy classification bias towards seen classes. As an ad-hoc solution, a calibration factor γ is deducted from classification scores before the class with largest classification score is selected:

$$\underset{c \in S \cup U}{\operatorname{argmax}}\, y^p_{ij}[c] - \gamma \mathbb{I}[c \in S] \tag{3.4}$$

The calibration factor γ is a hyper-parameter of the model.

3.3 Parallel generative models

Discriminative GZSSS networks suffer from heavy bias towards seen classes unless ad-hoc modifications are made to unseen/seen class likelihoods after inference [14].

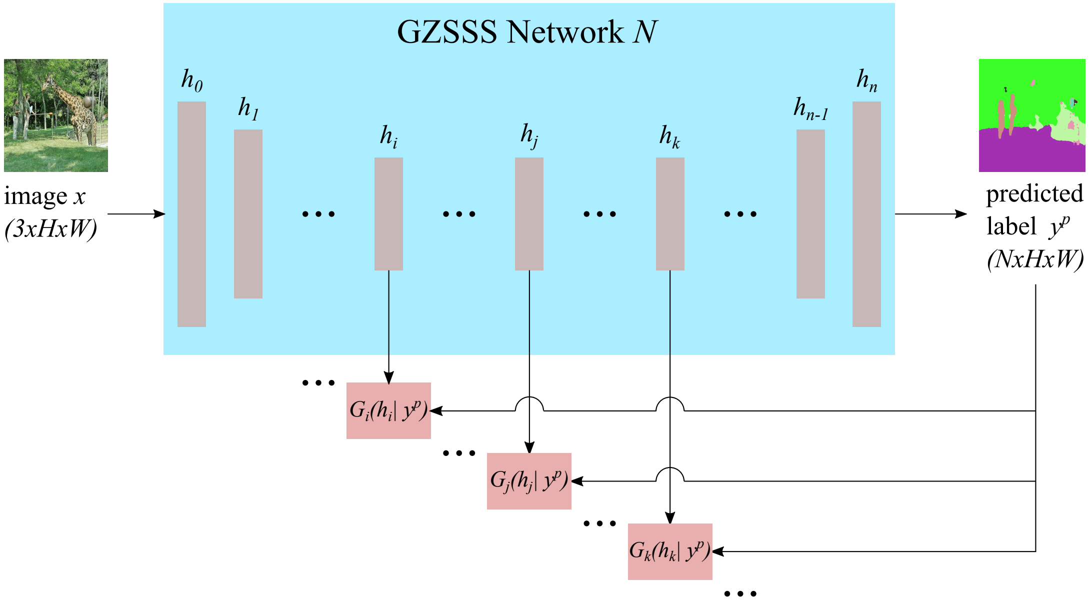

Figure 3.2: Model agnostic diagram of parallel generative networks.

We propose parallel generative models (PGM) to alleviate this bias in a more principled and less brittle way.

We formally define PGM as follows: Let N be a GZSSS network with parameters θ, layers $h_0,\ h_1 \cdots h_{n-1},\ h_n$ taking images x as input with multiway class prediction scores for each pixel $y^p = N(x;\theta)$, θ differentiable with respect to y^p. An array of generative networks $G_q(h_q|y^p;\theta_q)$ for some h_q are trained in this scheme to fit conditional distribution $h_q \sim H_q|y^p$ implied by N and the dataset D containing images x. The scheme does not require G_q to represent or sample the estimated distribution $\widetilde{H_q}|y^p$ explicitly, the only criteria for the loss metric L is that it should be differentiable with respect to y^p. Other than that, the metric decision to fit $H_q|y^p$ is made according to the nature of G_q.

Figure 3.2 showcases the model agnostic architecture. Here, an arbitrary GZSSS network makes inference on a certain image x. The output y^p is supplied as a conditional input to certain generative networks $G_i,\ G_j,\ G_k$. These networks in turn model hidden layer distributions $h_i,\ h_j,\ h_k$ respectively.

Algorithm 1 displays the generic training scheme for PGM. The G_q are first pretrained with fully trained N with fixed θ. After $\widetilde{H_q}|y^p$ is appropriately close to $H_q|y^p$,

Algorithm 1 Training with parallel generative models

Given: dataset D, fully trained GZSSS network N with parameters θ, array of generative models G_q with parameters θ_q, training iteration $a_N(D)$ for N, loss function L for G_q.

```
repeat
    sample images $x$ from $D$
    $y^p \leftarrow N(x; \theta)$
    for each $G_q$ do
        extract $h_q$ from $N(x; \theta)$
        $\theta_q \leftarrow \theta_q - \nabla_{\theta_q} L(G_q(h_q | y^p; \theta_q))$
    end for
until all $\theta_q$ converge
repeat
    $\theta \leftarrow a_N(D)$
    sample images $x$ from $D$
    $y^p \leftarrow N(x; \theta)$
    for each $G_q$ do
        extract $h_q$ from $N(x; \theta)$
        $\theta_q \leftarrow \theta_q - \nabla_{\theta_q} L(G_q(h_q | y^p; \theta_q))$
        $\theta \leftarrow \theta - \frac{\partial L}{\partial y^p} \frac{\partial y^p}{\partial \theta} L(G_q(h_q | y^p; \theta_q))$
    end for
until all $\theta,\ \theta_q$ converge
```

gradient signals from y^p are allowed to perturb N while supervision from the default training anchors the model. This should eventually result in lowered bias towards seen classes in y^p so that G_q can easily model a simpler posterior $H_q | y^p$ of seen and unseen classes instead of complex distributions marginalized to seen classes. Note that $\frac{\partial L}{\partial h_q} \frac{\partial h_q}{\partial \theta}$ does not contribute in the gradient descent of θ. This is because if we allow direct gradients from L to h_q the loss could easily deteriorate H_q. Limiting the gradient descent to $\frac{\partial L}{\partial y^p} \frac{\partial y^p}{\partial \theta}$ instead would result in weak gradient signals at early layers h_q that can be safely deemed negligible. That way we can modify $H_q | y^p$ without affecting H_q.

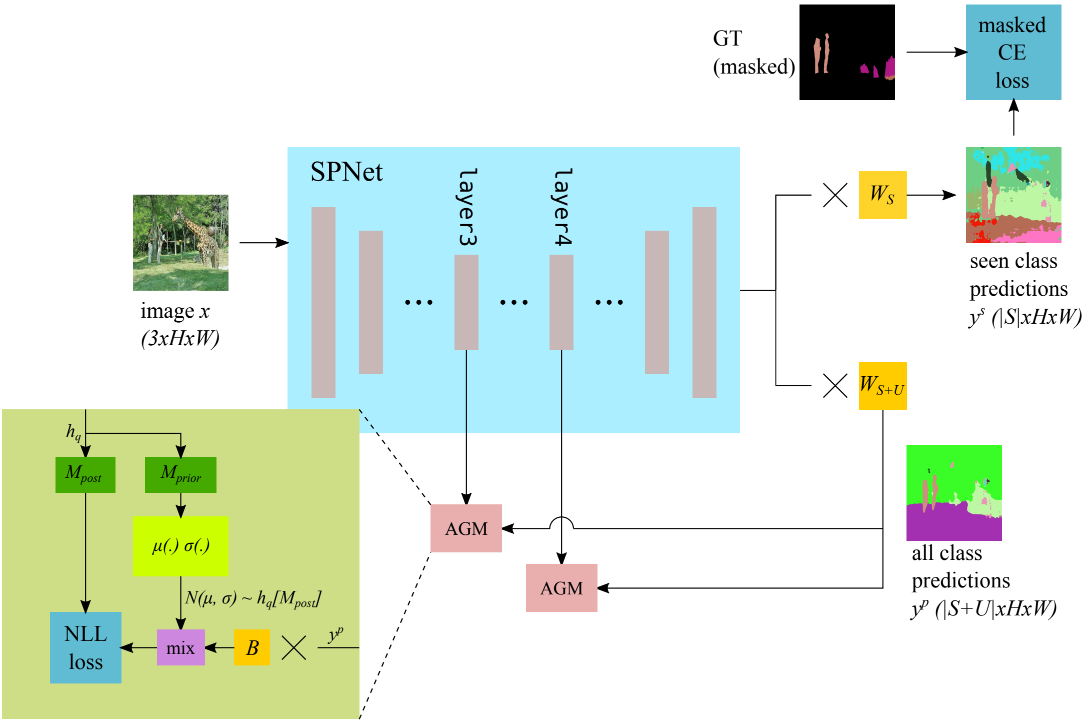

Figure 3.3: Instantiation of PGM with SPNet and the autoregressive gaussian models.

3.4 Instantiation of PGM with SPNet and autoregressive gaussian models

We instantiate the PGM architecture with the discriminative GZSSS network SPNet [14] and our simple custom generative model, autoregressive gaussian models (AGM). A summary of the instantiation can be viewed in Figure 3.3. In this instantiation, SPNet takes the role of N from Figure 3.2. It has two output branches with the word embedding matrices W_S, W_{S+U} for seen and seen+unseen classes respectively. The upper branch is a means of training SPNet in a supervised manner with the ground truth of the seen classes and substitutes training iteration a_N in Algorithm 1. The lower branch produces classification scores for all classes which are fed into the generative models as conditional input y^p. Finally, the AGM fit to the distribution of hidden layers as $G_q(h_q|y^p)$ from Figure 3.3.

The AGM formulation provides us a simple generative model that is easy to debug. Formally, let $h_q[M]$ indicate the vector obtained by eliminating channel indices of a hidden layer h_q except the ones specified in set M. An autoregressive gaussian model

$AGM(h_q)$ is defined as follows:

$$AGM(h_q) = \mathcal{N}(\mu(h_q[M_{prior}]), \sigma(h_q[M_{prior}])) \sim h_q[M_{post}] \tag{3.5}$$

with randomly chosen fixed indices $M_{prior} \cap M_{post} = \emptyset$ and $\mu(.)$, $\sigma(.)$ arbitrary neural networks with output dimension size $|M_{post}|$.

We employ a mixture of AGMs with random M_{prior}, M_{post} values to create the conditional model $G_q(h_q|y^p)$. The mixture model is defined as:

$$G_q(h_q|y^p) = [AGM_0(h_q),\ AGM_1(h_q) \cdots AGM_{n_{agm}}(h_q)]By^p \tag{3.6}$$

with B as a learnable weight matrix $B \in \mathbb{R}^{n_{agm} \times (|U|+|S|)}$ softmax normalized at its columns. We substitute class inference outputs of SPNet containing both unseen and seen class classification scores (without the ad-hoc calibration factor γ) for y^p. Since y^p is also softmax normalized in the architecture of SPNet in Equation 3.2, $G_q(h_q|y^p)$ is a proper mixture model.

For training loss L, we select the negative log-likelihood (NLL) of the feature layers:

$$L = -\lambda \log(p_{G_q}(h_q|y^p)) \tag{3.7}$$

where λ is a loss multiplier hyper-parameter. Figure 3.3 also visualizes the internals of the AGM modules. Here, h_q is split into two branches with channel masks M_{post} and M_{prior}. The branch on the right estimates a collection of gaussian distributions for the branch on the left. This collection then forms a mixture distribution with weights By^p.

We choose this custom model for two reasons. First, more complicated approaches such as GANs and normalizing flow networks [63] fail to improve the classification quality, since these advanced models can lower the loss L with respect to the complex distribution $H_q|y^p$ without perturbing y^p. This amounts to poor feedback to the classification backbone in terms of $\frac{\delta L}{\delta y^p}\frac{\delta y^p}{\delta \theta}$. In contrast, AGM is a simple mixture model with sufficient coupling capability for its channels with the autoregressive masks. Second, AGM is easier to inspect with the learnable weight matrix B, offering intuition regarding the complexity of the distributions for each class and the bias towards seen classes. We discuss the state of B after training in Chapter 4.

CHAPTER 4

EXPERIMENTS

In this chapter, we present the experimental results of our implementation. We first discuss the evaluation setup and define metrics specific to zero-shot segmentation. Then we provide the details of the baseline zero-shot segmentation network we employ in our experiments and elaborate our hyper-parameter selection process. Lastly, we present our qualitative and quantitative results and compare it to the baseline.

4.1 Evaluation setup and metrics

We choose the COCO-Stuff segmentation dataset [10] to assess the performance of our approach. Table 4.1 shows the image and class splits for validation and test sets. We pick the same data and class splits without disturbing the evaluation protocol in [14, 17, 16], however we utilize all of the images in the dataset during training as our approach can function in a transductive setting. We only retrieve ground truth labels if they belong to a seen class and the corresponding image is in the training split.

Table 4.1: Evaluation splits on the COCO-Stuff dataset.

Setting	# Seen Classes	# Unseen Classes	# Training Images	# Test Images
Validation	155	12	116282	2000
Test	155+12	15	116282+2000	5000

Seen class pixels dominate in terms of number with respect to unseen classes in GZSSS test splits. Following [14], we also employ mean harmonic intersection-over-union (IoU)

$$\text{Harmonic IoU} = \frac{2 \cdot \text{Seen IoU} \cdot \text{Unseen IoU}}{\text{Seen IoU} + \text{Unseen IoU}} \tag{4.1}$$

and mean harmonic accuracy in our evaluations:

$$\text{Harmonic Acc.} = \frac{2 \cdot \text{Seen Acc.} \cdot \text{Unseen Acc.}}{\text{Seen Acc.} + \text{Unseen Acc.}} \tag{4.2}$$

The harmonic evaluation metrics in equations 4.1 and 4.2 provide a clearer summary in terms of performance given that 15 unseen classes in the test setting account for a significantly smaller part of COCO-Stuff.

4.2 Baselines

To the best of our knowledge, only SPNet [14] as a discriminative GZSSS model reports results for COCO-Stuff. Our reproduction compared with the original work can be viewed in Table 4.2. Our only modification to match the numbers in the original results is to alter the number of training iterations, all of the other hyperparameters are the same as listed in [14].

Table 4.2: Reproduction results for SPNet.

	Iterations	γ	Seen IoU	Unseen IoU	Harmonic IoU
Reproduction	20000	0.4	34.66	8.12	13.16
Original	100000	0.4	34.52	8.33	13.42

Since the pre-trained architecture is not publicly available, we compare our results with the reproduced model. The comparison of our approach with the unmodified SPNet architecture might seem unfair as the additional AGM components increase the total number of parameters. However, the models incorporate the same architecture and the learning capacity during inference as these components are removed after training, making this comparison sound.

4.3 Implementation details and hyper-parameter selection

The models are implemented with PyTorch [64]. We utilize our reproduction of SPNet as the backbone GZSSS network N. The backbone is trained over a Deeplabv2 [26] architecture initialized with the parameters of ResNet-101 [65] pre-trained with ImageNet [3]. For the fixed word embedding matrix of the classes, we use the same word embeddings as [14]. During the co-training of SPNet and the generative models, the gradient-descent algorithm of SPNet and the algorithm parameters are kept the same as in [14] except the learning rate lr_{SPNet}.

For each AGM component in the generative models, we employ a shared triple stack of 1x1 convolution layers (so that the distributions are spatially independent) interleaved with ReLU function as $\mu(\cdot)$. For $\sigma(\cdot)$, we employ a fixed diagonal matrix with the same value σ_0 for every element. $M_{prior},\ M_{post}$ are randomly initialized and fixed at the beginning of the training, however $|M_{prior}|,\ |M_{post}|$ are equal across the components attached to the same layer.

We alternatively attach generative models $G_{lyr3},\ G_{lyr4}$ with fixed number of AGM components n_{AGM} to `layer3`, `layer4` in Deeplabv2 architecture during the hyper-parameter selection process. We found later layers in the encoder part of the Deeplabv2 architecture to be incompatible with the AGM model due to the large channel-wise dimensionality of these layers. The AGM mixture model fails to capture the coupling between channels with its limited learning capacity as the number of channels grow. In addition to the later layers, layers stacked below `layer3`, `layer4` also failed in our preliminary experiments, since these layers do not accumulate enough visual/spatial information to form a healthy distribution conditioned on classification scores.

Owing to the convolutional layers at the encoder of Deeplabv2, a hidden layer h_q and classification scores y^p might spatially mismatch. In this case, y^p is expanded to the size of h_q with bilinear interpolation to preserve normalized class scores.

For gradient-descent, all of the parameters outside of the SPNet architecture are trained with Adam algorithm [66] with $\beta_1 = 0.9,\ \beta_2 = 0.999$ and learning rate lr_{PGM}. We set λ to 1.0 during the pre-training of generative networks for 15000 iter-

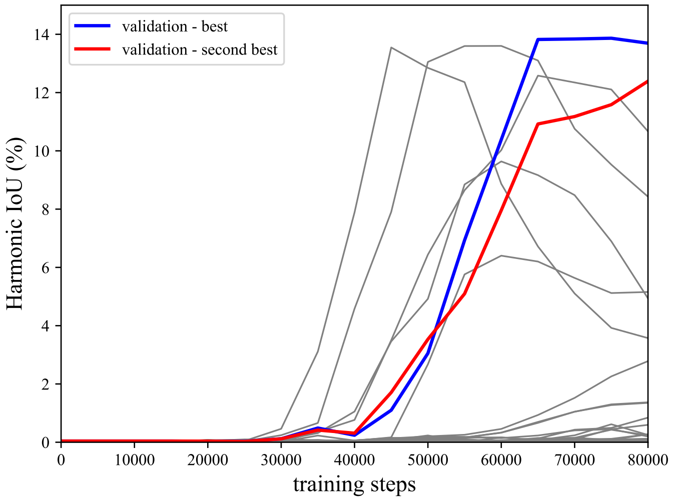

Figure 4.1: Harmonic IoU scores in cross-validation set during training.

ations. After co-training with SPNet begins, λ is incremented from 0 to λ_0 with linear warmup for 5000 iterations. The training terminates at iteration 80000. We use the same data augmentation and sampling protocol as [14] during the training of PGM with each iteration consuming a batch of 10 samples.

Classification score outputs of training and test split samples differ greatly in terms of hardness since SPNet is only exposed to training samples during training. Owing to this issue, only the y^p of test split samples are employed in training PGM.

Hyper-parameters lr_{SPNET}, lr_{PGM}, $n_{AGM}, \sigma_0, \lambda_0$ are chosen over a grid via the bayesian search algorithm of Weights and Biases platform [67] with Harmonic IoU as the objective metric. Table 4.3 lists the validation set runs (layers to attach generative networks are specified with tuples (`[layer_name`, $|M_{prior}|$, $|M_{post}|$]).

Figure 4.1 displays the Harmonic IoU values as the training progresses for the validation runs. In almost all of the relatively successful models the performance suffers after a certain number of iterations. We believe this happens when our assumption of gradients being negligible at layers h_q becomes invalid despite the `detach` process due to too many gradient-descent steps.

The GZSSS task poses a special problem with the results from validation settings

Table 4.3: Validation runs for hyper-parameter selection.

lr_{SPNet}	lr_{PGM}	AGM Details	n_{AGM}	σ_0	λ_0	Harmonic IoU (%)
2.5e-04	2.5e-05	`[[layer4,128,256]]`	21	1.0	0.1	0.01
1.0e-04	2.5e-05	`[[layer3,64,128],[layer4,128,256]]`	7	0.5	0.2	0.03
1.0e-04	2.5e-05	`[[layer4,128,256]]`	14	1.0	0.2	0.03
2.5e-04	2.5e-05	`[[layer3,64,128]]`	7	1.0	0.2	0.06
2.5e-04	2.5e-05	`[[layer3,64,128],[layer4,128,256]]`	7	0.5	0.2	0.11
1.0e-04	5.0e-05	`[[layer3,128,256],[layer4,128,256]]`	21	2.0	0.2	0.12
2.5e-04	1.0e-05	`[[layer3,128,256],[layer4,128,256]]`	21	0.5	0.4	0.20
2.5e-04	5.0e-05	`[[layer3,128,256],[layer4,128,256]]`	21	0.5	0.1	0.23
5.0e-04	2.5e-05	`[[layer3,64,128]]`	7	2.0	0.4	0.25
5.0e-04	2.5e-05	`[[layer3,64,128],[layer4,256,512]]`	7	1.0	0.1	0.27
5.0e-04	2.5e-05	`[[layer3,64,128],[layer4,128,256]]`	21	2.0	0.1	0.27
1.0e-04	5.0e-05	`[[layer3,64,128],[layer4,128,256]]`	7	0.5	0.4	0.60
5.0e-04	5.0e-05	`[[layer3,64,128],[layer4,256,512]]`	7	1.0	0.1	0.84
2.5e-04	5.0e-05	`[[layer3,64,128]]`	21	0.5	0.4	1.35
2.5e-04	5.0e-05	`[[layer3,64,128]]`	21	0.5	0.4	1.37
1.0e-04	5.0e-05	`[[layer3,128,256],[layer4,256,512]]`	21	0.5	0.4	2.78
5.0e-04	5.0e-05	`[[layer3,128,256],[layer4,256,512]]`	21	1.0	0.4	3.57
2.5e-04	5.0e-05	`[[layer3,128,256],[layer4,256,512]]`	7	2.0	0.4	4.94
5.0e-04	2.5e-05	`[[layer4,128,256]]`	7	2.0	0.4	5.16
5.0e-04	5.0e-05	`[[layer4,256,512]]`	21	2.0	0.4	8.43
2.5e-04	5.0e-05	`[[layer3,128,256],[layer4,256,512]]`	21	0.5	0.4	10.67
5.0e-04	5.0e-05	`[[layer3,128,256],[layer4,256,512]]`	21	0.5	0.2	12.39
5.0e-04	2.5e-05	`[[layer3,128,256],[layer4,256,512]]`	21	0.5	0.4	13.69

translating to test settings as the addition of new unseen classes cannot be explained away with simply sampling different examples from a similar distribution of images. Combined with the overfitting problem, this results in the visual defects shown in Figure 4.2 (unseen classes are marked with "*" in the legend, ground truth and unseen classes are shown in different rows for ease of viewing) when the same hyper-parameters with the best validation run (marked blue in Table 4.3 and Figure 4.1) are used for the model trained in the test setting. Due to this issue, we use the second best hyper-parameter selection (marked red) as the performance seems to deteriorate the latest. We halve lr_{SPNet}, lr_{PGM} to compensate for the higher ratio of unseen class pixels in the test setting.

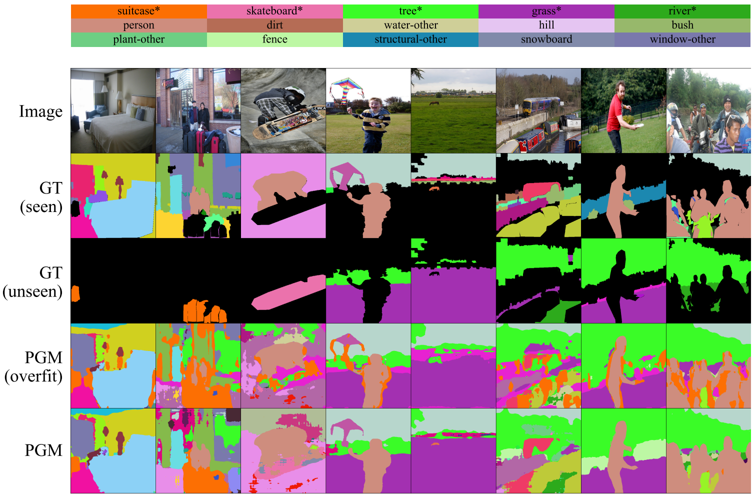

Figure 4.2: Overfitting problems in PGM.

4.4 Quantitative results

Table 4.4 lists the comparison of our final model with SPNet [14] in the test setting of COCO-Stuff. Although SPNet without the ad-hoc calibration factor dominates in terms of seen class accuracy, it fails significantly in harmonic scores and scores regarding the unseen classes. Our approach surpasses SPNet with ad-hoc calibration factor in unseen class and harmonic IoU scores and is competitive with SPNet in seen class IoU. It also surpasses SPNet in seen class accuracy, however it falls short in other accuracy scores. We believe the discrepancy between unseen/harmonic IoU and accuracy scores in both models is due to PGM slightly favoring some of the unseen classes in overfitting scenarios for the seams and masked regions, whereas SPNet classifies these regions as seen classes since the classification confidence is low in these areas. We show further evidence of this phenomenon in Section 4.5.

4.5 Qualitative results

4.5.1 Promising cases

Figure 4.3 displays the promising cases compared to SPNet. In all of the images, PGM successfully classifies most of the unseen classes. The segmentation outputs

Table 4.4: PGM compared to SPNet on COCO-Stuff.

	Unseen IoU (%)	Seen IoU (%)	Harmonic IoU (%)	Unseen Acc. (%)	Seen Acc. (%)	Harmonic Acc. (%)
SPNet - $\gamma = 0$	0.64	34.14	1.26	0.64	**50.02**	1.28
SPNet	8.12	**34.66**	13.15	**16.95**	45.62	**24.72**
PGM	**8.88**	34.53	**14.06**	13.02	48.34	20.52

of SPNet in the images first, second and third from the right shows the problem that arises with the ad-hoc calibration factor: only the seam areas of unseen parts are correctly classified as these parts have low confidence in seen class classification scores. This clearly shows that ad-hoc modifications fail to mitigate the seen class bias in SPNet. In contrast, our approach does not exhibit this issue.

4.5.2 Visualization of PGM parameters

In Figure 4.5.2 we visualize the weight matrices B of the mixture models G_{lyr3}, G_{lyr4} for a deeper intuition on the hidden layer distributions conditioned on y^p. Each horizontal strip in the heatmaps represent an AGM component in the mixture model. Most of the 182 classes seem to be evenly represented by each component given that the weights have a more-or-less uniform distribution. Note that there are no visible difference in component weights between unseen and seen classes. This is another observation that supports our claim of PGM not displaying bias towards seen classes.

Some of the classes in COCO-Stuff have smooth textures in the images (*metal*, *wall-tile*) and are relatively unimodal in terms of their hidden layer distributions. This is also reflected in the weight matrices in terms of component weights. For the most of the other classes, the weight distribution is close to uniform, however for the classes marked with their label names in Figure 4.5.2, the weights are focused on one or two AGM components. Since the hidden layer distributions of these classes are relatively simple, the AGM mixture model is able approximate them with few number of components.

As further evidence for this phenomenon, two heatmaps at the bottom-left of Figure 4.5.2 show the entropy of the categorical distribution induced by By^p for a sample image (normalized within the image spatially, brighter parts have high entropy). The

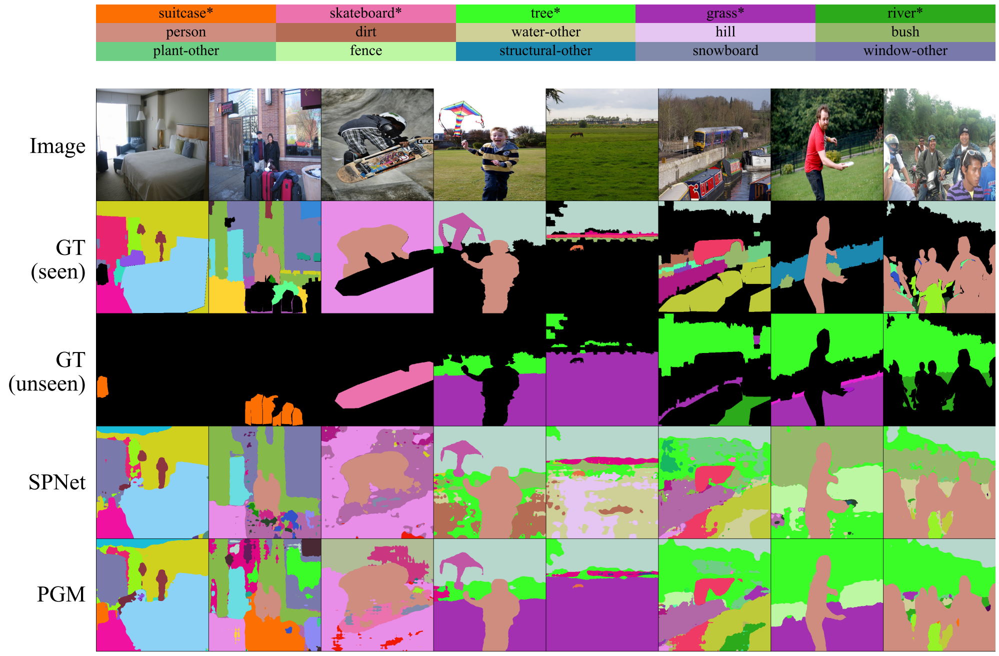

Figure 4.3: Promising outputs of PGM compared to SPNet in selected test images.

snow, which has a relatively smooth texture and color has the lowest entropy, implying that not many AGM components contribute to the distribution, whereas *person* has the highest entropy within the image, as people have complex shapes, textures and colors in the dataset. This outlook also implies that our model is able to somehow capture the conditional multi-modal nature of the hidden layer distributions.

4.5.3 Failure cases

Figure 4.5 lists some of the failure cases and the estimated pixelwise probabilities of the hidden layers by G_{lyr3} and G_{lyr4} of PGM (normalized within the image, low probability regions are brighter). Images except the first image from the right seem to be classified in mostly correct shapes but incorrect categories. Some of these categories are semantically very close or hierarchical, for example in the first image from the left the wall is classified as *wall-other* although the correct label is *wall-concrete*. Similarly in the fourth image from the left, the *road* is classified as *pavement*.

Some classification discrepancies may be due to human classification errors and other

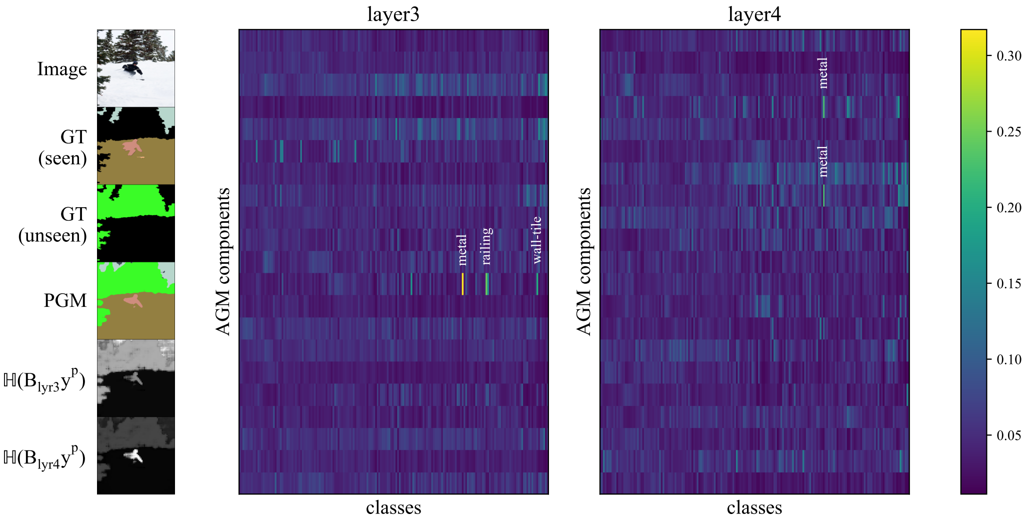

Figure 4.4: AGM mixture model weight matrices and mixture weight entropies.

classification ambiguity in COCO-Stuff. In the third image from right, the ground-truth label for the blue region of the image is *clouds* but there are no actual clouds in the image, in the fourth image from right, there is a fence in front of the trees but no part of the image is classified as fence in the ground truth.

Low estimated probabilities by generative models suggest that some of the classification errors are recoverable with further hyper-parameter engineering. This is prominent in small objects or in cases where part of the object is incorrectly classified as a class that is semantically distant from the ground truth. Part of the cow classified as *dog* in the second image from the left, the skateboard in the third image from the left, the flock of birds in the second image from the right have low estimated probabilities.

Lastly, it is difficult for the model to correctly segment shapes with unusual contexts. The fourth image from the left has an unusual arrangement of shoes the model misclassifies as *people*, most likely heavily influenced by the assumption that people are generally on top of the skateboards in the images. The first image from the right exhibits a similar problem.

Most of the failed classifications coincide with the baseline SPNet. This suggests that they are mostly inherited from the pre-trained SPNet model before the co-training stage with AGM. Still, the PGM model performs better with respect to the baseline

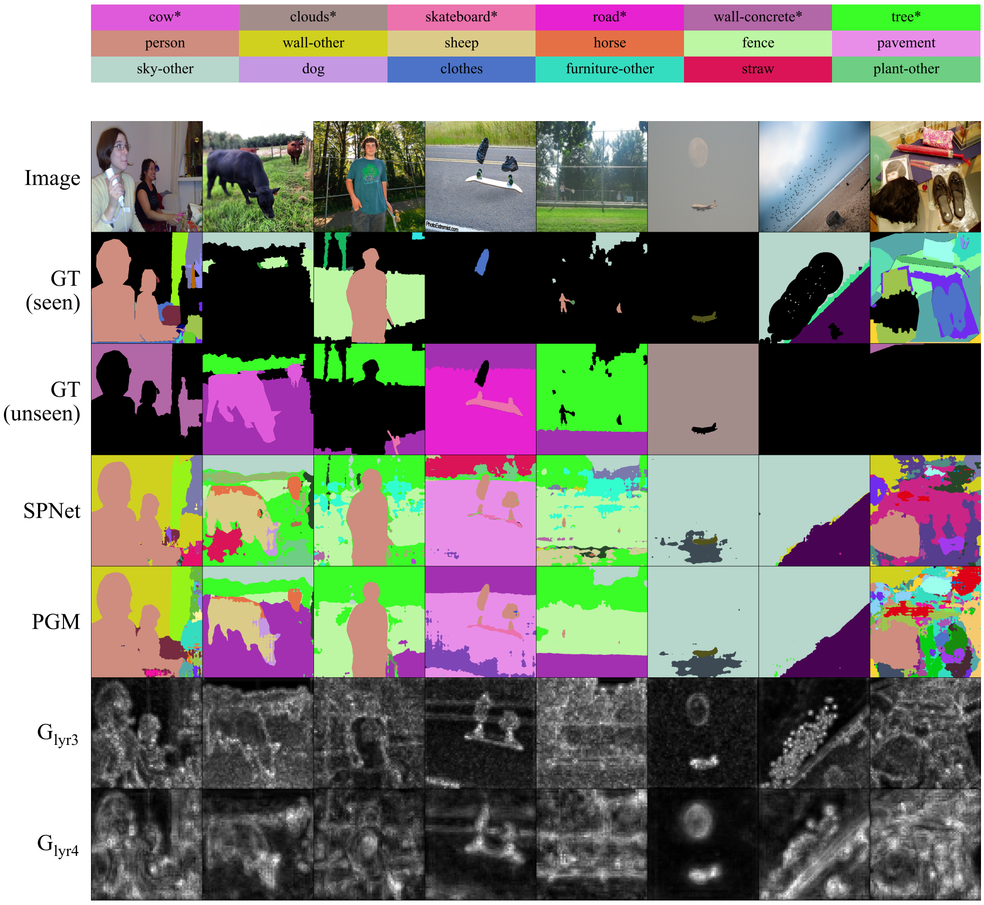

Figure 4.5: Failure cases.

in some of them: in the second, third, fourth and fifth images from the left, *grass* is misclassified as *straw*, *tree*, *plant-other* and *water-other* with the baseline, whereas it is mostly correctly classified with PGM. The baseline displays the setbacks of the ad-hoc calibration factor γ here as well in the fourth image from left, in which only the fringe regions of the *grass* segment is classified as unseen classes (*tree* and *plant-other*), which are still incorrect in terms of category.

CHAPTER 5

CONCLUSION

In this thesis we explored a principled way of reducing the classification bias towards seen classes in GZSSS networks. The experiments we conducted show that attaching parallel generative networks to GZSSS models leads to competitive performance with ad-hoc approaches. The parallel generative networks also enable the model to train in a semi-supervised setting.

In future work, we plan to adopt recent ideas for GZSSS problem in our approach. Pastore et al. [17] filter out pseudo-labels with consistency constraints on data augmentations. Gu et al. [16] condition synthetic unseen class feature generator with contextual information. These methods are readily applicable to our PGM approach in terms of consistency-constraints on the default conditional input of the PGMs and additional conditional inputs to PGM as context.

Our model is the first semi-supervised GZSSS approach to the best of our knowledge. We were only able to showcase this feature in a transductive scenario with COCO-stuff due to time and data constraints. We plan to demonstrate further benefits of semi-supervised nature of our approach with tests on out-of-domain scenarios in future studies.

REFERENCES

[1] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach. (Second edition)*. Prentice Hall, Nov. 2011.

[2] Y. Hirano, C. Garcia, R. Sukthankar, and A. Hoogs, "Industry and Object Recognition: Applications, Applied Research and Challenges," in *Toward Category-Level Object Recognition* (J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, eds.), Lecture Notes in Computer Science, pp. 49–64, Berlin, Heidelberg: Springer, 2006.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, Dec. 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[5] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11557–11568, June 2021.

[6] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, Jan. 2015.

[7] I. Alonso, L. Riazuelo, and A. C. Murillo, "Enhancing V-SLAM Keyframe Selection with an Efficient ConvNet for Semantic Analysis," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4717–4723, May 2019. ISSN: 2577-087X.

[8] A. Ouahabi and A. Taleb-Ahmed, “Deep learning for real-time semantic segmentation: Application in ultrasound imaging,” *Pattern Recognition Letters*, vol. 144, pp. 27–34, 2021.

[9] X. Guo, Z. Wang, Q. Yang, W. Lv, X. Liu, Q. Wu, and J. Huang, “GAN-Based virtual-to-real image translation for urban scene semantic segmentation,” *Neurocomputing*, vol. 394, pp. 127–135, June 2020.

[10] H. Caesar, J. Uijlings, and V. Ferrari, “Coco-stuff: Thing and stuff classes in context,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1209–1218, 2018.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), Lecture Notes in Computer Science, (Cham), pp. 740–755, Springer International Publishing, 2014.

[12] M. Zhang, Y. Zhou, J. Zhao, Y. Man, B. Liu, and R. Yao, “A survey of semi- and weakly supervised semantic segmentation of images,” *Artificial Intelligence Review*, vol. 53, pp. 4259–4288, Aug. 2020.

[13] M. Bucher, T.-H. VU, M. Cord, and P. Pérez, “Zero-Shot Semantic Segmentation,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 468–479, Curran Associates, Inc., 2019.

[14] 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)*, pp. 8248–8257, June 2019. ISSN: 2575-7075.

[15] N. Kato, T. Yamasaki, and K. Aizawa, “Zero-Shot Semantic Segmentation via Variational Mapping,” in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1363–1370, Oct. 2019. ISSN: 2473-9944.

[16] Z. Gu, S. Zhou, L. Niu, Z. Zhao, and L. Zhang, “Context-aware Feature Generation For Zero-shot Semantic Segmentation,” in *Proceedings of the 28th ACM*

International Conference on Multimedia, MM '20, (New York, NY, USA), pp. 1921–1929, Association for Computing Machinery, Oct. 2020.

[17] G. Pastore, F. Cermelli, Y. Xian, M. Mancini, Z. Akata, and B. Caputo, "A Closer Look at Self-training for Zero-Label Semantic Segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2687–2696, June 2021. ISSN: 2160-7516.

[18] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008. ISSN: 1063-6919.

[19] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, Aug. 1995.

[20] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," in *BMVC*, 2008.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, June 2005. ISSN: 1063-6919.

[22] N. Plath, M. Toussaint, and S. Nakajima, "Multi-class image segmentation using conditional random fields and global classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, (New York, NY, USA), pp. 817–824, Association for Computing Machinery, June 2009.

[23] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov. 2004.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, Sept. 1995.

[25] L. Ladický, C. Russell, P. Kohli, and P. H. Torr, "Associative hierarchical CRFs for object class image segmentation," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 739–746, Sept. 2009. ISSN: 2380-7504.

[26] L. 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,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, Apr. 2018.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, June 2014. ISSN: 1063-6919.

[28] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Advances in Neural Information Processing Systems* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., 2015.

[29] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, July 2017. ISSN: 1063-6919.

[31] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, “Feedforward semantic segmentation with zoom-out features,” pp. 3376–3385, IEEE Computer Society, June 2015. ISSN: 1063-6919.

[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[33] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,” in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), Lecture Notes in Computer Science, (Cham), pp. 833–851, Springer International Publishing, 2018.

[34] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, “Gated-scnn: Gated shape cnns for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[35] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), Lecture Notes in Computer Science, (Cham), pp. 234–241, Springer International Publishing, 2015.

[36] V. Cheplygina, M. de Bruijne, and J. P. W. Pluim, “Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis,” *Medical Image Analysis*, vol. 54, pp. 280–296, May 2019.

[37] W. Bai, O. Oktay, M. Sinclair, H. Suzuki, M. Rajchl, G. Tarroni, B. Glocker, A. King, P. Matthews, and D. Rueckert, “Semi-supervised learning for network-based cardiac MR image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2017*, Springer Verlag, Sept. 2017. ISSN: 0302-9743.

[38] X. Li, L. Yu, H. Chen, C.-W. Fu, L. Xing, and P.-A. Heng, “Transformation-Consistent Self-Ensembling Model for Semisupervised Medical Image Segmentation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 523–534, Feb. 2021.

[39] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[40] S. Sedai, D. Mahapatra, S. Hewavitharanage, S. Maetschke, and R. Garnavi, “Semi-supervised Segmentation of Optic Cup in Retinal Fundus Images Using Variational Autoencoder,” in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2017* (M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne, eds.), (Cham), pp. 75–82, Springer International Publishing, 2017.

[41] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3581–3589, 2014.

[42] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, “Semantic Segmentation using Adversarial Networks,” in *NIPS Workshop on Adversarial Training*, (Barcelona, Spain), Dec. 2016.

[43] N. Souly, C. Spampinato, and M. Shah, “Semi Supervised Semantic Segmentation Using Generative Adversarial Network,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5689–5697, Oct. 2017. ISSN: 2380-7504.

[44] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, pp. 2672–2680, 2014.

[45] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, “Adversarial Learning for Semi-Supervised Semantic Segmentation,” *arXiv:1802.07934 [cs]*, July 2018. arXiv: 1802.07934.

[46] S. Mittal, M. Tatarchenko, and T. Brox, “Semi-Supervised Semantic Segmentation With High- and Low-Level Consistency,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 1369–1379, Apr. 2021.

[47] Z. Ke, D. Qiu, K. Li, Q. Yan, and R. W. H. Lau, “Guided Collaborative Training for Pixel-Wise Semi-Supervised Learning,” in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), Lecture Notes in Computer Science, (Cham), pp. 429–445, Springer International Publishing, 2020.

[48] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler, “Datasetgan: Efficient labeled data factory with minimal human effort,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10145–10155, June 2021.

[49] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, June 2020. ISSN: 2575-7075.

[50] D. Li, J. Yang, K. Kreis, A. Torralba, and S. Fidler, "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8300–8311, June 2021.

[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," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2251–2265, Sept. 2019.

[52] 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 *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc., 2013.

[53] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-Embedding for Image Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 1425–1438, July 2016.

[54] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2927–2936, June 2015. ISSN: 1063-6919.

[55] E. Kodirov, T. Xiang, and S. Gong, "Semantic Autoencoder for Zero-Shot Learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4447–4456, July 2017. ISSN: 1063-6919.

[56] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature Generating Networks for Zero-Shot Learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5542–5551, June 2018. ISSN: 2575-7075.

[57] V. K. Verma, G. Arora, A. Mishra, and P. Rai, “Generalized Zero-Shot Learning via Synthesized Examples,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4281–4289, June 2018. ISSN: 2575-7075.

[58] F. Jurie, M. Bucher, and S. Herbin, “Generating Visual Representations for Zero-Shot Classification,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2666–2673, Oct. 2017. ISSN: 2473-9944.

[59] Y. Li, K. Swersky, and R. Zemel, “Generative Moment Matching Networks,” in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1718–1727, PMLR, June 2015. ISSN: 1938-7228.

[60] M. B. Sariyildiz and R. G. Cinbis, “Gradient Matching Generative Networks for Zero-Shot Learning,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2163–2173, June 2019. ISSN: 2575-7075.

[61] S. Naha and Y. Wang, “Object figure-ground segmentation using zero-shot learning,” in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2842–2847, Dec. 2016.

[62] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR)*, 2017.

[63] I. Kobyzev, S. J. Prince, and M. A. Brubaker, “Normalizing flows: An introduction and review of current methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3964–3979, 2021.

[64] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[65] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recog-

nition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV, USA), pp. 770–778, IEEE, June 2016.

[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[67] L. Biewald, "Experiment tracking with weights and biases," 2020. Software available from wandb.com.