Feature Generation and Augmentation for Few-Shot Learning

Thesis Proposal

Jingyi Xu

Advisor: Professor Dimitris Samaras

May 2023

Abstract

Machine learning has been highly successful in data-intensive applications, but is often hampered when the data set is small. Recently, few-shot learning (FSL) has been proposed to tackle this problem. It can be of great use in the scenario where data with supervised information are hard or impossible to acquire.

FSL aims to develop methods that can rapidly generalize to new tasks using very few (in extreme cases one or even zero) samples with labels. A common way to alleviate this data insufficiency issue in few-shot learning is data augmentation. In this thesis proposal, we explore the use of feature augmentation techniques on FSL problems. We aim to use generative models to model the feature distribution via a continuous space from which we can sample new data for augmentation. We apply the use of this feature generation framework on three different tasks: few-shot image classification, fine-grained few-shot classification and few-shot object detection (FSOD). Although each of these tasks poses different sets of challenges, we demonstrate that they can be resolved within our proposed feature generation framework.

Specifically, for the task of few-shot image classification, we focus on generating representative samples that can reflect the key characteristics of the corresponding category. We propose a sample selection method to collect representative samples and use them to train a generative model. Next, for fine-grained few-shot classification, we use a variational autoencoder (VAE) to model the intra-class variance via a common distribution, from which we can sample multiple feature instances to diversify few-shot training samples. Finally, for few-shot object detection, we propose a novel VAE based architecture to generate samples with increased crop-related diversity. We show the generated features significantly improve the current state-of-the-art FSOD performance.

To conclude this thesis, we propose to use feature generation to perform the task of class-agnostic object counting (CAC). Existing CAC methods require users to annotate a few boxes of the counting objects. We instead propose using generated features as object prototypes for counting. This will eliminate the need of human-annotated inputs and enable many real-world applications.

Contents

Chapter 1

Introduction

Deep learning has achieved significant success in numerous computer vision tasks with sufficiently large-scale labeled training data. However, in many real-world scenarios, annotated data can be hard and costly to obtain, such as rare medical conditions, rare animal species, satellite images or failure cases in autonomous driving systems. Hence, much attention has been recently paid to the development of few-shot learning [96, 116, 132], which aims to design models that can learn to solve tasks using just a few or even zero labeled examples.

Data augmentation is recognized as a straightforward way to address the data insufficiency issue in few-shot learning by increasing sample richness. According to the dimension of information, data augmentation methods can be categorized into two types, *i.e.*, data-level augmentation and feature-level augmentation. Data-level augmentation mainly focuses on increasing the number of samples by means of pixel transformations and pixel generation [17, 114]. Chen *et al* [17] propose an image deformation framework that learns to synthesize diverse deformed images to augment the one-shot training images. Tsutsui *et al* [114] propose a meta-learning framework to combine Generative Adversarial Networks (GAN) generated images with original images, so that the resulting combined images can improve one-shot learning.

Although data-level augmentation methods have made promising progress, they often require large-scale datasets as support, which is not easy to achieve in the FSL settings. In contrast, feature-level augmentation maps pixel information into a high-dimensional latent space, which carries more valid information than mere original pixels. By modeling the

valid information in a compressed manner, it is generally more effective than data-level augmentation [62, 109].

In this thesis proposal, we explore feature-level augmentation solutions to FSL problems. We aim to use generative models to model the feature distribution via a continuous space from which we can sample new data for augmentation. We apply the use of this feature generation framework on three few-shot learning tasks: few-shot image classification, fine-grained few-shot classification and few-shot object detection. We show that our generated features can effectively complement the original few-shot samples and significantly improve the model performance for all the three tasks, achieving state-of-the-art performance on the benchmark for each task.

In Chapter 3, we present our work on fine-grained few-shot recognition. We propose a feature disentanglement framework that allows us to generate features with enlarged intra-class variations while preserving the class-discriminative features. Specifically, we disentangle a feature representation into two components: one represents the intra-class variance and the other encodes the class-discriminative information. We assume that the intra-class variance induced by variations in poses, backgrounds, or illumination conditions is shared across all classes and can be modeled via a common distribution. We sample features repeatedly from the learned intra-class variance distribution and add them to the class-discriminative features to get the augmented features. Such a data augmentation scheme ensures that the augmented features inherit crucial class-discriminative features while exhibiting large intra-class variance. We show that our method significantly outperforms the state-of-the-art methods on multiple challenging fine-grained few-shot image classification benchmarks.

In Chapter 4, we present our work on few-shot classification. The feature representations from few-shot classes are often biased due to data scarcity. To mitigate this issue, we propose to generate visual samples based on semantic embeddings using a conditional variational autoencoder (CVAE) model. We train this CVAE model on base classes and use it to generate features for novel classes. More importantly, we guide this VAE to strictly generate representative samples by removing non-representative samples from the base training set when training the CVAE model. We show that this training scheme enhances the representativeness of the generated samples and therefore, improves the few-shot clas-

sification results. Experimental results show that our method improves three few-shot learning baseline methods by substantial margins, achieving state-of-the-art few-shot classification performance on *mini*ImageNet and *tiered*ImageNet datasets for both 1-shot and 5-shot settings.

In Chapter 5, we present our work on few-shot object detection. We observe that in object detection, the object proposals generated by detectors often do not contain the objects perfectly but overlap with them in many possible ways, exhibiting great variability in the difficulty levels of the proposals. Training a robust classifier against this crop-related variation requires abundant training data, which is not available in few-shot settings. To mitigate this issue, we propose a novel variational autoencoder (VAE) based data generation model, which is capable of generating data with increased crop-related diversity. The main idea is to transform the latent space such latent codes with different norms represent different crop-related variations. This allows us to generate features with increased crop-related diversity in difficulty levels by simply varying the latent norm. In particular, each latent code is rescaled such that its norm linearly correlates with the Intersection-Over-Union (IoU) score of the input crop *w.r.t.* the ground-truth box. Here the IoU score is a proxy that represents the difficulty level of the crop. We train this VAE model on base classes conditioned on the semantic code of each class and then use the trained model to generate features for novel classes. In our experiments, our generated features consistently improve state-of-the-art few-shot object detection methods on the PASCAL VOC and MS COCO datasets.

In Chapter 6, we conclude this thesis proposal and discuss the future work. We focus on applying feature generation in class-agnostic object counting. To relax the dependency of labeled samples, we propose the task of zero-shot object counting (ZSC), in which the model only needs the class name to count the number of object instances in the image. Our current strategy is to generate a class prototype via a generative model and use it as the matching template. One possible direction to improve this method is to apply different templates dynamically according to the input image, which we leave as future work.

1.1 Contributions

To summarize, the contributions of this thesis proposal are as follows:

- We propose a data augmentation method via feature disentanglement to address the data scarcity problem in few-shot fine-grained classification. The generated features can enlarge the intra-class variance for novel set images while preserving the class-discriminative features.
- We propose a novel sample selection method to collect representative samples. We show that these representative samples can be used to train a VAE model to obtain reliable data points for constructing class-representative prototypes in FSL.
- We propose a novel VAE architecture that can effectively increase the crop-related diversity of the generated samples to support the training of few-shot object detection classifiers.

Chapter 2

Literature Review

2.1 Few-Shot Object Recognition

Few-shot object recognition aims to recognize an image of a novel class with very few labeled examples available. Few-shot methods can be broadly organized into three categories: metric learning based, optimization based and data augmentation based.

Metric learning based methods [107, 112, 113, 116, 134, 137, 141] utilize the similarities between images to regularize the embedding space. Matching Networks [116] use an attention mechanism over a learned embedding of the labeled set of examples to predict classes for the unlabeled points. The Prototypical Network [107] learns to classify query samples based on their Euclidean distance to prototype representations of each class. Sung *et al.* [112] propose to measure the distance metric with a CNN-based relation module. Ye *et al* [137] propose to use a transformer on top of the prototypical network embeddings to learn a better mapping for class representations. Oreshkin *et al* [85] proposed a task dependent adaptive metric learning (TADAM) for few-shot learning with distance metric scaling for the output features with a learnable temperature parameter and with task conditioning, where the parameters of the feature extractor are task-dependent.

Optimization based methods [31, 59, 65, 66, 93, 95, 99, 103] aim to design models that can generalize to new tasks efficiently. MAML [31] uses a meta-learner to find an initialization which can be adapted to new categories within few gradient updates using small training data. Meta-

SGD [65] learns to learn not only the learner initialization but also the learner update direction and learning rate. Lee *et al.* propose MetaOptNet [59], which uses discriminatively trained linear predictors as base learners to learn feature representations for FSL. Ravi and Larochelle [95] utilize a long short-term memory (LSTM) network to optimize the updates for the model parameters when training a network in a few-shot setting. An important issue of optimization based methods is how to avoid catastrophic forgetting in a dynamic setting, which means that information on the old tasks should not be forgotten [120].

Data augmentation based methods [3, 101, 121] generate additional training examples to alleviate the problem of data insufficiency. These methods can be categorized as two groups, *i.e.*, data-level augmentation and feature-level augmentation, according to the dimension of information. DAGAN [3] uses conditional generative adversarial network (GAN) to transform image features, which can be applied to novel unseen classes of data. Wang *et al.* [121] propose to combine a meta-learner with a hallucinator, which can effectively hallucinate novel instances of new classes. Some methods transfer intra-class variance from the base classes to the novel class. The Δ-encoder [101] extracts transferable intra-class deformations from image pairs of the same class and uses them to augment samples of the novel classes. Gao *et al* [32] explore the underlying distribution behind few-shot data and propose an adversarial covariance augmentation network to overcome the limitations of FSL. Chu *et al* [18] try to compute feature representations for each patch, rather than the entire image. Each small patch is connected by a recurrent neural network (RNN) and the features of the image are further fused. In FSL, feature-level augmentation is generally more effective than data-level augmentation by modeling the valid information in a compressed manner.

2.2 Few-Shot Object Detection

Few-shot object detection aims at detecting novel objects with only few annotated instances. A number of prior methods [28–30, 36, 37, 46, 46, 56, 64, 74, 76, 89, 110, 125–127, 146] have been proposed to address this challenging task. One line of work focuses on the **meta-learning** paradigm, which has been widely explored in few-shot classification [27, 45, 102, 122, 129, 133, 135, 136]. Meta-learning based ap-

proaches introduce a meta-learner to acquire meta-knowledge that can be then transferred to novel classes. [45] propose a meta feature learner and a reweighting module to fully exploit generalizable features from base classes and quickly adapt the prediction network to predict novel classes. [122] propose specialized meta-strategies to disentangle the learning of category-agnostic and category-specific components in a CNN based detection model. Meta R-CNN [133] extends Faster / Mask R-CNN by proposing meta-learning over RoI (Region-of-Interest) features.

Another line of work adopts a **two-stage fine-tuning** strategy and has shown great potential recently [11, 90, 110, 118, 128]. [118] propose to fine-tune only box classifier and box regressor with novel data while freezing the other paramters of the model. This simple stragetegy outperforms previous meta-learners. FSCE [110] leverages a contrastive proposal encoding loss to promote instance level intra-class compactness and inter-class variance. FADI [11] associates each novel category to one base category and then the network is trained to align the feature distribution of the novel category to the associated base category. Guirguis *et al* [33] propose a constraint-based finetuning approach (CFA) to alleviate catastrophic forgetting, while achieving competitive results without increasing the model capacity. Fan *et al* [30] propose a few-shot detector without forgetting, Retentive R-CNN, which can assist novel class adaptation with base class knowledge and ensemble base and novel class detectors.

Other **data augmentation** works try to increase the variance of the data for novel categories. Zhang *et al* [145] introduce a hallucinator network that learns to generate additional training examples for novel categories. The features in the RoI head of novel category samples are augmented by leveraging the shared within-class feature variation from base categories. Kaul *et al* [46] show in their experiments that data augmentation, *i.e.*, color jittering, random cropping, mosaicing, and dropout for the extracted features for each RoI, significantly improves the performance.

2.3 Variational Autoencoder

Different variational autoencoder (VAE) variants have been proposed to generate diverse data [35, 42, 52, 104]. β-VAE [42] imposes a heavy penalty on the KL divergence term to enhance the disentanglement of the latent dimensions. By traversing the values of latent variables, β-VAE can gen-

erate data with disentangled variations. ControlVAE [104] improves upon β-VAE by introducing a controller to automatically tune the hyperparameter added in the VAE objective. However, disentangled representation learning can not capture the desired properties without supervision. Some VAE methods allow explicitly controllable feature generation including CSVAE [52] and PCVAE [35]. CSVAE [52] learns latent dimensions associated with binary properties. The learned latent subspace can easily be inspected and independently manipulated. PCVAE [35] uses a Bayesian model to inductively bias the latent representation. Thus, moving along the learned latent dimensions can control specific properties of the generated data.

Using a conditional VAE to model a feature distribution has also been used before in many computer vision tasks such as image classification [49, 100, 132, 142], image generation [25, 71], image restoration [23], or video processing [87]. Using VAE models for generating features conditioned on the corresponding semantic embedding is fairly common in zero-shot learning (ZSL) methods[5, 34, 83, 100, 139, 143]. Mishra *et al* [83] are the first to propose to use a conditional VAE for ZSL where they view ZSL as a case of missing data. They find that such an approach can handle well the domain shift problem. Similarly, Arora *et al* [4] show that a conditional VAE can be used together with a GAN system to synthesize images for unseen classes effectively. Keshari *et al* [47] focus on generating a specific set of *hard* samples which are closer to another class and the decision boundary. For the most part, ZSL methods aim to model the whole distribution of data [6, 12, 75, 100].

Chapter 3

Feature Augmentation via Variational Disentangling for Fine-grained Few-shot Classification

3.1 Overview

Fine-grained visual data are hard to collect and costly to annotate [48, 115, 123]. Fine-grained visual datasets often become quite long-tailed and lead to classifiers overfitting to the abundant classes when trained in vanilla settings. Fine-grained few-shot learning (FSL) methods alleviate this problem since they learn discriminative class features, among visually similar classes, using as few as 5 or 1 training instances.

Augmenting the few-shot classes by generating additional data is a straightforward way to mitigate issues of overfitting in FSL. Nevertheless, generating diverse data reliably remains an open question [53, 105]. The generated samples should contain the class-discriminative features while exhibiting high intra-class diversity. A typical data synthesis approach is generating new samples based on adversarial frameworks [3, 32, 54, 55, 57, 61, 114, 144]. However, these methods suffer from a lack of diversity in the generated samples as adversarial training often mode-collapses. Another approach is the feature transfer that transfers the intra-class variance from the base classes, which have many training

samples, to augment features for the novel classes, in which only few samples are available [38, 101, 138]. These methods are based on a common assumption that intra-class variations induced by poses, backgrounds, or illumination conditions are shared across categories. The intra-class variations are either modelled as low-level statistics [138] or pairwise variations [38, 101] and are applied directly on the novel samples. In this work, we discuss two potential issues with these approaches. First, these transformations can introduce certain class-discriminative features that could alter the class-identity of the transformed features. For example, only 8.7% of the augmented features using the Δ-encoder[101] have their nearest "real sample" neighbors belong to the same classes as the original samples (see Fig. 6.1). Second, the extracted variations might not be relevant to a specific novel sample, i.e., some bird species would never appear in sea backgrounds. Applying irrelevant variations would result in noisy or meaningless samples and degrade classification results (see Sec. 3.4.1). These two issues are more pronounced for fine-grained classification since a small change in feature space might change the category of the feature due to the small inter-class distances.

We address these issues in this work via a novel data augmentation framework. First, we disentangle each feature into two components: one that captures the intra-class variance, which we refer as intra-class variance features, and the other that encodes the class-discriminative features. Second, we model intra-class variance via a common distribution from which we can easily sample the new intra-class variations that are relevant for diversifying a specific instance. We show that both the feature disentanglement and the distribution of intra-class variability can be approximated using data from the base classes and it generalizes well to the novel classes. The two key supervision signals that drive the training of our framework are: 1) A classification loss that ensures that the class-discriminative features contain class specific information, 2) A Variational Auto-Encoder (VAE) [51] system that explicitly models intra-class variance via an isotropic Gaussian distribution. Our method works especially well for fine-grained datasets where the intra-class variations are similar across classes, achieving state-of-the-art few-shot classification performances on the CUB[123], NAB[115], and Stanford Dogs[48] datasets, outperforming previous methods [59, 101] by a large margin. We show in our analyses that the data generated by our method lies closely to the real-and-unseen features of the same class and can closely approximate the

distribution of the real data.

To sum up, our contributions are:

1. We are the first to propose a VAE-based feature disentanglement method for fine-grained FSL.

2. We show that we can train such a system using sufficient data from the base classes. We can sample from the learnt distribution to obtain relevant variations to diversify novel training instances in a reliable manner.

3. Our method outperforms state-of-the-art FSL methods in multiple fine-grained datasets by a large margin.

3.2 Proposed Method

3.2.1 Few-shot Learning Preliminaries

In FSL, abundant labeled images of base classes and a small number of labeled images of novel classes are given. Our goal is to train a classifier that can correctly classify novel class images with the few given examples. The standard FSL procedure includes a training stage and a fine-tuning stage. During the training stage, we use base class images to train a feature extractor and the classifier. Then in the fine-tuning stage, we freeze the parameters of the pre-trained feature extractor and train a new classifier head using the few labeled examples in the novel classes . In the testing stage, the learned classifier predicts labels on a set of unseen novel class images.

Since the available samples during the fine-tuning stage are scarce and lack diversity, the learned classifier tends to overfit to the few samples and thus performs poorly on the test images. To address this, we augment the training samples with our proposed data augmentation method, which significantly improves the performance of the baseline.

3.2.2 Overall Pipeline

Our goal is to generate additional features of the few novel class images which contain larger intra-class variance. Fig. 3.1 illustrates the pipeline

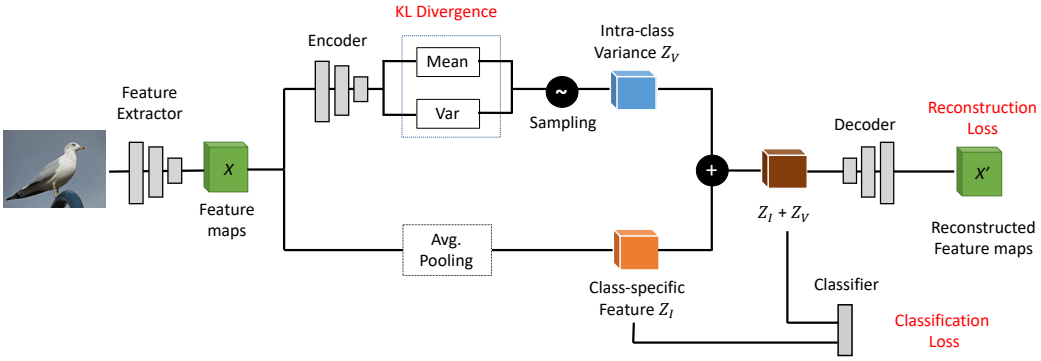


Figure 3.1: **The pipeline of our proposed method**. The input image is mapped into the image feature maps $\boldsymbol{X}$. We input $\boldsymbol{X}$ into an Encoder to obtain the mean and variance of the intra-class variability distribution that are used to sample the intra-class variance feature z_V. The class-specific feature z_I is obtained by max-pooling $\boldsymbol{X}$. z_V is forced to follow an isotropic multivariate Gaussian distribution. Both z_I and the combined features are used to train a classifier. We sample from the learned distribution repeatedly to get multiple z_V and add them to the class-specific feature z_I to get the augmented features. These augmented features are used together with the original ones to train a more robust classifier.

of our proposed method. We decompose the feature representation of an input image into two components, the class-specifc feature z_I and the intra-class variance feature z_V. z_V is constrained to follow a prior distribution. Then we repeatedly sample new intra-class variance features $\tilde{z}_V$ from the distribution and add them to the class-specific feature z_I to get augmented features. The augmented features are used together with the original features to train the final classifier. In the following sections, we will describe how we model the distribution of intra-class variability via variational inference and how we use it to diversify samples from the novel set.

3.2.3 Variational Inference for Intra-class Variance

Given an input image (i), we first use a feature extractor to map it into a feature map $X^{(i)}$. We then compute the intra-class variance feature $z_V^{(i)}$ and the class-specific feature $z_I^{(i)}$ from $X^{(i)}$ such that the embedding of the

input image, $z^{(i)}$, can be expressed as:

$$z^{(i)} = z_I^{(i)} + z_V^{(i)}. \tag{3.1}$$

Here we assume that the intra-class variance feature is generated from some conditional distribution $p(z_V)$ and the feature map $X^{(i)}$ is generated from some conditional distribution $p(X|z)$.

The class-specific feature $z_I^{(i)}$ can be learned by minimizing the cross-entropy loss given the class label $y^{(i)}$:

$$L_{cls}(X^{(i)}) = L_{cross-entropy}\left(W(z_I^{(i)}), y^{(i)}\right) \tag{3.2}$$

where W is a classifier with a single fully connected layer.

We use variational inference to model the posterior distribution of the variable z_V. Specifically, we approximate the true posterior distribution $p(z_V|X)$ with another distribution $q(z_V|X)$. The Kullback-Leibler divergence between the true distribution and the approximation is:

$$KL[q(z_V|X)||p(z_V|X)] = \int_Z q(Z|X)\log\frac{q(Z|X)}{p(Z|X)}. \tag{3.3}$$

Since the Kullback-Leibler divergence is always greater than or equal to zero, maximizing the marginal likelihood $p(X^{(i)})$ is equivalent to maximizing the evidence lower bound (ELBO) defined as follows:

$$\begin{aligned} ELBO^{(i)} = E_{q(z_V^{(i)}|X^{(i)})}[\log p(X^{(i)}|z_V^{(i)})] \\ - KL\left(q(z_V^{(i)}|X^{(i)})||p(z_V)\right). \end{aligned} \tag{3.4}$$

Prior work [68, 138, 142] has shown that the distribution of intra-class variability can be modelled with a Gaussian distribution. Here we set the prior distribution of z_V to be a centered isotropic multivariate Gaussian: $p(z_V) = \mathcal{N}(0, I)$. For the posterior distribution, we set it to be a multivariate Gaussian with diagonal covariance:

$$q(z_V^{(i)}|X^{(i)}) = \mathcal{N}(\mu^{(i)}, \sigma^{(i)}), \tag{3.5}$$

where $\mu^{(i)}$ and $\sigma^{(i)}$ are computed by a probablistic encoder. With the reparameterization trick, we obtain $z_V^{(i)}$ as follows:

$$z_V^{(i)} = \mu^{(i)} + \sigma^{(i)} * \epsilon, \epsilon \sim \mathcal{N}(0, I). \tag{3.6}$$

Since $z_I^{(i)}$ is deterministic given $X^{(i)}$, we have $p(X^{(i)}|z_V^{(i)}) = p(X^{(i)}|z_V^{(i)}, z_I^{(i)}) = p(X^{(i)}|z^{(i)})$. To estimate the maximum likelihood $p(X^{(i)}|z^{(i)})$, we use a decoder to reconstruct the original feature map from $z^{(i)}$ and minimize the $L2$ distance between the original feature map and the reconstructed one.

From Eq. 3.4, we now derive the loss function for the modeling of intra-class variance:

$$L_{intra}(X^{(i)}) = \|X^{(i)} - \hat{X}^{(i)}\|^2 + KL\Big(q(z_V^{(i)}|X^{(i)})\|p(z_V)\Big), \tag{3.7}$$

where $\hat{X}^{(i)}$ is the reconstructed feature map synthesized from the sum of class-specific feature $z_I^{(i)}$ and intra-class variance feature $z_V^{(i)}$ sampled from the distribution $\mathcal{N}(\mu^{(i)}, \sigma^{(i)})$.

The L_{intra} loss includes two terms. The first term is the reconstruction term, which ensures that the encoder extracts meaningful information from the inputs. The second term is a regularization term, which forces the latent code, $z_V^{(i)}$, to follow a standard normal distribution. Here, instead of minimizing the Kullback-Leibler divergence directly, we decompose it into three terms as in [13]:

$$\begin{aligned} KL[q(z_V|X)\|p(z_V)] = KL(q(z_V, X)\|q(z_V)p(X)) + \\ KL(q(z_V)\|\prod_j q(z_{V_j})) + \sum_j KL(q(z_{V_j})\|p(z_{V_j})), \end{aligned} \tag{3.8}$$

where z_{V_j} denotes the j-th dimension of the latent variable.

The three terms in Eq. 3.8 are referred to as the *index-code mutual information*, *total correlation*, and *dimension-wise* KL respectively. Prior work [2, 10, 13] has shown that penalizing the index-code mutual information and total correlation terms leads to a more disentangled representation while the dimension-wise KL term ensures that the latent variables do not deviate too far form the prior. Similar to [13], we penalize the total correlation with a weight α and rewrite L_{intra} as follows:

$$L_{intra}(X^{(i)}) = \|X^{(i)} - \hat{X}^{(i)}\|^2 + KL\Big(q(z_V^{(i)}, X^{(i)})\|q(z_V^{(i)})p(X)\Big) + \\ \alpha * KL\left(q(z_V^{(i)})\|\prod_j q(z_{V_j}^{(i)})\right) + \sum_j KL\Big(q(z_{V_j}^{(i)})\|p(z_{V_j})\Big). \tag{3.9}$$

The combination of L_{cls} and L_{intra} drives the model to extract discriminative class-specific features $z_I^{(i)}$ and model the distribution of intra-class variability simultaneously.

3.2.4 Objective Function

Given the distribution of intra-class variability, we can generate additional samples for the base classes during the training stage. For input image (i) with extracted class-specific feature $z_I^{(i)}$ and intra-class variability mean and variance $\mu^{(i)}$ and $\sigma^{(i)}$ respectively, we sample new intra-class variance features, $\tilde{z}_V^{(i)}$, for this image from the distribution $\mathcal{N}(\mu^{(i)}, \sigma^{(i)})$ and add them to $z_I^{(i)}$ to obtain the augmented features $\tilde{z}^{(i)} = z_I^{(i)} + \tilde{z}_V^{(i)}$. We use these features to train our system using the following cross-entropy loss:

$$L_{aug}(X^{(i)}) = L_{cross-entropy}\Big(W(\tilde{z}^{(i)}), y^{(i)}\Big). \tag{3.10}$$

The overall loss function in the training stage is a weighted combination of the aforementioned terms:

$$L = L_{cls} + L_{intra} + \beta * L_{aug}, \tag{3.11}$$

where β is the coefficient of L_{aug}.

3.2.5 Diversifying Samples for Few-Shot Classes

In this section, we discuss how to use our model to diversify samples for few-shot classes. Our intra-class variance is modelled by an isotropic Gaussian distribution. Sampling from this distribution would result in an arbitrary intra-class variance feature. However, we conjecture that such an arbitrary feature may not be relevant for all instances, i.e., some birds

never appear with a background of the sea. Note that here as all intra-class variations are mapped into a common continuous embedding space via variational inference and closely related or similar intra-class variations likely form local neighborhoods in the embedding space. Thus, instead of sampling from the zero-mean and unit-variance distribution, we only sample from the mean and variance estimated directly from the conditional sample to obtain the likely relevant intra-class variations to this sample.

Specifically, given an image of novel class $(i)^*$ with class label $y^{(i)*}$, we first extract the feature map $X^{(i)*}$, the class-specific feature $z_I^{(i)*}$, and the mean and variance of the intra-class variability distribution $\mu^{(i)*}$ and $\sigma^{(i)*}$ for this instance. We then generate additional features by adding the class-specific features $z_I^{(i)*}$ with a biased term sampled from the distribution of intra-class variability.

$$\tilde{z}^{(i)*} = z_I^{(i)*} + \tilde{z}_V^{(i)*}, \tilde{z}_V^{(i)*} \sim N(\mu^{(i)*}, \sigma^{(i)*}), \tag{3.12}$$

where $\tilde{z}^{(i)*}$ is the augmented feature and $\tilde{z}_V^{(i)*}$ is sampled from the posterior distribution $N\left(\mu^{(i)*}, \sigma^{(i)*}\right)$. By sampling from $N\left(\mu^{(i)*}, \sigma^{(i)*}\right)$ multiple times, we get multiple augmented features $\tilde{z}^{(i)*}$ that can be used to train the classifier. In Sec. 3.4.1, we verify the effectiveness of this sampling scheme.

3.3 Experiments

3.3.1 Datasets

We evaluate our method on three fine-grained image classification datasets: Caltech UCSD Birds (CUB) [123], North America Birds (NAB) [115] and Stanford Dogs [48]. The CUB dataset contains 11,788 bird images from 200 bird species in total. Following the setup introduced in [123], we sample the base classes from the 100 classes provided for training, and sample the novel set from the 50 classes provided for testing. The NAB dataset contains 48,527 bird images with 555 classes, which is four times larger than CUB. Similar to [114], we adopt a 2:1:1 training, validation and test set split. The Stanford Dogs dataset is a subset of the

Imagenet dataset designed for fine-grained image classification with 90 categories for training and validation and 30 testing categories.

3.3.2 Implementation Details

We conduct experiments with two architectures of our feature extractor: ResNet12 and Conv4 for fair comparisons with other methods using similar architectures. **ResNet12** [41] contains 4 Residual blocks. Each residual block is composed of 3 *conv* layers with 3×3 kernels. A 2×2 max-pooling layer is applied at the end of each residual block. **Conv4** consists of 4 layers with 3×3 convolutions and 32 filters, followed by batch normalization (BN) , a ReLU nonlinearity, and 2×2 max-pooling.

The class-specific features are calculated by average-pooling the output of the feature extractor. The encoder consists of three *conv* blocks followed by two fully-connected heads that output the μ and $\log\sigma^2$ respectively. The decoder consists of a fully connected layer followed by three Convolutional blocks.

Training policies. The whole network is trained from scratch in an end-to-end manner. In the training stage, we use the Adam optimizer [50] on all datasets with initial learning rate 0.001 . We train our model for 100 epochs in total with a batch size of 16 and reduce the learning rate by 0.1 at the 40-th and 80-th epochs. We empirically set $\alpha = 4$ in Eq. 3.9 and $\beta = 1$ in Eq.3.11.

We follow a standard few-shot evaluation scheme. In the fine-tuning stage, we select 5 classes from the novel classes randomly. For each class, we pick k instances as the support set and 16 instances for the query set for a k-shot task. The extracted features of all support set images along with the augmented features are used to train a linear classifier for 100 iterations with a batch size of 4. For each feature extracted from a support image, we obtain five augmented features. The final results are averaged over 600 experiments. For data augmentation, we adopt random cropping, horizontal flipping and color jittering as in [14]. The final size of the input images is 84×84 .

3.3.3 Results

Tab. 3.1 summarizes the 5-way classification accuracy of various methods using ResNet12 backbones. The results are obtained using the publicly

Method	CUB		NAB		Stanford Dogs	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Baseline [14]	63.90 ± 0.88	82.54 ± 0.54	70.36 ± 0.89	87.91 ± 0.49	63.53 ± 0.89	79.95 ± 0.59
Baseline++ [14]	68.46 ± 0.85	81.02 ± 0.46	76.00 ± 0.85	90.99 ± 0.41	58.30 ± 0.35	73.77 ± 0.68
MAML [31]	71.11 ± 1.00	82.08 ± 0.72	80.08 ± 0.93	88.87 ± 0.54	66.56 ± 0.66	79.32 ± 0.35
MatchingNet [116]	72.62 ± 0.90	84.14 ± 0.50	73.91 ± 0.72	88.17 ± 0.45	65.87 ± 0.81	80.70 ± 0.42
ProtoNet [107]	71.57 ± 0.89	86.37 ± 0.49	73.60 ± 0.83	89.72 ± 0.41	65.02 ±0.92	83.69 ± 0.48
RelationNet [112]	70.20 ± 0.84	84.28 ± 0.46	67.41 ± 0.82	85.47 ± 0.43	59.38 ± 0.79	79.10 ± 0.37
MTL [111]	73.31 ± 0.92	82.29 ± 0.51	78.69 ± 0.78	87.74 ± 0.34	54.96 ± 1.03	68.76 ± 0.65
Δ-encoder [101]	73.91 ± 0.87	85.60 ± 0.62	79.42 ± 0.77	92.32 ± 0.59	68.59 ± 0.53	78.60 ± 0.78
MetaOptNet [59]	75.15 ± 0.46	87.09 ± 0.30	84.56 ± 0.46	93.31 ± 0.22	65.48 ± 0.49	79.39 ± 0.25
Ours	**79.12** ± 0.83	**91.48** ± 0.39	**88.62** ± 0.73	**95.22** ± 0.32	**76.24** ± 0.87	**88.00** ± 0.47

Table 3.1: Few-shot classification accuracy on the CUB [123], NAB [115], and Stanford Dogs [48] dataset. All experiments are from 5-way classification with the same backbone network (ResNet12). The best performance is indicated in bold.

available code of each method. Our proposed method outperforms the previous methods by a large margin for both 1-shot and 5-shot settings on all three datasets. Compared with the Δ-encoder [101], another data augmentation based method, our proposed method achieves 7.40%, 9.20% and 7.65% performance gain for the 1-shot setting and 5.88%, 2.90% and 9.40% performance gain for the 5-shot setting on the three datasets respectively. It can be seen that our improvement in the 1-shot setting is more pronounced than in the 5-shot setting since the 1-shot setting is a more extreme case of data scarcity, in which augmenting the training data tends to be more useful.

We compare with methods using Conv4 architectures as the backbone networks in Tab. 3.2. Here the majority of methods only report their results on the CUB and Stanford Dogs datasets. Our proposed method achieves state-of-the-art performance for both the 1-shot and 5-shot settings. Especially for the 1-shot setting, our method obtains 2.12% performance gain for the CUB and 2.19% gain for the Stanford Dogs over MattML [148], a newly proposed method that is aimed specifically at fine-grained few-shot visual recognition.

Our method also achieves competitive few-shot classification performances on non fine-grained datasets such as CIFAR-FS[7] and mini-ImageNet[95, 116].

Method	CUB		Stanford Dogs	
	1-shot	5-shot	1-shot	5-shot
MatchingNet [116]	45.30 ± 1.03	59.50 ± 1.01	35.80 ± 0.99	47.50 ± 1.03
ProtoNet [107]	37.36 ± 1.00	45.28 ± 1.03	37.59 ±1.00	48.19 ± 1.03
RelationNet [112]	58.99 ± 0.52	71.20 ± 0.40	43.29 ± 0.46	55.15 ± 0.39
MAML [31]	58.13 ± 0.36	71.51 ± 0.30	44.84 ± 0.31	58.61 ± 0.30
adaCNN [84]	56.76 ± 0.50	61.05 ± 0.44	42.16 ± 0.43	54.12 ± 0.39
CovaMNet [73]	52.42 ± 0.76	63.76 ± 0.64	49.10 ± 0.76	63.04 ± 0.65
DN4 [63]	53.15 ± 0.84	81.90 ± 0.60	45.73 ± 0.76	61.51 ± 0.85
LRPABN [44]	63.63 ± 0.77	76.06 ± 0.58	45.72 ± 0.75	60.94 ± 0.66
MattML [148]	66.29 ± 0.56	80.34 ± 0.30	54.84 ± 0.53	71.34 ± 0.38
Ours	**68.42** ± 0.92	**82.42** ± 0.61	**57.03** ± 0.86	**73.00** ± 0.66

Table 3.2: Few-shot classification accuracy on the CUB [123] and Stanford Dogs [48] dataset. All experiments are from 5-way classification with the same backbone network (Conv4). The best performance is indicated in bold.

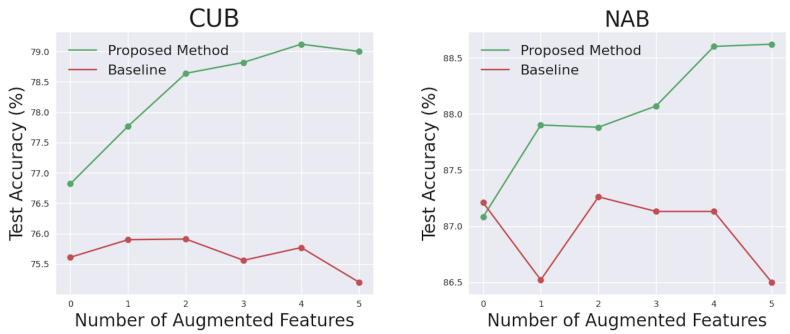


Figure 3.2: **Analysis on the generated intra-class variations.** We augment samples with the intra-class variance features sampled from the estimated mean and variance (green lines) or from the zero-mean and unit-variance (red lines). Our sampling scheme generates features that consistently improve classification.

3.4 Analyses

In this section, we provide additional experiments to clarify different aspects of our methods.

3.4.1 Analysis on the Generated Intra-class Variations

We conduct a simple experiment to verify the effectiveness of our sampling method (Sec.3.2.5). Instead of sampling from the instance-conditioned mean and variance, we sample the intra-class variance feature from the zero-mean and unit-variance distribution.

Fig. 3.2 summarizes the results of this experiment for 5-way 1-shot classification on the CUB and NAB dataset. As can be seen, intra-class variance features sampled from zero-mean and unit-variance do not improve the results (red lines). In contrast, our method of sampling from the instance-conditioned posterior distribution generates features that consistently improve classification performance as the number of augmented samples increases.

3.4.2 Comparison to Other Augmentation Methods

We compare our method with two other data augmentation based FSL methods: MetaIRNet[114] and Δ-encoder[101]. MetaIRNet uses a pre-trained image generator to synthesize additional images and combine them with the original images to form additional training samples. The Δ-encoder learns to synthesize transferable non-linear deformations between pairs of examples of seen classes and apply these deformations to the few provided samples of novel categories.

We use the additional samples synthesized by both of these methods to train three types of classifiers: K-nearest neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression (LR), which are then used to classify novel images. The comparisons between these methods and our method are shown in Tab. 3.3. The superior performance of our method demonstrates that the augmented features obtained by our framework is beneficial for various types of classifiers. Note that for MetaIRNet [114], the results in Tab. 3.3 are lower than their numbers reported in the original paper since they pre-trained the backbone network on ImageNet while here all methods are trained from scratch.

In Tab. 3.4, we directly compare our method withthe Δ-encoder using K-NN classifiers (K=1). Interestingly, it can be seen that the augmented features generated using the delta-encoder decrease classification performance. In fact, we observe that the majority (91.3%) of the nearest neighbors of the Δ-encoder's generated features belong to different classes

Method	KNN		SVM		LR	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MetaIRNet [114]	63.18	74.82	63.76	76.77	63.53	79.95
Δ-Encoder [101]	67.31	82.67	76.02	82.87	76.22	85.17
Ours	**75.46**	**83.17**	**79.07**	**87.59**	**78.34**	**89.30**

Table 3.3: **Analysis of different classifiers.** Few-shot classification accuracy on the CUB [123] dataset in 1-shot and 5-shot settings with different types of classifiers.

(some are visualized in Fig. 6.1), suggesting that the pairwise transformations extracted from this method might alter the class-identities of the transformed features. On the other hand, our generated features preserve well the class identity and mildly improve the classification results.

Method	Δ-Encoder		Ours	
	w/o Aug	w/ Aug	w/o Aug	w/ Aug
5-way	69.37	67.31	74.95	75.46
10-way	58.69	52.19	62.05	63.17
20-way	48.10	38.84	50.19	50.72

Table 3.4: **Effect of augmented features on 1NN classifier.** Few-shot classification accuracy on the CUB [123] dataset using 1NN classifier with original features vs augmented features. The original features of the Δ-Encoder are from a pre-trained ResNet18 network.

Intra-class distribution model	CUB		NAB	
	1-shot	5-shot	1-shot	5-shot
Gaussian Mixture Model [22]	75.16	86.46	86.49	94.72
Covariance Matrix [138]	75.28	87.84	84.71	94.19
No disentanglement [142]	73.40	86.60	81.83	92.83
Isotropic Gaussian (Proposed)	**79.12**	**91.48**	**88.62**	**95.22**

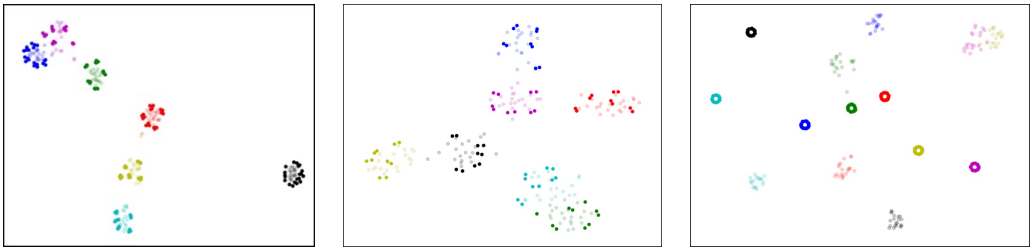
Table 3.5: Few-shot classification accuracy on the CUB [123] and NAB [115] dataset in 1-shot and 5-shot setting with different methods to model intra-class variance.

3.4.3 Comparison to Other Intra-class Variance Modeling Methods

We assume that the intra-class variance can be modelled with an isotropic multivariate Gaussian distribution in a latent space. In this section, we compare this method with other methods that model the intra-class variance including Gaussian mixture variational autoencoder (GMVAE) [22], covariance matrix [138], and a baseline model where we do not disentangle intra-class variance features from class-discriminative features.

Tab. 3.5 summarizes the results. The first row shows the results for GMVAE. This method enforces that the latent space is divided into distinct clusters for different classes. However, for this model, the accuracy drops by 6.15% and 2.13% for the 1-shot setting and 5.02% and 0.50% for the 5-shot setting on the CUB and NAB datasets respectively. The results align with our assumption that the intra-class variance is shared across different classes. Thus, enforcing a multi-modal prior distribution would lead to performance degradation.

The second row shows the results for the method proposed in [138] based on covariance matrices. Specifically, this method assumes a Gaussian prior on the distribution of intra-class variability across different classes which can be transferred from the base classes to the rare classes. However, instead of modelling the distribution by variational inference, [138] uses a covariance matrix to estimate the feature variance distribution.Here we apply this method on our extracted features to generate additional features on the CUB and NAB datasets under both 1-shot and 5-shot settings. Compared with the non-parametric estimate of the Gaussian distribution, modelling intra-class variance via variational inference in an end-to-end manner brings 6.03% and 1.91% improvement for the 1-shot setting and 3.64% and 1.03% improvement for the 5-shot setting on the CUB and NAB dataset respectively. Last, we provide the results for our method without feature disentanglement, denoted as "No disentanglement" in the third row. In spirit, this model is similar to [142] which models each point as a distribution via variational inference. Given a new sample, we augment it via sampling repeatedly from the estimated mean and variance. Without feature disentanglement and explicit modelling of the intra-class variance, this model does not achieve comparable results compared to other methods.



(a) Ours - Base Classes (b) Ours - Novel Classes (c) Δ-encoder - Novel Classes

Figure 3.3: **Distance Distributions.** Kernel Density Estimation of the distance between the estimated prototypes and the ground truth prototype. A smaller value means the estimated prototypes are closer to the ground truth prototypes.

3.4.4 Data Distribution Analysis

We compare the data distributions between the real data and the generated data from our method in comparison to other state-of-the-art data generation methods [101, 138]. Here we measure the average intra-class variance, the distances between classes (inter-class distances), and the data clusterability via the Davies–Bouldin index (DBI) [19]. Specifically, the DBI for a cluster i is calculated by:

$$DBI_i = \max_{i \neq j} \frac{Intra_{(i)} + Intra_{(j)}}{Inter_{(i,j)}} \tag{3.13}$$

where $Intra_{(i)}$ is the intra-class variance of cluster i, calculated by taking the average of squared deviations from the class center. $Inter_{(i,j)}$ is the distance between the two class centers of clusters i and j. The lower the value of the DBI, the better the separation between the clusters and the "tightness" inside the clusters.

Tab. 3.6 shows the average values of the intra-class variance, inter-class distances, and the DBI (denoted as D_{intra}, D_{inter}, and DBI respectively) across all novel classes of the CUB dataset. The inter-class distances are averaged across all pairs of classes. As shown in the table, features from the support set exhibit smaller intra-class variance compared to features from all data. All methods augment features from the support set. Interestingly, both sets of generated features using the method proposed in [138] and the Δ-encoder[101] decrease intra-class variance. On the other

	D_{intra}	D_{inter}	DBI
Support data (5 samples)	21.52	32.77	2.21
All data	28.97	35.89	3.02
Covariance matrix [138]	17.98	35.24	1.79
Encoder-based Model [101]	10.34	11.69	1.77
Ours	27.27	34.12	2.53

Table 3.6: **Data Distribution analysis for different sets of features**. We augment features using our method and other data generation method based on covariance matrices [138] or the Δ-encoder [101]. All methods augment features from the support set (first row).

hand, the set of features augmented by our method closely approximate the data distribution of the set of all real features.

Fig. 3.3 demonstrates how real samples and generated samples from our method are distributed in a 2D space in comparison with the Δ-encoder [101] using t-SNE [77]. The original features are marked as light colors while the augmented features are marked as dark colors. Different colors denote different classes. The visualization for the base classes with the augmented features from our method is shown in Fig. 3.3a. Fig. 3.3b visualizes the real features and the generated features of our method for the novel classes. Our method generates samples that follow closely the real samples. The visualization for the novel classes and the generated features from the Δ-encoder is shown in Fig. 3.3c. As can be seen, the generated data from each novel class forms into a new cluster and does not lie close to the actual data points.

3.5 Conclusion

We have proposed a simple, yet effective, feature augmentation method via feature disentanglement and variational inference to address the data scarcity problem in few-shot fine-grained classification. The generated features enlarge the intra-class variance for novel set images while preserving the class-discriminative features. The consistent performance improvement with the increase of the number of augmented samples suggests that the learned features are meaningful and nontrivial. The higher accuracy compared with other data augmentation based methods further demonstrate the superiority of our method. While this work mainly fo-

cuses on few-shot recognition problems, a promising future direction is to apply the feature transfer idea to other data-scarce or label-scarce tasks.

Chapter 4

Generating Representative Features for Few-shot Classification

4.1 Overview

Few-shot learning (FSL) methods aim to learn useful representations with limited training data. They are extremely useful for situations where machine learning solutions are required but large labelled datasets are not trivial to obtain (e.g. rare medical conditions [86, 117], rare animal species [123], failure cases in autonomous systems [78, 79, 98]). Generally, FSL methods learn knowledge from a fixed set of base classes with a surplus of labelled data and then adapt the learned model to a set of novel classes for which only a few training examples are available [120].

Many FSL methods [15, 49, 72, 107, 107, 130, 140] employ a prototype-based classifier for its simplicity and good performance. They aim to find a prototype for each novel class such that it is close to the testing samples of the same class and far away from testing samples for other classes. However, it is challenging to estimate a representative prototype just from a few available support samples [70, 134]. An effective strategy to enhance the representativeness of the prototype is to employ textual semantic embeddings learned via NLP models[21, 82, 88, 91] using large unsupervised text corpora [130, 140]. These semantic embeddings implicitly associate a class name, such as "Yorkshire Terriers", with the class representative

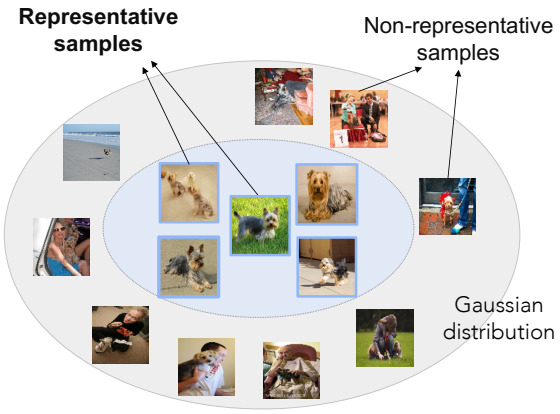


Figure 4.1: **Representative Samples.** We refer representative samples to the "easy-to-recognize" samples that faithfully reflect the key characteristics of the category. We identify those samples and then use them to train a VAE model for feature generation, conditioned on class-representative semantic embeddings. We show that the generated data significantly improves few-shot classification performance.

semantic attributes such as "smallest dog" or "long coat" [1] (Fig. 6.1), providing strong and unbiased priors for category recognition.

For the most part, current FSL methods focus on learning to adaptively leverage the semantic information to complete the original biased prototype estimated from the few available samples. For example, the recent FSL method of Zhang *et al* [140] learns to fuse the primitive knowledge and attribute features into a representative prototype, depending on the set of given few-shot samples. Similarly, Xing *et al* [130] propose a method that computes an adaptive mixture coefficient to combine features from the visual and textual modalities. However, learning to recover an arbitrarily biased prototype is challenging due to the drastic variety of the possible combinations of few-shot samples.

In this work, we propose a novel method to obtain class-representative prototypes. Inspired by zero-shot learning (ZSL) methods[5, 34, 143], we propose to generate visual features via a variational autoencoder (VAE) model [108] conditioned on the semantic embedding of each class. This VAE model learns to associate a distribution of features to a conditioned semantic code. We assume that such association generalizes across the

base and novel classes [4, 83]. Therefore, the model trained with sufficient data from the base classes can generate novel-class features that align with the real unseen features. We then use the generated features together with the few-shot samples to construct class prototypes. We show that this strategy achieves state-of-the-art results on both *mini*ImageNet and *tiered*ImageNet datasets. It works exceptionally well for 1-shot scenarios where our method outperforms state-of-the-art methods[124, 137] by $5 \sim 6\%$ in terms of classification accuracy.

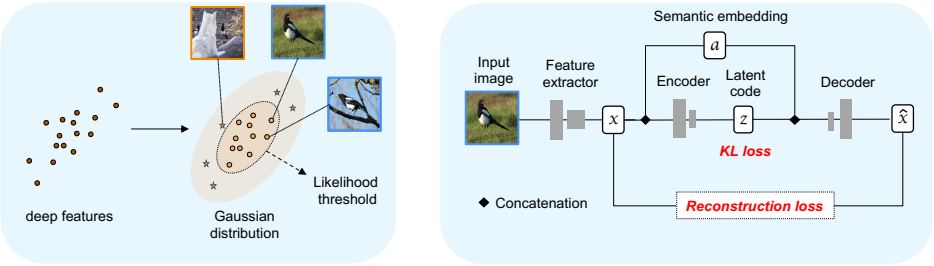
Moreover, to enhance the representativeness of the prototype, we guide the VAE to generate more *representative* samples. Here we refer representative samples to the "*easy-to-recognize*" samples that faithfully reflect the key characteristics of the category (see Fig. 6.1). The embeddings of these representative samples often lie close to their corresponding class centers, which are particularly useful for constructing class-representative prototypes.

Specifically, we guide the VAE model to generate representative samples by selecting only representative data from the base classes for training it. In essence, our VAE model is trained to model the data distribution of the training set. As the training set contains only representative data, the trained VAE model outputs samples that are also representative. Specifically, to select those representative features, we first assume that the feature vectors of each class follow a multivariate Gaussian distribution and estimate this distribution for each base class. Based on these distributions, we compute the probability of each sample belonging to its corresponding category to measure the representativeness for the sample. We filter out the non-representative samples and train the VAE using only representative samples. Interestingly, we show that the representativeness of the training set highly corresponds to the accuracy of the few-shot classifier. We obtain the highest accuracy when training the VAE with the most representative samples. In this case, we only use a small percentage of the whole training set, e.g., 10% for the case of *mini*Imagenet dataset, to obtain the best results. Our analyses show that this approach consistently improves the FSL classification performance by $1 \sim 2\%$ across all benchmarks for three different baselines[15, 72, 107].

Our main contributions can be summarized as follows:

- We are the first to use a VAE-based feature generation approach conditioned on class semantic embeddings for few-shot classification.

- We propose a novel sample selection method to collect representative samples. We use these samples to train a VAE model to obtain reliable data points for constructing class-representative prototypes.
- Our experiments show that our methods achieve state-of-the-art performance on two challenging datasets, *tiered*ImageNet and *mini*-ImageNet.



(a) Sample Selection method **(b) Conditional VAE model**

Figure 4.2: **Overview –** The key aspect of our approach is to subset our training set to the most representative samples to train a conditional VAE model that generates more representative features. **(a)** To select representative samples, we assume that the features of each class follow a multivariate Gaussian distribution. We estimate the distribution parameters and compute a probability for each data point belonging to the class distribution. We identify a set of representative samples by setting a threshold on the probability. **(b)** We train a VAE to generate visual features, conditioned on the semantic embedding of each class. Using only representative samples (the output of the sample selection step) to train this VAE model improves the representativeness of the generated samples.

4.2 Proposed Method

4.2.1 Problem Definition

In a typical few-shot classification setting, we are given a set of data-label pairs $\mathrm{D} = \{(x^i, y^i)\}$. Here $x^i \in R^d$ is the feature vector of a sample and $y^i \in C$, where C denotes the set of classes. The set of classes is divided into base

classes C_b and novel classes C_n. The sets of class C_b and C_n are disjoint, *i.e.* $C_b \cap C_n = \emptyset$. For a N-way K-shot problem, we sample N classes from the novel set C_n, and K samples are available for each class. K is often small (*i.e.*, $K = 1$ or $K = 5$). Our goal is to classify query samples correctly using the few samples from the support set.

4.2.2 Overall Pipeline

Fig. 4.2 gives an overview of our sample selection method and VAE training approach. We propose a method to select a set of representative samples from a set of base classes. We use these selected representative data to train a conditional VAE model for feature generation. To select representative samples, we assume that the features of each class follow a multivariate Gaussian distribution. We estimate the parameters for each class distribution and compute the probability for each data point belonging to its class. By setting a threshold on the probabilities, we identify a set of representative samples. We then use these selected representative samples to train a VAE model that generates samples conditioned on the semantic attributes of each class.

We train this VAE on the base classes and use the trained model to generate samples for the novel classes. The generated features are then used together with the few-shot samples to construct the prototype for each class. Our method is a simple plug-and-play module and can be built on top of any pretrained feature extractors. In our experiments, we show that our method consistently improves three baseline few-shot classification methods: Meta-Baseline [15], ProtoNet [107] and E3BM [72] by large margins.

Class-representative Sample Selection

In this work, we are interested in representative samples as they can serve as reliable data points for constructing a class-representative prototype[15, 107]. The main idea is to train a feature generator with only representative data to obtain more representative generated samples.

To select the representative features, we assume that the feature distribution of the base classes follows a Gaussian distribution and estimate the parameters of this distribution for each class. We calculate the Gaussian

mean of a base class i as the mean of every single dimension in the vector:

$$\mu^i = \frac{1}{n^i} \sum_{j=1}^{n^i} x^j, \tag{4.1}$$

where x^j is a feature vector of the j-th sample from the base class i and n^i is the total number of samples in class i. The covariance matrix Σ^i for the distribution of class i is calculated as:

$$\Sigma^i = \frac{1}{n^i - 1} \sum_{j=1}^{n^i} (x^j - \mu^i)(x^j - \mu^i)^T. \tag{4.2}$$

Once we estimate the parameters of the Gaussian distribution using the adequate samples from the base classes, the probability density of observing a single feature, x^j, being generated from the Gaussian distribution of class i is given by:

$$p(x^j|\mu^i, \Sigma^i) = \frac{\exp\{-\frac{1}{2}(x^j - \mu^i)^T {\Sigma^i}^{-1}(x^j - \mu^i)\}}{(2\pi)^{k/2}|\Sigma^i|^{1/2}}, \tag{4.3}$$

where k is the dimension of the feature vector.

Here we assume that the probability of a single sample belongs to its category's distribution reflects the representativeness of the sample, *i.e.*, the higher the probability, the more representative the sample is. By setting a threshold ϵ on the estimated probability, we filter out those samples with small probabilities and get a set of representative features for class i:

$$\mathbb{D}^i = \{x^j \mid p(x^j|\mu^i, \Sigma^i) > \epsilon\}, \tag{4.4}$$

where $\mathbb{D}^i$ stores the features for class i with the probabilities larger than a threshold ϵ.

Conditional VAE Model for Feature Generation

We use our sample selection method to select a set of representative samples and use them for training our feature generation model. We develop our feature generator based on a conditional variational autoencoder (VAE) architecture[108] (see Fig. 4.2b). The VAE is composed of

an Encoder $E(x, a)$, which maps a visual feature x to a latent code z, and a decoder $G(z, a)$ which reconstructs x from z. Both E and G are conditioned on the semantic embedding a. The loss function for training the VAE for a feature x^j of class i can be defined as:

$$\begin{aligned} L_V(x^j) =& \mathrm{KL}\Big(q(z|x^j, a^i) \| p(z|a^i)\Big) \\ & - \log p(x^j|z, a^i), \end{aligned} \tag{4.5}$$

where a^i is the semantic embedding of class i. The first term is the Kullback-Leibler divergence between the VAE posterior $q(z|x, a)$ and a prior distribution $p(z|a)$. The second term is the decoder's reconstruction error. $q(z|x, a)$ is modeled as $E(x, a)$ and $p(x|z, a)$ is equal to $G(z, a)$. The prior distribution is assumed to be $\mathcal{N}(0, I)$ for all classes.

The loss for training the feature generator is the loss over all selected representative training samples:

$$L_V = \sum_{i=1}^{C_b} \sum_{x \in \mathbb{D}^i} L_V(x) \tag{4.6}$$

Constructing Class Prototypes

After the VAE is trained on the base set, we generate a set of features for a class y by inputting the respective semantic vector a^y and a noise vector z to the decoder G:

$$\mathbb{G}^y = \{\hat{x} | \hat{x} = G(z, a^y), z \sim \mathcal{N}(0, I)\}. \tag{4.7}$$

The generated features along with the original support set features for a few-shot task is then served as the training data for a task-specific classifier. Following our baseline methods, we compute the prototype for each class and apply the nearest neighbour classifier. Specifically, we first compute two separated prototypes: one using the support features and the other using the generated features. Each prototype is the mean vector of the features of each group. We then take a weighted sum of the two prototypes to obtain the final prototype p^y for class y:

$$\mathrm{p}^y = w_g * \frac{1}{|\mathbb{G}^y|} \sum\nolimits_{\hat{x}^j \in \mathbb{G}^y} \hat{x}^j + w_s * \frac{1}{|\mathbb{S}^y|} \sum\nolimits_{x^j \in \mathbb{S}^y} x^j, \tag{4.8}$$

Method	Backbone	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
Matching Net [116]	ResNet-12	65.64 ± 0.20	78.72 ± 0.15	68.50 ± 0.92	80.60 ± 0.71
MAML [31]	ResNet-18	64.06 ± 0.18	80.58 ± 0.12	-	-
SimpleShot [119]	ResNet-18	62.85 ± 0.20	80.02 ± 0.14	69.09 ± 0.22	84.58 ± 0.16
CAN [43]	ResNet-12	63.85 ± 0.48	79.44 ± 0.34	69.89 ± 0.51	84.23 ± 0.37
S2M2 [80]	ResNet-18	64.06 ± 0.18	80.58 ± 0.12	-	-
TADAM [85]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30	62.13 ± 0.31	81.92 ± 0.30
AM3 [130]	ResNet-12	65.30 ± 0.49	78.10 ± 0.36	69.08 ± 0.47	82.58 ± 0.31
DSN [106]	ResNet-12	62.64 ± 0.66	78.83 ± 0.45	66.22 ± 0.75	82.79 ± 0.48
Variational FSL [142]	ResNet-12	61.23 ± 0.26	77.69 ± 0.17	-	-
MetaOptNet [59]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
Robust20-distill [24]	ResNet-18	63.06 ± 0.61	80.63 ± 0.42	65.43 ± 0.21	70.44 ± 0.32
FEAT [137]	ResNet-12	66.78 ± 0.20	82.05 ± 0.14	70.80 ± 0.23	84.79 ± 0.16
RFS [113]	ResNet-12	62.02 ± 0.63	79.64 ± 0.44	69.74 ± 0.72	84.41 ± 0.55
Neg-Cosine [69]	ResNet-12	63.85 ± 0.81	81.57 ± 0.56	-	-
FRN [124]	ResNet-12	66.45 ± 0.19	82.83 ± 0.13	71.16 ± 0.22	86.01 ± 0.15
Meta-Baseline [15]	ResNet-12	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.29 ± 0.18
Meta-Baseline + SVAE (Ours)	ResNet-12	69.96 ± 0.21	79.92 ± 0.16	73.05 ± 0.24	83.96 ± 0.18
Meta-Baseline + R-SVAE (Ours)	ResNet-12	72.79 ± 0.19	80.70 ± 0.16	73.90 ± 0.24	84.17 ± 0.18
ProtoNet [137]	ResNet-12	62.39	80.53	68.23	84.03
ProtoNet + SVAE (Ours)	ResNet-12	73.01 ± 0.24	83.13 ± 0.40	76.36 ± 0.65	85.65 ± 0.50
ProtoNet + R-SVAE(Ours)	ResNet-12	**74.84 ± 0.23**	**83.28 ± 0.40**	76.98 ± 0.65	85.77 ± 0.50
E3BM [72]	ResNet-12	64.09 ± 0.37	80.29 ± 0.25	71.34 ± 0.41	85.82 ± 0.29
E3BM + SVAE (Ours)	ResNet-12	73.07 ± 0.39	80.82 ± 0.31	79.85 ± 0.43	86.82 ± 0.32
E3BM + R-SVAE(Ours)	ResNet-12	73.35 ± 0.37	80.95 ± 0.31	**80.46 ± 0.43**	**86.99 ± 0.32**

Table 4.1: **Comparison to prior works on *mini*ImageNet and *tiered*ImageNet**. Average 5-way 1-shot and 5-way 5-shot accuracy (%) with 95% confidence intervals. SVAE denotes our method using the VAE trained with all features in the base set. R-SVAE denotes the one trained with only representative features. The **best** performance is highlighted in bold.

where $\mathbb{S}^y$ is the support set features and (w_g, w_s) are the coefficients of the generated feature prototype and the real feature prototype, respectively. We classify samples by finding the nearest class prototype for an embedding query feature. We conduct further analysis to show that our generated features can benefit all types of classifiers (see Section 4.4.2). Compared to the methods that correct the original biased prototype, our model does not require any carefully designed combination scheme.

4.3 Experiments

4.3.1 Experimental Settings

Datasets. We evaluate our method on two widely-used benchmarks for few-shot learning, *mini*ImageNet [116] and *tiered*ImageNet [96]. ***mini*ImageNet** is a subset of the ILSVRC-12 dataset [20] It contains 100 classes and each class consists of 600 images. The size of each image is 84×84. Following the evaluation protocol of [95], we split the 100 classes into 64 base classes, 16 validation classes, and 20 novel classes for pre-training, validation, and testing. ***tiered*ImageNet** is a larger subset of ILSVRC-12 dataset, which contains 608 classes sampled from hierarchical category structure. The average number of images in each class is 1281. It is first partitioned into 34 super-categories that are split into 20 classes for training, 6 classes for validation, and 8 classes for testing. This leads to 351 actual categories for training, 97 for validation, and 160 for testing.

Baseline methods. Our method can be used as a simple plug-and-play module for many existing few-shot learning methods without fine-tuning their feature extractors. We investigate three baseline few-shot classification methods used in conjunction with our method: ProtoNet [137], Meta-Baseline [15] and E3BM [72]. ProtoNet is known as a strong and classic prototypical approach. In our experiments, we use the ProtoNet implementation of Ye *et al* [137]. Meta-Baseline [15] uses a ProtoNet model to fine-tune a generic classifier via meta-learning. E3BM [72] meta-learns the ensemble of epoch-wise models to achieve robust predictions for FSL. For each baseline method, we extract the corresponding feature representations to train our feature generation VAE model. We then use the trained VAE to generate features and obtain the class prototypes for few-shot classification.

Evaluation protocol. We use the top-1 accuracy as the evaluation metric to measure the performance of our method. We report the accuracy on standard 5-way 1-shot and 5-shot settings with 15 query samples per class. We randomly sample 2000 episodes from the test set and report the mean accuracy with the 95% confidence interval.

4.3.2 Implementation Details

All the three baselines use ResNet12 backbone as the feature extractor. The feature representation is extracted by average pooling the final residual block outputs. The dimension of the feature representation is 640 for ProtoNet [137], 512 for Meta-Baseline [15], and 640 for E3BM[72]. For our feature generation model, both the encoder and the decoder are two-layer fully-connected (FC) networks with 4096 hidden units. LeakyReLU and ReLU [40] are the nonlinear activation functions in the hidden and output layers, respectively. The dimensions of the latent space and the semantic vector are both set to be 512. The network is trained using the Adam optimizer with 10^{-4} learning rate. Our semantic embeddings are extracted from CLIP [91]. We empirically set the combination weights $[w_g, w_s]$ in Equation 4.8 to $[\frac{1}{2}, \frac{1}{2}]$ for 1-shot settings and to $[\frac{1}{6}, \frac{5}{6}]$ for 5-shot settings. We set the probability threshold to 0.9 for the main experiments and discuss the performance under different values of this threshold in Section 4.4.1.

4.3.3 Results

Table 4.1 presents the 5-way 1-shot and 5-way 5-shot classification results of our methods on *mini*ImageNet and *tiered*ImageNet in comparision with previous FSL methods. Here all methods use ResNet12/ResNet18 architectures as feature extractors with input images of size 84×84. Thus, the comparison is fair. For the rest of this chapter, we denote our VAE trained with all data as **SVAE** (Semantic-VAE) and the model trained with only representative data as **R-SVAE** (Representative-SVAE).

We apply our methods on top of the Meta-Baseline [15], ProtoNet[137], and E3BM[72]. Our methods consistently improve all three baselines under all settings and for all datasets. They work particularly well under the 1-shot settings, in which sample bias is a more pronounced issue. Using the model trained on all data - SVAE, we report $6.8\% \sim 10\%$ 1-shot accuracy improvements for all three baselines. Our 1-shot performance for all the baselines outperforms the state-of-the-art method [124] by large margins. In 5-shot, our method consistently brings a $0.5 \sim 2.7\%$ performance gains to all baselines.

Using representative samples to train our VAE model further improves the three baseline methods under all settings and for all datasets. Com-

pared to SVAE, training on strictly representative data improves the 1-shot classification accuracy by 0.3% $\sim$ 2.8% and the 5-shot classification accuracy by 0.2% $\sim$ 0.8%. R-SVAE achieves state-of-the-art few-shot classification on *mini*ImageNet dataset with the ProtoNet baseline and on *tiered*ImageNet dataset with the E3BM baseline.

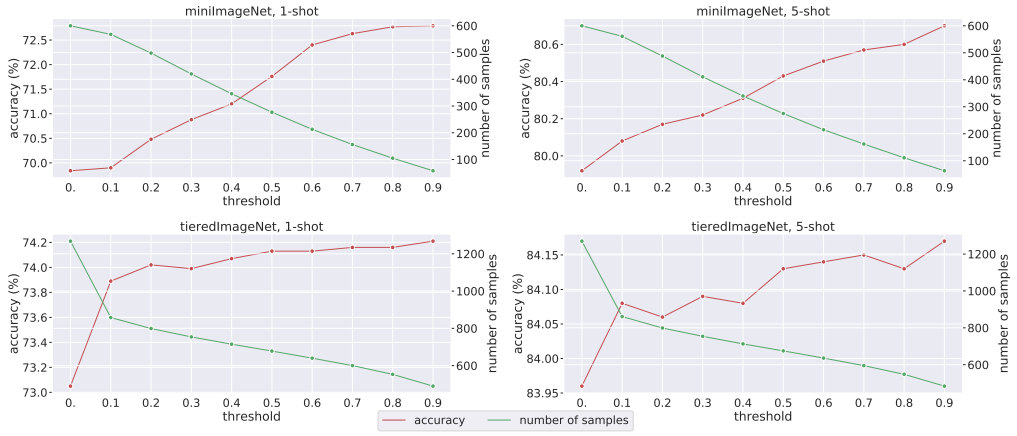


Figure 4.3: **Few-shot classification results with different probability thresholds.** We report the classification accuracy (%) (red) and the number of samples (green) when setting different thresholds for the probabilities. A higher threshold means we select samples that are more representative, resulting in a less amount of training data points. In general, the classification performance increases when the number of training samples decreases with increasing representativeness thresholds.

4.4 Analyses

All the following analyses use the feature extractor from the Meta-Baseline method [15].

4.4.1 Analysis on the Probability Threshold

In our main setting, we set a threshold of 0.9 on the probabilities to select those class-representative samples as the training data for our VAE model (the higher, the more representative). In this section, we conduct

Classifier	*mini*ImageNet			*tiered*ImageNet		
	support samples	+ SVAE	+ R-SVAE	support samples	+ SVAE	+R-SVAE
Prototype [15]	63.17 ± 0.23	69.96 ± 0.21	**72.79 ± 0.19**	68.62 ± 0.27	73.05 ± 0.24	**73.90 ± 0.24**
1-N-N	63.28 ± 0.23	67.25 ± 0.20	**69.27 ± 0.19**	68.73 ± 0.26	68.05 ± 0.25	**69.82 ± 0.24**
SVM	63.41 ± 0.23	70.30 ± 0.20	**72.84 ± 0.19**	68.88 ± 0.25	69.26 ± 0.25	**71.28 ± 0.24**
LR	63.33 ± 0.22	72.11 ± 0.20	**73.41 ± 0.19**	69.15 ± 0.25	74.99 ± 0.23	**75.98 ± 0.23**

Table 4.2: **Choices of the classifiers**. One-shot classification accuracy on *mini*ImageNet and *tiered*ImageNet using different types of classifiers, *i.e.*, 1-N-N, SVM and LR. All methods use the feature extractor from the Meta-Baseline method [15].

experiments with different threshold values to see how it affects the classifier's performance. Fig. 4.3 shows the classification accuracy under different thresholds on *mini*ImageNet and *tiered*ImageNet datasets. As the threshold increases, more non-representative samples are filtered out, resulting in less training data for R-SVAE. Interestingly, we observe that the model generally performs better with higher threshold values under both 1-shot and 5-shot settings. For example, under the 1-shot setting on *mini*ImageNet dataset, we only use 58 images per class on average when setting the threshold to 0.9. Training the VAE model with this small set of images improves the performance by 2.95% compared with the model trained using all data in the base set with 600 images per class on average. The results suggest that the performance of our method strongly corresponds to the representativeness of training data. Moreover, it shows that our sample selection method provides a reliable measurement for the representativeness of the training samples.

4.4.2 Performance with Different Classifiers

In our main experiments, we classify samples by finding the nearest neighbor among class prototypes. In this section, we apply another three different types of classifiers: 1-nearest neighbor classifier (1-N-N), Support Vector Machine (SVM), and Logistic Regression (LR).

Table 4.2 shows the 1-shot performance of different classifiers using our generated features on *mini*ImageNet and *tiered*ImageNet datasets. It shows that the features generated by our VAEs improve the performance of all three classifiers. For example, the 1-shot accuracy on *mini*ImageNet using LR is improved by 8.8% with SVAE and by 10.1% with R-SVAE. The consistent performance improvements show that our generated features

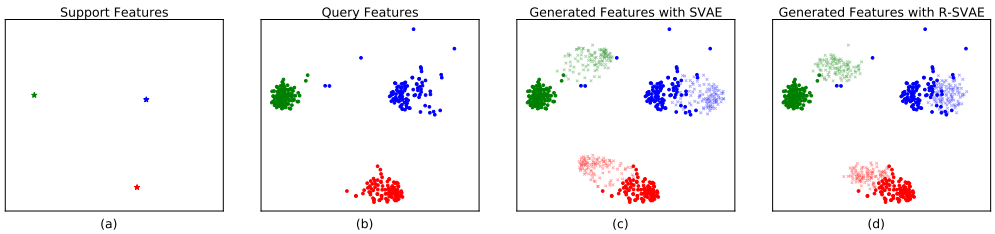


Figure 4.4: **Feature Visualization.** We show the t-SNE visualization of the original features (marked as dark points) and our generated features (marked as transparent points) on *tiered*ImageNet dataset. Different colors represent different classes. From left to right, we show the original support set (a), the query set (b), the features generated by SVAE (c), and the features generated by R-SVAE (d).

can benefit different types of classifiers.

4.4.3 Feature Distribution Analysis

In Fig. 4.4, we show the t-SNE representation [77] of different sets of features for three classes from the novel set of *tiered*ImageNet dataset. From left to right, we visualize the distribution of the original support set (a), the query set (b), the features generated by SVAE (c), and the features generated by R-SVAE (d). Note that our methods do not rely on the support features to generate features.

Fig. 4.4(c) and (d) visualize the effect of our sample selection method. Fig. 4.4(c) visualizes features generated from our method trained with all available data from the base classes, which consist of 1281 images per class on average. In Fig. 4.4(d), we train the same model with only 484 representative images per class on average. Our model trained with a representative subset of data generates features that lie closer to the real features, showing the effectiveness of our sample selection method.

Moreover, we plot the distance distributions between the estimated prototypes and the ground truth prototypes of each class. Specifically, for each class, we first obtain the ground-truth prototype by taking the mean of all the features of the class. Then we calculate the L_2 distance between the ground truth prototype and three different prototypes: 1) Baseline: the prototype was estimated using only the support samples. 2) SVAE:

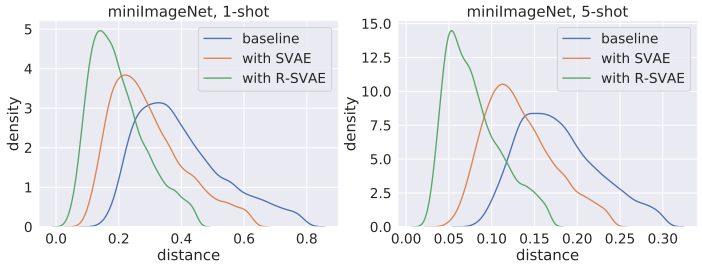


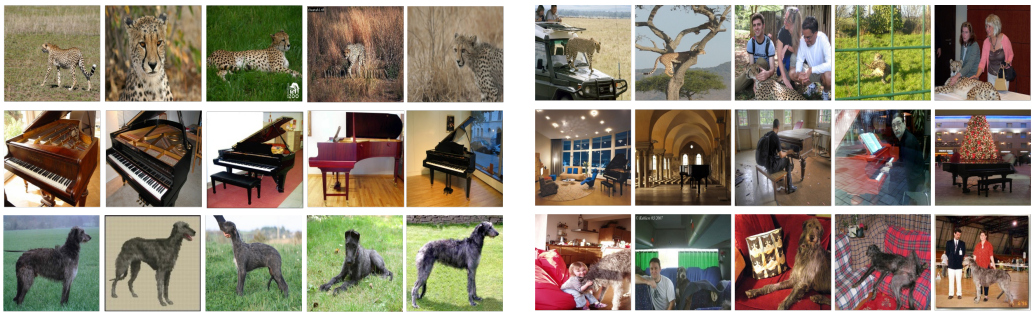
Figure 4.5: **Distance Distributions.** Kernel Density Estimation of the distance between the estimated prototypes and the ground truth prototype. A smaller value means the estimated prototypes are closer to the ground truth prototypes.

the prototype was estimated using the support samples and the generated samples from our SVAE model. 3) R-SVAE: the prototype was estimated using the support samples and the generated samples from our R-SVAE model.

We sample 2400 tasks from *mini*ImageNet dataset under both 5-way 1-shot and 5-way 5-shot settings. For each task, we obtain five distances, one distance per class. Then we plot the probability density distribution of the distance, shown in Fig. 4.5. The probability density is calculated by binning and counting observations and then smoothing them with a Gaussian kernel, namely, Kernel Density Estimation [16]. As can be seen the Fig., our estimated class prototypes are much closer to the ground truth prototypes, compared to the baseline.

4.4.4 Sample Visualization

In Fig. 4.6, we visualize some representative samples and non-representative samples based on the representativeness probability computed via our method. The samples on the left panel are images with high probabilities. These images mostly contain the main object of the category and are easy to recognize. On the contrary, the samples on the right panel are those with small probabilities. They contain various class-unrelated objects and can lead to noisy features for constructing class prototypes.



Representative Non-representative

Figure 4.6: **Examples of representative samples (left) and non-representative samples (right).** We visualize 5 images with high probabilities and 5 images with small probabilities computed via our proposed method for 3 classes from *tiered*ImageNet dataset.

	1-shot	5-shot
Meta-Baseline	63.17 ± 0.23	79.26 ± 0.17
Meta-Baseline + SVAE	67.39 ± 0.21	79.77 ± 0.17
Meta-Baseline + R-SVAE	**68.03 ± 0.22**	**79.93 ± 0.16**

Table 4.3: **Classification accuracy using Word2Vec[81] as the semantic feature extractor.**

4.4.5 Performance with Different Semantic Embedding

We use CLIP features in our main experiments. The performance of our method trained with Word2Vec[81] features are shown in Table 4.3. Note that CLIP model is trained with 400M pairs (image and its text title) collected from the web while Word2Vec is trained with only text data. Our model outperforms state-of-the-art methods in both cases.

4.5 Limitations and Discussion

We propose a feature generation method using a conditional VAE model. Here we focus on modeling the distribution of the representative samples rather than the whole data distribution. To accomplish that, we propose a sample selection method to collect a set of strictly representative training samples for training our VAE model. We show that our method brings con-

sistent performance improvements over multiple baselines and achieves state-of-the-art performance on both *mini*ImageNet and *tiered*ImageNet datasets. Our method requires a pre-trained NLP model to obtain the semantic embedding of each class. It might also inherit some potential biases from the textual domain. Note that our method does not aim to generate diverse data with large intra-class variance [68, 132]. Building a system that can generate both representative and non-representative samples can greatly benefit various downstream computer vision tasks and is an interesting direction to extend our work.

Chapter 5

Controllable Feature Generation for Few-shot Object Detection

5.1 Overview

Object detection plays a vital role in many computer vision systems. However, training a robust object detector often requires a large amount of training data with accurate bounding box annotations. Thus, there has been increasing attention on few-shot object detection (FSOD), which learns to detect novel object categories from just a few annotated training samples. It is particularly useful for problems where annotated data can be hard and costly to obtain such as rare medical conditions [86, 117], rare animal species [123], satellite images [9, 58], or failure cases in autonomous driving systems [78, 79, 98].

For the most part, state-of-the-art FSOD methods are built on top of a two-stage framework [97], which includes a region proposal network that generates multiple image crops from the input image and a classifier that labels these proposals. While the region proposal network generalizes well to novel classes, the classifier is more error-prone due to the lack of training data diversity [110]. To mitigate this issue, a natural approach is to generate additional features for novel classes [39, 145, 147]. For example, Zhang *et al* [145] propose a feature hallucination network to use the variation from base classes to diversify training data for novel classes. For zero-shot detection (ZSD), Zhu *et al* [147] propose to synthesize visual features for unseen objects based on a conditional variational auto-encoder.

Although much progress has been made, the lack of data diversity is still a challenging issue for FSOD methods.

In short, our contributions are:

- We propose the use of an adversarial critic to train a shadow remover from unpaired shadow and non-shadow patches, providing an alternative solution to the data dependency issue.
- We propose a set of physics-based constraints that define a transformation closely modelling shadow removal, which enables the training of shadow remover with only an adversarial training signal.
- Our shadow-removal system trained without any shadow-free images achieves competitive results compared to fully-supervised state-of-the-art methods on the ISTD dataset.
- We propose a method to obtain paired shadow data of complex shadows in complex scenes based on time-lapse videos. We introduced SBU-Timelapse, a video shadow removal dataset for evaluating shadow removal methods. We show that our shadow removal system can be fine-tuned for free to better remove shadows on testing videos.

Here we discuss a specific type of data diversity that greatly affects the accuracy of FSOD algorithms. Specifically, given a test image, the classifier needs to accurately classify multiple object proposals[1] that overlap the object instance in various ways. The features of these image crops exhibit great variability induced by different object scales, object parts included in the crops, object positions within the crops, and backgrounds. We observe a typical scenario where the state-of-the-art FSOD method, DeFRCN [90], only classifies correctly a few among many proposals overlapping an object instance of a few-shot class. In fact, different ways of cropping an object can result in features with various difficulty levels. An example is shown in Figure 6.1a where the image crop shown in the top row is classified correctly while another crop shown in the bottom row confuses the classifier due to some missing object parts. In general, the performance of the method on those hard cases is significantly worse than on easy cases (see section 5.4.4). However, building a classifier robust against

[1] Note that an RPN typically outputs 1000 object proposals per image.

crop-related variation is challenging since there are only a few images per few-shot class.

In this work, we propose a novel data generation method to mitigate this issue. Our goal is to generate features with diverse crop-related variations for the few-shot classes and use them as additional training data to train the classifier. Specifically, we aim to obtain a diverse set of features whose difficulty levels vary from easy to hard *w.r.t.* how the object is cropped.[2] To achieve this goal, we design our generative model such that it allows us to control the difficulty levels of the generated samples. Given a model that generates features from a latent space, our main idea is to enforce that the magnitude of the latent code linearly correlates with the difficulty level of the generated feature, *i.e.*, the latent code of a harder feature is placed further away from the origin and vice versa. In this way, we can control the difficulty level by simply changing the norm of the corresponding latent code.

In particular, our data generation model is based on a conditional variational autoencoder (VAE) architecture. The VAE consists of an encoder that maps the input to a latent representation and a decoder that reconstructs the input from this latent code. In our case, inputs to the VAE are object proposal features, extracted from a pre-trained object detector. The goal is to associate the norm (magnitude) of the latent code with the difficulty level of the object proposal. To do so, we rescale the latent code such that its norm linearly correlates with the Intersection-Over-Union (IoU) score of the input object proposal *w.r.t.* the ground-truth object box. This IoU score is a proxy that partially indicates the difficulty level: A high IoU score indicates that the object proposal significantly overlaps with the object instance while a low IoU score indicates a harder case where a part of the object can be missing. With this rescaling step, we can bias the decoder to generate harder samples by increasing the latent code magnitude and vice versa. In this work, we use latent codes with different norms varying from small to large to obtain a diverse set of features which can then serve as additional training data for the few-shot classifier.

To apply our model to FSOD, we first train our VAE model using abundant data from the base classes. The VAE is conditioned on the semantic code of the input instance category. After the VAE model is trained, we use the semantic embedding of the few-shot class as the conditional

[2]In this work, the difficulty level is strictly related to how the object is cropped.

code to synthesize new features for the corresponding class. In our experiments, we use our generated samples to fine-tune the baseline few-shot object detector - DeFRCN [90]. Surprisingly, a vanilla conditional VAE model trained with only ground-truth box features brings a 3.7% nAP50 improvement over the DeFRCN baseline in the 1-shot setting of the PASCAL VOC dataset [26]. Note that we are the first FSOD method using VAE-generated features to support the training of the classifier. Our proposed Norm-VAE can further improve this new state-of-the-art by another 2.1%, *i.e.*, from 60% to 62.1%. In general, the generated features from Norm-VAE consistently improve the state-of-the-art few-shot object detector [90] for both PASCAL VOC and MS COCO [67] datasets.

Our main contributions can be summarized as follows:

- We show that lack of crop-related diversity in training data of novel classes is a crucial problem for FSOD.
- We propose Norm-VAE, a novel VAE architecture that can effectively increase crop-related diversity in difficulty levels into the generated samples to support the training of FSOD classifiers.
- Our experiments show that the object detectors trained with our additional features achieve state-of-the-art FSOD in both PASCAL VOC and MS COCO datasets.

5.2 Proposed Method

In this section, we first review the problem setting of few-shot object detection and the conventional two-stage fine-tuning framework. Then we introduce our method that tackles few-shot object detection via generating features with increased crop-related diversity.

5.2.1 Preliminaries

In few-shot object detection, the training set is divided into a base set D^B with abundant annotated instances of classes C^B, and a novel set D^N with few-shot data of classes C^N, where C^B and C^N are non-overlapping. For a sample $(x, y) \in D^B \cup D^N$, x is the input image and $y = \{(c_i, b_i), i = 1, ..., n\}$ denotes the categories $c \in C^B \cup C^N$ and bounding box coordinates b of the n object instances in the image x. The number of objects for each class in

C^N is K for K-shot detection. We aim to obtain a few-shot detection model with the ability to detect objects in the test set with classes in $C^B \cup C^N$.

Recently, two-stage fine-tuning methods have shown great potential in improving few-shot detection. In these two-stage detection frameworks, a Region Proposal Network (RPN) takes the output feature maps from a backbone feature extractor as inputs and generates region proposals. A Region-of-Interest (RoI) head feature extractor first pools the region proposals to a fixed size and then encodes them as vector embeddings, known as the RoI features. A classifier is trained on top of the RoI features to classify the categories of the region proposals.

The fine-tuning often follows a simple two-stage training pipeline, *i.e.*, the data-abundant base training stage and the novel fine-tuning stage. In the base training stage, the model collects transferable knowledge across a large base set with sufficient annotated data. Then in the fine-tuning stage, it performs quick adaptation on the novel classes with limited data. Our method aims to generate features with diverse crop-related variations to enrich the training data for the classifier head during the fine-tuning stage. In our experiments, we show that our generated features significantly improve the performance of DeFRCN [90].

5.2.2 Overall Pipeline

Figure 5.1 summarizes the main idea of our proposed VAE model. For each input object crop, we first use a pre-trained object detector to obtain its RoI feature. The encoder takes as input the RoI feature and the semantic embedding of the input class to output a latent code z. We then transform z such that its norm linearly correlates with the IoU score of the input object crop *w.r.t.* the ground-truth box. The new norm is the output of a simple linear function $g(\cdot)$ taking the IoU score as the single input. The decoder takes as input the new latent code and the class semantic embedding to output the reconstructed feature. Once the VAE is trained, we use the semantic embedding of the few-shot class as the conditional code to synthesize new features for the class. To ensure the diversity *w.r.t.* object crop in generated samples, we vary the norm of the latent code when generating features. The generated features are then used together with the few-shot samples to fine-tune the object detector.

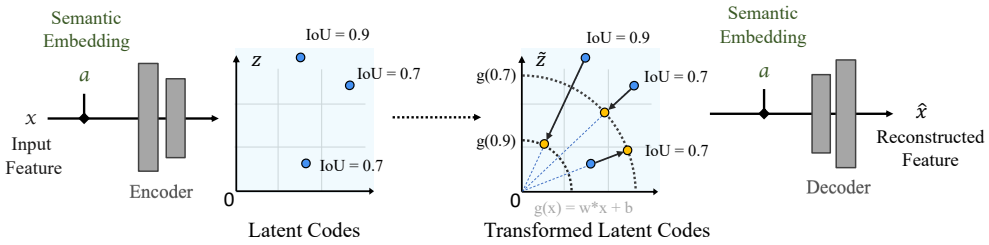


Figure 5.1: **Norm-VAE for modelling crop-related variations.** The original latent code z is rescaled to $\hat{z}$ such that the norm of $\hat{z}$ linearly correlates with the IoU score of the input crop (*w.r.t.* the ground truth box). The original latent codes are colored in **blue** while the rescaled ones are colored in **yellow**. The norm of the new latent code is the output of a simple linear function $g(\cdot)$ taking the IoU score as the single input. As can be seen, the two points whose IoU = 0.7 are both rescaled to norm $g(0.7)$ while another point whose IoU = 0.9 is mapped to norm $g(0.9)$. As a result, different latent norms represent different crop-related variations, enabling diverse feature generation.

Norm-VAE for Feature Generation

We develop our feature generator based on a conditional VAE architecture [108]. Given an input object crop, we first obtain its Region-of-Interest (RoI) feature f via a pre-trained object detector. The RoI feature f is the input for the VAE. The VAE is composed of an Encoder $E(f,a)$, which maps a visual feature f to a latent code z, and a decoder $G(z,a)$ which reconstructs the feature f from z. Both E and G are conditioned on the class semantic embedding a. We obtain this class semantic embedding a by inputting the class name into a semantic model [82, 92]. It contains class-specific information and serves as a controller to determine the categories of the generated samples. Conditioning on these semantic embeddings allows reliably generating features for the novel classes based on the learned information from the base classes [131]. Here we assume that the class names of both base and novel classes are available and we can obtain the semantic embedding of all classes.

We first start from a vanilla conditional VAE model. The loss function

for training this VAE for a feature f_i of class j can be defined as:

$$L_V(f_i) = \mathrm{KL}\left(q(z_i|f_i, a^j)\|p(z|a^j)\right) - \mathrm{E}_{q(z_i|f_i,a^j)}[\log p(f_i|z_i, a^j)], \tag{5.1}$$

where a^j is the semantic embedding of class j. The first term is the Kullback-Leibler divergence between the VAE posterior $q(z|f,a)$ and a prior distribution $p(z|a)$. The second term is the decoder's reconstruction error. $q(z|f,a)$ is modeled as $E(f,a)$ and $p(f|z,a)$ is equal to $G(z,a)$. The prior distribution is assumed to be $\mathcal{N}(0,I)$ for all classes.

The goal is to control the crop-related variation in a generated sample. Thus, we establish a direct correspondence between the latent norm and the crop-related variation. To accomplish this, we transform the latent code such that its norm correlates with the IoU score of the input crop. Given an input RoI feature f_i of a region with an IoU score s_i, we first input this RoI feature to the encoder to obtain its latent code z_i. We then transform z_i to $\tilde{z_i}$ such that the norm of $\tilde{z_i}$ correlates to s_i. The new latent code $\tilde{z_i}$ is the output of the transformation function $\mathcal{T}(\cdot,\cdot)$:

$$\tilde{z_i} = \mathcal{T}(z_i, s_i) = \frac{z_i}{\|z_i\|} * g(s_i), \tag{5.2}$$

where $\|z_i\|$ is the L_2 norm of z_i, s_i is the IoU score of the input proposal *w.r.t.* its ground-truth object box, and $g(\cdot)$ is a simple pre-defined linear function that maps an IoU score to a norm value. With this new transformation step, the loss function of the VAE from equation 5.1 for an input feature f_i from class j with an IoU score s_i thus can be rewritten as:

$$L_V(f_i, s_i) = \mathrm{KL}\left(q(z_i|f_i, a^j)\|p(z|a^j)\right) - \mathrm{E}_{q(z_i|f_i,a^j)}\left[\log p(f_i|\mathcal{T}(z_i, s_i), a^j)\right]. \tag{5.3}$$

Generating Diverse Data for Improving Few-shot Object Detection

After the VAE is trained on the base set, we generate a set of features with the trained decoder. Given a class y with a semantic vector a^y and a noise vector z, we generate a set of augmented features $\mathbb{G}^y$:

$$\mathbb{G}^y = \{\hat{f}|\hat{f} = G(\frac{z}{\|z\|} * \beta, a^y)\}, \tag{5.4}$$

where we vary β to obtain generated features with more crop-related variations. The value range of β is chosen based on the mapping function $g(\cdot)$. The augmented features are used together with the few-shot samples to fine-tune the object detector. We fine-tune the whole system using an additional classification loss computed on the generated features together with the original losses computed on real images. This is much simpler than the previous method of [145] where they fine-tune their system via an EM-like (expectation-maximization) manner.

5.3 Experiments

5.3.1 Datasets and Evaluation Protocols

We conduct experiments on both PASCAL VOC (07 + 12) [26] and MS COCO datasets [67]. For fair comparison, we follow the data split construction and evaluation protocol used in previous works [45]. The PASCAL VOC dataset contains 20 categories. We use the same 3 base/novel splits with TFA [118] and refer them as Novel Split 1,2, 3. Each split contains 15 base classes and 5 novel classes. Each novel class has K annotated instances, where $K = 1, 2, 3, 5, 10$. We report AP50 of the novel categories (nAP50) on VOC07 test set. For MS COCO, the 60 categories disjoint with PASCAL VOC are used as base classes while the remaining 20 classes are used as novel classes. We evaluate our method on shot 1,2,3,5,10,30 and COCO-style AP of the novel classes is adopted as the evaluation metrics.

5.3.2 Implementation Details

Feature generation methods like ours in theory can be built on top of many few-shot object detectors. In our experiments, we use the pre-trained Faster-RCNN [97] with ResNet-101 [41] following previous work DeFRCN [90]. The dimension of the extracted RoI feature is 2048. For our feature generation model, the encoder consists of three fully-connected (FC) layers and the decoder consists of two FC layers, both with 4096 hidden units. LeakyReLU and ReLU are the non-linear activation functions in the hidden and output layers, respectively. The dimensions of the latent space and the semantic vector are both set to be 512. Our semantic embeddings are extracted from a pre-trained CLIP [92] model in all main experiments.

An additional experiment using Word2Vec [81] embeddings is reported in Section 5.4.2. After the VAE is trained on the base set with various augmented object boxes , we use the trained decoder to generate $k = 30$ features per class and incorporate them into the fine-tuning stage of the DeFRCN model. We set the function $g(\cdot)$ in Equation 5.2 to a simple linear function $g(x) = w * x + b$ which maps an input IoU score x to the norm of the new latent code. Note that x is in range $[0.5, 1]$ and the norm of the latent code of our VAE before the rescaling typically centers around $\sqrt{512}$ (512 is the dimension of the latent code). We empirically choose $g(\cdot)$ such that the new norm ranges from $\sqrt{512}$ to $5 * \sqrt{512}$. We provide further analyses on the choice of $g(\cdot)$ in the supplementary material. For each feature generation iteration, we gradually increase the value of the controlling parameter β in Equation 5.4 with an interval of 0.75.

5.3.3 Few-shot Detection Results

	Novel Split 1					Novel Split 2					Novel Split 3				
Method	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
TFA w/ fc [118]	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
TFA w/ cos [118]	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
MPSR [128]	41.7	-	51.4	55.2	61.8	24.4	-	39.2	35.1	39.9	47.8	-	42.3	48.0	49.7
FsDetView [129]	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
FSCE [110]	44.2	43.8	51.4	61.9	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5
CME [60]	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5
SRR-FSD [146]	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
Halluc. [145]	45.1	44.0	44.7	55.0	55.9	23.2	27.5	35.1	34.9	39.0	30.5	35.1	41.4	49.0	49.3
FSOD-MC [29]	40.1	44.2	51.2	62.0	63.0	33.3	33.1	42.3	46.3	52.3	36.1	43.1	43.5	52.0	56.0
FADI [11]	50.3	54.8	54.2	59.3	63.2	30.6	35.0	40.3	42.8	48.0	45.7	49.7	49.1	48.3	51.5
CoCo-RCNN [74]	43.9	44.5	53.1	64.6	65.5	29.4	31.3	43.8	44.3	51.8	39.1	43.9	47.2	54.7	60.3
MRSN [76]	47.6	48.6	57.8	61.9	62.6	31.2	38.3	46.7	47.1	50.6	35.5	30.9	45.6	54.4	57.4
FCT [37]	49.9	57.1	57.9	63.2	67.1	27.6	34.5	43.7	49.2	51.2	39.5	54.7	52.3	57.0	58.7
Pseudo-Labelling [46]	54.5	53.2	58.8	63.2	65.7	32.8	29.2	50.7	49.8	50.6	48.4	52.7	55.0	59.6	59.6
DeFRCN [90]	56.3	60.3	62.0	67.0	66.1	35.7	45.2	51.5	54.1	53.3	54.5	55.6	56.6	60.8	62.7
Vanila-VAE (Ours)	60.0	63.3	66.3	68.3	67.1	39.3	46.2	52.7	53.5	53.4	56.0	58.8	57.1	62.6	63.6
Norm-VAE (Ours)	**62.1**	**64.9**	**67.8**	**69.2**	**67.5**	**39.9**	**46.8**	**54.4**	**54.2**	**53.6**	**58.2**	**60.3**	**61.0**	**64.0**	**65.5**

Table 5.1: **Few-shot object detection performance (nAP50) on PASCAL VOC dataset**. We evaluate the performance on three different splits. Our method consistently improves upon the baseline for all three splits across all shots. Best performance in bold.

We use the generated features from our VAE model together with the few-shot samples to fine-tune DeFRCN. We report the performance of two models: "Vanilla-VAE" denotes the performance of the model trained

Method	nAP						nAP75					
	1	2	3	5	10	30	1	2	3	5	10	30
TFA w/ fc [118]	2.9	4.3	6.7	8.4	10.0	13.4	2.8	4.1	6.6	8.4	9.2	13.2
TFA w/ cos [118]	3.4	4.6	6.6	8.3	10.0	13.7	3.8	4.8	6.5	8.0	9.3	13.2
MPSR [128]	2.3	3.5	5.2	6.7	9.8	14.1	2.3	3.4	5.1	6.4	9.7	14.2
FADI [11]	5.7	7.0	8.6	10.1	12.2	16.1	6.0	7.0	8.3	9.7	11.9	15.8
FCT [37]	-	7.9	-	-	17.1	21.4	-	7.9	-	-	17.0	22.1
Pseudo-Labelling [46] †	-	-	-	-	17.8	**24.5**	-	-	-	-	**17.8**	**25.0**
DeFRCN [90]	6.6	11.7	13.3	15.6	18.7	22.4	7.0	12.2	13.6	15.1	17.6	22.2
Vanilla-VAE (ours)	8.8	13.0	14.1	**15.9**	**18.7**	22.5	7.9	12.5	13.4	15.1	17.6	22.2
Norm-VAE (ours)	**9.5**	**13.7**	**14.3**	**15.9**	**18.7**	22.5	**8.8**	**13.7**	**14.2**	**15.3**	**17.8**	22.4

Table 5.2: **Few-shot detection performance for the novel classes on MS COCO dataset**. Our approach outperforms baseline methods in most cases, especially in low-shot settings ($K < 10$). † applies mosaic data augmentation introduced in [8] during fine-tuning. Best performance in bold.

with generated features from a vanilla VAE trained on the base set of ground-truth bounding boxes and "Norm-VAE" denotes the performance of the model trained with features generated from our proposed Norm-VAE model.

PASCAL VOC Table 5.1 shows our results for all three random novel splits from PASCAL VOC. Simply using a VAE model trained with the original data outperforms the state-of-the-art method DeFRCN in all shot and split on PASCAL VOC benchmark. In particular, vanilla-VAE improves DeFRCN by 3.7% for 1-shot and 4.3% for 3-shot on Novel Split 1. Using additional data from our proposed Norm-VAE model consistently improves the results across all settings. We provide qualitative examples in the supplementary material.

MS COCO Table 5.2 shows the FSOD results on MS COCO dataset. Our generated features bring significant improvements in most cases, especially in low-shot settings ($K \leq 10$). For example, Norm-VAE brings a 2.9% and a 2.0% nAP improvement over DeFRCN in 1-shot and 2-shot settings, respectively. Pseudo-Labeling is better than our method in higher shot settings. However, they apply mosaic data augmentation [8] during fine-tuning.

	Data	1-shot	2-shot	3-shot
DeFRCN [90]	-	56.3	60.3	62.0
VAE	Orginal	60.0	63.3	66.3
VAE	Augmented	60.1	62.7	66.4
Norm-VAE	Augmented	**62.1**	**64.9**	**67.8**

Table 5.3: **Performance comparisons between vanilla VAE and Norm-VAE on PASCAL VOC dataset**. Training a the vanilla VAE with the augmented data does not bring performance improvement. One possible reason is that the generated samples are not guaranteed to be diverse even with sufficient data.

5.4 Analyses

5.4.1 Effectiveness of Norm-VAE

We compare the performance of Norm-VAE with a baseline vanilla VAE model that is trained with the same set of augmented data. As shown in Table 5.3, using the vanilla VAE with more training data does not bring performance improvement compared to the VAE model trained with the base set. This suggests that training with more diverse data does not guarantee diversity in generated samples *w.r.t.* a specific property. Our method, by contrast, improves the baseline model by 1.3% $\sim$ 1.9%, which demonstrates the effectiveness of our proposed Norm-VAE.

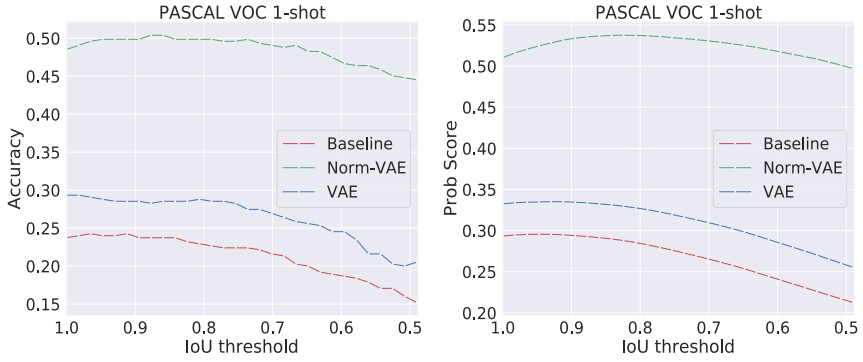
Method	Semantic Embedding	Novel Split 1			Novel Split 2			Novel Split 3		
		1-shot	2-shot	3-shot	1-shot	2-shot	3-shot	1-shot	2-shot	3-shot
DeFRCN [90]	-	56.3	60.3	62.0	35.7	45.2	51.5	54.5	55.6	56.6
Vanilla VAE	Word2Vec	60.4	62.9	**66.7**	38.7	45.2	52.9	55.6	58.7	57.9
Norm-VAE	Word2Vec	**61.6**	**63.4**	66.3	**40.7**	**46.4**	**53.3**	**56.8**	**59.0**	**60.2**
Vanilla VAE	CLIP	60.0	63.3	66.3	39.3	46.2	52.7	56.0	58.8	57.1
Norm-VAE	CLIP	**62.1**	**64.9**	**67.8**	**39.9**	**46.8**	**54.4**	**58.2**	**60.3**	**61.0**

Table 5.4: **FSOD Performance of VAE models trained with different class semantic embeddings**. CLIP [92] is trained with 400M pairs (image and its text title) collected from the web while Word2Vec [81] is trained with only text data.

5.4.2 Performance Using Different Semantic Embeddings

We use CLIP [92] features in our main experiments. In Table 5.4, we compare this model with another model trained with Word2Vec [81] on PASCAL VOC dataset. Note that CLIP model is trained with 400M pairs (image and its text title) collected from the web while Word2Vec is trained with only text data. Our Norm-VAE trained with Word2Vec embedding achieves similar performance to the model trained with CLIP embedding. In both cases, the model outperform the state-of-the-art FSOD method in all settings.

5.4.3 Robustness against Inaccurate Localization



(a) Accuracy (b) Probability score

Figure 5.2: **Classification accuracy and probability score of the object detector on the augmented box**. We compare between the baseline DeFRCN [90], the model trained with features from vanilla VAE and our proposed Norm-VAE. By generating features with diverse crop-related variations, we increase the object detector's robustness against inaccurate object box localization.

In this section, we conduct experiments to show that our object detector trained with features with diverse crop-related variation is more robust against inaccurate bounding box localization. Specifically, we randomly select 1000 testing instances from PASCAL VOC test set and create 30 augmented boxes for each ground-truth box. Each augmented box is created by enlarging the ground-truth boxes by x% for each dimension

where x ranges from 0 to 30. The result is summarized in Figure 5.2 where "Baseline" denotes the performance of DeFRCN[90], "VAE" is the performance of the model trained with features generated from a vanilla VAE, and "Norm-VAE" is the model trained with generated features from our proposed model.

Figure 5.2 (a) shows the classification accuracy of the object detector on the augmented box as the IoU score between the augmented bounding box and the ground-truth box decreases. For both the baseline method DeFRCN and the model trained with features from a vanilla VAE, the accuracy drops by $\sim 10\%$ as the IoU score decreases from 1.0 to 0.5. These results suggest that these models perform much better for boxes that have higher IoU score *w.r.t.* the ground-truth boxes. Our proposed method has higher robustness to these inaccurate boxes: the accuracy of the model trained with features from Norm-VAE only drops by $\sim 5\%$ when IoU score decreases from 1 to 0.5.

Figure 5.2 (b) plots the average probability score of the classifier on the ground-truth category as the IoU score decreases. Similarly, the probability score of both baseline DeFRCN and the model trained with features from a vanilla VAE drops around 0.08 as the IoU score decreases from 1.0 to 0.5. The model trained with features from Norm-VAE, in comparison, has more stable probability score as the IoU threshold decreases.

Method	1-shot	2-shot	3-shot
DeFRCN[90]	16.6	13.3	15.2
Ours (↑ Improvement)	18.8 (↑2.2)	16.4 (↑ 3.1)	19.2 (↑4.0)

Table 5.5: **AP50~75 of our method and DeFRCN on PASCAL VOC dataset**. AP 50~75 refers to the average precision computed on the proposals with the IoU thresholds between 50% and 75% and discard the proposals with IoU scores larger than 0.75, i.e., only *"hard"* cases.

5.4.4 Performance on Hard Cases

In Table 5.5, we show AP 50~75 of our method on PASCAL VOC dataset (Novel Split 1) in comparison with the state-of-the-art method DeFRCN. Here AP 50~75 refers to the average precision computed on the proposals

with the IoU thresholds between 50% and 75% and discard the proposals with IoU scores (*w.r.t.* the ground-truth box) larger than 0.75. Thus, AP 50∼75 implies the performance of the model in *"hard"* cases where the proposals do not significantly overlap the ground-truth object boxes. In this extreme test, the performance of both models are worse than their AP50 counterparts (Table 5.1), showing that FSOD methods are generally not robust to those hard cases. Our method mitigates this issue, outperforming DeFRCN by substantial margins. However, the performance is still far from perfect. Addressing these challenging cases is a fruitful venue for future FSOD work.

5.5 Conclusion

We tackle the lack of crop-related variability in the training data of FSOD, which makes the model not robust to different object proposals of the same object instance. To this end, we propose a novel VAE model that can generate features with increased crop-related diversity. Experiments show that such increased diversity in the generated samples significantly improves the current state-of-the-art FSOD performance for both PASCAL VOC and MS COCO datasets. Our proposed VAE model is simple, easy to implement, and allows modifying the difficulty levels of the generated samples. In general, generative models whose outputs can be manipulated according to different properties, are crucial to various frameworks and applications. In future work, we plan to address the following limitations of our work: 1) We bias the decoder to increase the diversity in generated samples instead of explicitly enforcing it. 2) Our proposed method is designed to generate visual features of object boxes for FSOD. Generating images might be required in other applications. Another direction to extend our work is to represent other variational factors in the embedding space to effectively diversify generated data.

Chapter 6

Summary and Future Work

This thesis proposal investigates how to apply feature augmentation and generation methods for few-shot learning tasks, *i.e.*, few-shot classification and few-shot object detection. Chapter 3 introduces a variational feature disentangling method for fine-grained few-shot classification. Specifically, we decompose features into two components, *i.e.*, intra-class variance features and class-discriminative features. We model the intra-class variance features via a common distribution, from which we can sample the intra-class variations to diversify a specific instance. In this way, the generated features enlarge the intra-class variance while preserving the class-discriminative features, which can benefit fine-grained few-shot learning.

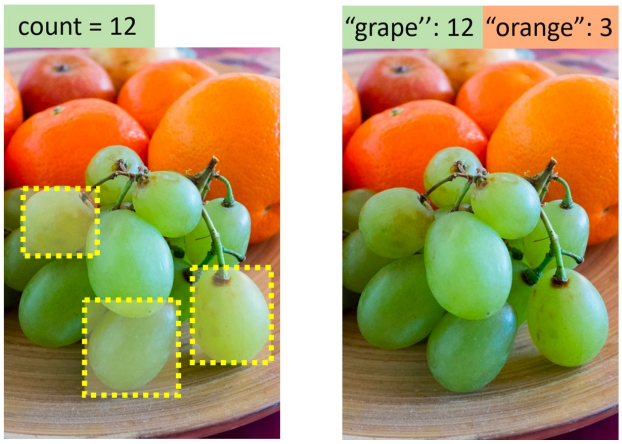
In Chapter 4, we introduce a VAE-based feature generation method focusing on generating samples that faithfully reflect the key characteristics of the category for the few-shot classification task. To do this, we propose a sample selection method to collect a set of strictly representative training samples for training the VAE model. This training scheme effectively enhances the representativeness of the generated samples and therefore, improves the few-shot classification results.

In Chapter 5, we present a VAE-based feature generation method for few-shot object detection. We first show that the lack of crop-related diversity in few-shot training data is a crucial problem. To resolve this issue, we propose a novel VAE architecture that can effectively increase crop-related diversity in difficulty levels in the generated samples to support the training of FSOD classifiers. We show the increased diversity in the generated samples significantly improves the current state-of-the-art FSOD perfor-

mance.

6.1 Future Work

We have explored the idea of feature augmentation and generation for a series of computer vision tasks in few-shot learning. For future work, we target to apply this idea to class-agnostic object counting. To relax the dependency of labeled samples, we propose the task of zero-shot object counting (ZSC). Unlike traditional few-shot counting methods that require a few human-annotated exemplars as inputs, in ZSC, the counting model only needs the class name to count the number of object instances in the image (as shown in Figure 6.1). This is a useful setting since having humans in the loop is not practical for many real-world applications, such as fully automated wildlife monitoring systems or visual anomaly detection systems.



(a) Few-shot Counting (b) Zero-Shot Counting

Figure 6.1: Our proposed task of zero-shot object counting (ZSC). Traditional few-shot counting methods require a few exemplars of the object category (a). We propose zero-shot counting where the counter only needs the class name to count the number of object instances. (b). Few-shot counting methods require human annotators at test time while zero-shot counters can be fully automatic.

One straightforward solution for this task is to utilize a generative model to generate a visual prototype based on the given class name. This

generated prototype can resemble object exemplar features in traditional exemplar-based object counting. Similar idea has been discussed in Chapter 4 and Chapter 5, where we use VAE to generate features conditioned on semantic embeddings for the task of few-shot classification and few-shot object detection respectively.

In particular, we use the MS-COCO detection to train a VAE model conditioned on the semantic embeddings extracted from CLIP [92]. Given a previously unseen class for counting, we first generate a set of features by inputting the respective semantic vector to the decoder of the VAE model, we then take the mean of all the generated features to obtain the class prototype. The generated prototype will be used to do correlation matching with the features of input images to get the similarity map. A pre-trained exemplar-based counting model takes the similarity map as input to get the density map and final count.

We implement the above method on a recent class-agnostic object counting dataset, FSC-147 [94]. The Mean Average Errors (MAE) on the validation set and the test set are 48.56 and 41.33 respectively, which is a significant drop compared to the previous exemplar-based counting methods. The potential explanation for this drop is that the same prototype is applied to all the objects from different images. However, these objects typically exhibit large variations. One possible direction to improve the performance is to select different exemplars dynamically according to the input image, which we leave as future work.

Chapter 7

Bibliography

[1] https://www.hillspet.com/dog-care/dog-breeds/yorkshire-terrier. URL https://www.hillspet.com/dog-care/dog-breeds/yorkshire-terrier. 27

[2] A. Alessandro and S. Stefano. Emergence of invariance and disentanglement in deep representations. In *J. Mach. Learn. Res.*, 2018. 14

[3] A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial eetworks. In *arXiv preprint arXiv:1711.04340*, 2018. 6, 9

[4] G. Arora, V. K. Verma, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 8, 28

[5] J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. 8, 27

[6] N. Bendre, K. Desai, and P. Najafirad. Generalized zero-shot learning using multimodal variational auto-encoder with semantic concepts. *ArXiv*, abs/2106.14082, 2021. 8

[7] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi. Meta-learning

with differentiable closed-form solvers. *ArXiv*, abs/1805.08136, 2019. 18

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020. 51

[9] A. Borowicz, H. Le, G. Humphries, G. Nehls, C. Höschle, V. Kosarev, and H. Lynch. Aerial-trained deep learning networks for surveying cetaceans from satellite imagery. *PLoS ONE*, 14, 2019. 42

[10] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae. In *arXiv: Machine Learning*, 2018. 14

[11] Y. Cao, J. Wang, Y. Jin, T. Wu, K. Chen, Z. Liu, and D. Lin. Few-shot object detection via association and discrimination. In *NeurIPS*, 2021. 7, 50, 51

[12] Y. chao Gu, L. Zhang, Y. Liu, S.-P. Lu, and M.-M. Cheng. Generalized zero-shot learning via vae-conditioned generative flow. *ArXiv*, abs/2009.00303, 2020. 8

[13] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *arXiv preprint arXiv:1802.04942*, 2018. 14

[14] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang. A closer look at few-shot classification. In *International Conference on Machine Learning(ICML)*, 2019. 17, 18

[15] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *ICCV*, 2021. 26, 28, 30, 33, 34, 35, 36, 37

[16] Y.-C. Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1:161 – 187, 2017. 39

[17] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, and M. Hebert. Image deformation meta-networks for one-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8672–8681, 2019. 1

[18] W.-H. Chu, Y.-J. Li, J.-C. Chang, and Y. Wang. Spot and learn: A maximum-entropy patch sampler for few-shot image classification. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6244–6253, 2019. 6

[19] D. L. Davies and D. W. Bouldin. A cluster separation measure. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1979. 23

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 34

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 26

[22] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. In *arXiv preprint arXiv:1611.02648*, 2017. 21, 22

[23] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao. Conditional variational image deraining. *IEEE Transactions on Image Processing*, 29: 6288–6301, 2020. 8

[24] N. Dvornik, C. Schmid, and J. Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 33

[25] P. Esser, E. Sutter, and B. Ommer. A variational u-net for conditional appearance and shape generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 8

[26] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 2009. 45, 49

[27] Q. Fan, W. Zhuo, and Y.-W. Tai. Few-shot object detection with attention-rpn and multi-relation detector. *CVPR*, pages 4012–4021, 2020. 6

[28] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *CVPR*, June 2020. 6

[29] Q. Fan, C.-K. Tang, and Y.-W. Tai. Few-shot object detection with model calibration. In *ECCV*, 2022. 50

[30] Z. Fan, Y. Ma, Z. Li, and J. Sun. Generalized few-shot object detection without forgetting. In *CVPR*, June 2021. 6, 7

[31] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning(ICML)*, 2017. 5, 18, 19, 33

[32] H. Gao, Z. Shou, A. Zareian, H. Zhang, and S.-F. Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 6, 9

[33] K. Guirguis, A. Hendawy, G. Eskandar, M. Abdelsamad, M. Kayser, and J. Beyerer. Cfa: Constraint-based finetuning approach for generalized few-shot object detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4038–4048, 2022. 7

[34] J. Guo and S. Guo. A novel perspective to zero-shot learning: Towards an alignment of manifold structures via semantic feature expansion. *IEEE Transactions on Multimedia*, 23:524–537, 2021. 8, 27

[35] X. Guo, Y. Du, and L. Zhao. Property controllable variational autoencoder via and invertible mutual dependence. In *ICLR*, 2021. 7, 8

[36] G. Han, Y. He, S. Huang, J. Ma, and S.-F. Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *ICCV*, October 2021. 6

[37] G. Han, J. Ma, S. Huang, L. Chen, and S.-F. Chang. Few-shot object detection with fully cross-transformer. In *CVPR*, pages 5321–5330, 2022. 6, 50, 51

[38] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision (ICCV)*, June 2017. 10

[39] N. Hayat, M. Hayat, S. Rahman, S. H. Khan, S. W. Zamir, and F. S. Khan. Synthesizing the unseen for zero-shot object detection. In *ACCV*, 2020. 42

[40] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 35

[41] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 17, 49

[42] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 7

[43] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen. Cross attention network for few-shot classification. In *NeurIPS*, 2019. 33

[44] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. In *arXiv preprint arXiv:1908.01313*, 2019. 19

[45] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell. Few-shot object detection via feature reweighting. *ICCV*, 2019. 6, 7, 49

[46] P. Kaul, W. Xie, and A. Zisserman. Label, verify, correct: A simple few-shot object detection method. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 6, 7, 50, 51

[47] R. Keshari, R. Singh, and M. Vatsa. Generalized zero-shot learning via over-complete distribution. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13297–13305, 2020. 8

[48] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization(FGVC), IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2011. 9, 10, 16, 18, 19

[49] J. Kim, T.-H. Oh, S. Lee, F. Pan, and I. S. Kweon. Variational prototyping-encoder: One-shot learning with prototypical images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8, 26

[50] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 17

[51] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 10

[52] J. Klys, J. Snell, and R. S. Zemel. Learning latent subspaces in variational autoencoders. In *NeurIPS*, 2018. 7, 8

[53] H. Le and D. Samaras. Shadow removal via shadow image decomposition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 9

[54] H. Le and D. Samaras. From shadow segmentation to shadow removal. In *European Conference on Computer Vision(ECCV)*, 2020. 9

[55] H. Le and D. Samaras. Physics-based shadow image decomposition for shadow removal. In *arXiv preprint arXiv:2012.13018*, 2020. 9

[56] H. Le, C.-P. Yu, G. Zelinsky, and D. Samaras. Co-localization with category-consistent features and geodesic distance propagation. In *ICCV Workshop*, 2017. 6

[57] H. Le, T. F. Y. Vicente, V. Nguyen, M. Hoai, and D. Samaras. A+D Net: Training a shadow detector with adversarial shadow attenuation. In *European Conference on Computer Vision(ECCV)*, 2018. 9

[58] H. Le, B. Goncalves, D. Samaras, and H. Lynch. Weakly labeling the antarctic: The penguin colony case. In *CVPR Workshops*, June 2019. 42

[59] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 6, 10, 18, 33

[60] B. Li, B. Yang, C. Liu, F. Liu, R. Ji, and Q. Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 50

[61] K. Li, Y. Zhang, K. Li, and Y. Fu. Adversarial feature hallucination networks for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 9

[62] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales. A simple feature augmentation for domain generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[63] W. Li, L. Wan, J. Xu, J. Huo, Y. Gao, and J. Luo. Revisiting local descriptor based image-toclass measure for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 19

[64] Y. Li, H. Zhu, Y. Cheng, W. Wang, C. S. Teo, C. Xiang, P. Vadakkepat, and T. H. Lee. Few-shot object detection via classification refinement and distractor retreatment. In *CVPR*, June 2021. 6

[65] Z. Li, F. Zhou, F. Chen, and H. Li. Meta-sgd: Learning to learn quickly for few-shot learning. In *arXiv preprint arXiv:1707.09835*, 2017. 5, 6

[66] Y. Lifchitz, Y. Avrithis, S. Picard, and A. Bursuc. Dense classification and implanting for few-shot learning. pages 9250–9259, 2019. 5

[67] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 45, 49

[68] X. Lin, Y. Duan, Q. Dong, J. Lu, and J. Zhou. Deep variational metric learning. In *European Conference on Computer Vision(ECCV)*, 2018. 13, 41

[69] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision(ECCV)*, 2020. 33

[70] J. Liu, L. Song, and Y. Qin. Prototype rectification for few-shot learning. In *European Conference on Computer Vision(ECCV)*, 2020. 26

[71] M.-Y. Liu, T. M. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 8

[72] Y. Liu, B. Schiele, and Q. Sun. An ensemble of epoch-wise empirical bayes for few-shot learning. In *European Conference on Computer Vision(ECCV)*, 2020. 26, 28, 30, 33, 34, 35

[73] W. Luo, X. Yang, X. Mo, Y. Lu, L. S. Davis, J. Li, J. Yang, and S.-N. Lim. Cross-x learning for fine-grained visual categorization. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 19

[74] J. Ma, G. Han, S. Huang, Y. Yang, and S.-F. Chang. Few-shot end-to-end object detection via constantly concentrated encoding across heads. In *ECCV*, 2022. 6, 50

[75] P. Ma and X. Hu. A variational autoencoder with deep embedding model for generalized zero-shot learning. In *AAAI*, 2020. 8

[76] T. Ma, M. Bi, J. Zhang, W. Yuan, Z. Zhang, Y. Xie, S. Ding, and L. Ma. Mutually reinforcing structure with proposal contrastive consistency for few-shot object detection. In *ECCV*, 2022. 6, 50

[77] L. V. D. Maaten and G. E. Hinton. Visualizing data using t-sne. In *Journal of Machine Learning Research*, 2008. 24, 38

[78] A. Majee, K. Agrawal, and A. Subramanian. Few-shot learning for road object detection. *ArXiv*, abs/2101.12543, 2021. 26, 42

[79] A. Majee, A. Subramanian, and K. Agrawal. Meta guided metric learner for overcoming class confusion in few-shot road object detection. *ArXiv*, abs/2110.15074, 2021. 26, 42

[80] P. Mangla, M. K. Singh, A. Sinha, N. Kumari, V. N. Balasubramanian, and B. Krishnamurthy. Charting the right manifold: Manifold mixup for few-shot learning. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2207–2216, 2020. 33

[81] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 40, 50, 52, 53

[82] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41, 1992. 26, 47

[83] A. Mishra, M. S. K. Reddy, A. Mittal, and H. A. Murthy. A generative model for zero shot learning using conditional variational autoencoders. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2269–22698, 2018. 8, 28

[84] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning(ICML)*, 2018. 19

[85] B. N. Oreshkin, P. R. López, and A. Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 5, 33

[86] C. Ouyang, C. Biffi, C. Chen, T. Kart, H. Qiu, and D. Rueckert. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *ECCV*, 2020. 26, 42

[87] J. Pan, C. Wang, X. Jia, J. Shao, L. Sheng, J. Yan, and X. Wang. Video generation from single semantic label map. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3728–3737, 2019. 8

[88] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 26

[89] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang. Incremental few-shot object detection. In *CVPR*, June 2020. 6

[90] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. *ICCV*, 2021. 7, 43, 45, 46, 49, 50, 51, 52, 53, 54

[91] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning(ICML)*, 2021. 26, 35

[92] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 47, 49, 52, 53, 58

[93] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019. 5

[94] V. Ranjan, U. Sharma, T. Nguyen, and M. Hoai. Learning to count everything. In *CVPR*, 2021. 58

[95] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 5, 6, 18, 34

[96] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. *ArXiv*, abs/1803.00676, 2018. 1, 34

[97] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 2015. 42, 49

[98] M. Rezaei and M. Shahidi. Zero-shot learning and its applications from autonomous vehicles to covid-19 diagnosis: A review. *Intelligence-Based Medicine*, 3:100005 – 100005, 2020. 26, 42

[99] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 5

[100] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

[101] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, R. Feris, A. Kumar, R. Giryes, and A. M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, June 2018. 6, 10, 18, 20, 21, 23, 24

[102] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, S. Pankanti, R. S. Feris, A. Kumar, R. Giryes, and A. M. Bronstein. Repmet: Representative-based metric learning for classification and few-shot object detection. In *CVPR*, 2019. 6

[103] T. R. Scott, K. Ridgeway, and M. Mozer. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *NeurIPS*, 2018. 5

[104] H. Shao, S. Yao, D. Sun, A. Zhang, S. Liu, D. Liu, J. Wang, and T. F. Abdelzaher. Controlvae: Controllable variational autoencoder. In *ICML*, 2020. 7, 8

[105] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2016. 9

[106] C. Simon, P. Koniusz, R. Nock, and M. T. Harandi. Adaptive subspaces for few-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4135–4144, 2020. 33

[107] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5, 18, 19, 26, 28, 30

[108] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 27, 31, 47

[109] Y. Song, T.-Y. Wang, S. K. Mondal, and J. P. Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 2022. 2

[110] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *CVPR*, 2021. 6, 7, 42, 50

[111] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele. Meta-transfer learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 18

[112] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 5, 18, 19

[113] Y. Tian, Y. Wang, D. Krishnan, J. Tenenbaum, and P. Isola. Rethinking few-shot image classification: a good embedding is all you need? *ArXiv*, abs/2003.11539, 2020. 5, 33

[114] S. Tsutsui, Y. Fu, and D. Crandall. Meta-Reinforced Synthetic Data for One-Shot Fine-Grained Visual Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, June 2019. 1, 9, 16, 20, 21

[115] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 9, 10, 16, 18, 21

[116] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1, 5, 18, 19, 33, 34

[117] W. Wang, Q. Xia, Z. Hu, Z. Yan, Z. Li, Y. Wu, N. Huang, Y. Gao, D. N. Metaxas, and S. Zhang. Few-shot learning by a cascaded framework with shape-constrained pseudo label assessment for whole heart segmentation. *IEEE Transactions on Medical Imaging*, 40:2629–2641, 2021. 26, 42

[118] X. Wang, T. E. Huang, T. Darrell, J. Gonzalez, and F. Yu. Frustratingly simple few-shot object detection. *ArXiv*, abs/2003.06957, 2020. 7, 49, 50, 51

[119] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *ArXiv*, abs/1911.04623, 2019. 33

[120] Y. Wang, Q. Yao, J. T.-Y. Kwok, and L. M. shuan Ni. Generalizing from a few examples: A survey on few-shot learning. *arXiv: Learning*, 2019. 6, 26

[121] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 6

[122] Y.-X. Wang, D. Ramanan, and M. Hebert. Meta-learning to detect rare objects. In *ICCV*, 2019. 6, 7

[123] 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, California Institute of Technology, 2010. 9, 10, 16, 18, 19, 21, 26, 42

[124] D. Wertheimer, L. Tang, and B. Hariharan. Few-shot classification with feature map reconstruction networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8008–8017, 2021. 28, 33, 35

[125] A. Wu, Y. Han, L. Zhu, and Y. Yang. Universal-prototype enhancing for few-shot object detection. *ICCV*, pages 9547–9556, 2021. 6

[126] A. Wu, S. Zhao, C. Deng, and W. Liu. Generalized and discriminative few-shot object detection via svd-dictionary enhancement. In *NeurIPS*, 2021.

[127] A. Wu, Y. Han, L. Zhu, and Y. Yang. Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:4178–4193, 2022. 6

[128] J. Wu, S. Liu, D. Huang, and Y. Wang. Multi-scale positive sample refinement for few-shot object detection. *ArXiv*, abs/2007.09384, 2020. 7, 50, 51

[129] Y. Xiao and R. Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020. 6, 50

[130] C. Xing, N. Rostamzadeh, B. N. Oreshkin, and P. H. O. Pinheiro. Adaptive cross-modal few-shot learning. In *NeurIPS*, 2019. 26, 27, 33

[131] J. Xu and H. Le. Generating representative samples for few-shot classification. In *CVPR*, 2022. 47

[132] J. Xu, H. Le, M. Huang, S. Athar, and D. Samaras. Variational feature disentangling for fine-grained few-shot classification. In *Proceedings of the International Conference on Computer Vision*, 2021. 1, 8, 41

[133] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *ICCV*, 2019. 6, 7

[134] S. Yang, L. Liu, and M. Xu. Free lunch for few-shot learning: Distribution calibration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 5, 26

[135] Y. Yang, F. Wei, M. Shi, and G. Li. Restoring negative information in few-shot object detection. *ArXiv*, abs/2010.11714, 2020. 6

[136] Z. Yang, Y. Wang, X. Chen, J. Liu, and Y. Qiao. Context-transformer: Tackling object confusion for few-shot detection. In *AAAI*, 2020. 6

[137] H.-J. Ye, H. Hu, D. Zhan, and F. Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8805–8814, 2020. 5, 28, 33, 34, 35

[138] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Feature transfer learning for deep face recognition with under-represented data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 10, 13, 21, 22, 23, 24

[139] Y. Yu, Z. Ji, J. Han, and Z. Zhang. Episode-based prototype generating network for zero-shot learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14032–14041, 2020. 8

[140] B. Zhang, X. Li, Y. Ye, Z. Huang, and L. Zhang. Prototype completion with primitive knowledge for few-shot learning. In *CVPR*, pages 3754–3762, 2021. 26, 27

[141] C. Zhang, Y. Cai, G. Lin, and C. Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12200–12210, 2020. 5

[142] J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang. Variational few-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 8, 13, 21, 22, 33

[143] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3010–3019, 2017. 8, 27

[144] R. ZHANG, T. Che, Z. Ghahramani, and Y. S. Yoshua Bengi and. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 9

[145] W. Zhang and Y.-X. Wang. Hallucination improves few-shot object detection. *CVPR*, pages 13003–13012, 2021. 7, 42, 49, 50

[146] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *CVPR*, 2021. 6, 50

[147] P. Zhu, H. Wang, and V. Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *CVPR*, 2020. 42

[148] Y. Zhu, C. Liu, and S. Jiang. Multi-attention meta learning for few-shot fine-grained image recognition. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence(IJCAI)*, 2020. 18, 19